



Universidad de Valladolid

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Matemáticas

Técnicas Topológicas para el Reconocimiento en Visión Computacional

Autor: Adrián Hernández Sanz

Tutor: F. Javier Finat Codes

Prefacio

Planteamiento del problema:

El reconocimiento automático de la forma y/o el movimiento es uno de los problemas más sobresalientes en visión computacional. Buena parte de las aplicaciones industriales están enfocadas hacia el reconocimiento de objetos rígidos \mathbf{B} . Para casos más amplios correspondientes a objetos deformables $\mathbf{B}(\lambda)$ (donde $\lambda \in \Lambda$) o que evolucionan a lo largo del tiempo $\mathbf{B}(t)$, la resolución de este problema sugiere introducir un soporte deformable para la forma S (shape), características cinemáticas para el movimiento M y los procedimientos de estimación para ambos.

Nuestro enfoque utiliza estructuras superpuestas para la “forma” y diferentes tipos de campos para representar sus deformaciones (reales o aparentes) y el movimiento. Los campos que representan la evolución espacio-temporal pueden ser escalares, vectoriales y tensores correspondientes a la representación como funciones (incluidas las funciones de probabilidad de densidad), distribuciones \mathcal{D} de los campos vectoriales (eventualmente aleatorios) o su versión dual dada por los sistemas \mathcal{S} de formas diferenciales o sus productos formales correspondientes a los tensores.

Todos los datos anteriores deben ser estimados preservando características de la forma o del movimiento que permitan organizar la información según varias jerarquías. La estimación es la clave para actualizar, corregir y adaptar las versiones provisionales de la forma o el movimiento, mediante estrategias de refinamientos sucesivos (coarse-to-fine). Por lo tanto, la retroalimentación entre los modelos diferenciales y su estimación estadística desempeña un papel central.

Las distribuciones dispersas e irregulares de los datos capturados por diferentes dispositivos de imagen o rango sugieren el uso de mallas superpuestas para formas y trayectos múltiples para una descripción en teoría de conjuntos de las características de movimiento. Por lo tanto, la topología algebraica básica proporciona un marco natural inicial. Además, para recuperar las propiedades globales o para completar la información disponible es necesario identificar las correspondencias entre los datos, así como los modelos de propagación usando estructuras de datos cinemáticos que actualizan la información disponible en trozos frecuentemente desconectados entre sí a los que llamaremos “parches”.

Desde el punto de vista de la inteligencia artificial (AI: Artificial Intelligence), se introducen diferentes variantes de redes neuronales artificiales (ANN, Artificial Neural Networks). Se describen en términos de varias capas con funciones sobre nodos y diferentes tipos de conexividad que facilitan el aprendizaje de objetos o de tareas. La variante a utilizar debe proporcionar un apoyo a las funcionalidades características más relevantes para el reconocimiento: detección, descripción y clasificación. Las ANNs utilizan una estructura multicapa con un enrejado regular (habitualmente cuadrangular) para cada capa con pesos variables para los nodos en cada capa. Los pesos se ajustan en función de la disparidad entre los resultados obtenidos y los deseados (función objetivo).

Los enfoques habituales para las tareas de aprendizaje y formación en las ANNs siguen un esquema secuencial entre las capas sucesivas. Las entradas para cada capa se modifican de acuerdo con las transformaciones locales dadas por convoluciones discretas. La arquitectura de las ANNs habituales está compuesta por 5 capas (entrada, salida y tres capas ocultas intermedias). En el esquema inicial (aún predominante) las células en cada capa eran idénticas y sólo estaban conectadas con células en capas adyacentes. Las operaciones se describen en términos de productos tensoriales por matrices de pesos (núcleo de una convolución discreta). Los pesos w_{ij} en los nodos se modifican inicialmente de manera aleatoria, y después por medio de campos de control para lograr una convergencia del resultado proporcionado

por la red hacia los modelos esperados.

El desarrollo de los procedimientos de aprendizaje profundo (Deep Learning, en lo sucesivo) utiliza las ANNs con un número muy elevado de capas, un número muy alto de células (actualmente de dos tipos) por capa eventualmente conectadas. El entrenamiento utiliza millones en lugar de unos pocos miles de datos para el entrenamiento y puede llevarse a cabo de forma distribuida tanto en la misma capa como en capas contiguas, con procedimientos de realimentación.

Los algoritmos desarrollados inicialmente para las ANN utilizaban procedimientos de fuerza bruta, asignación aleatoria de pesos en procesos inicialmente secuenciales (posteriormente en la misma capa usando convoluciones) o bien refinamientos basados en el método del gradiente (descenso más rápido) para funciones objetivo sencillas asociadas a problemas de optimización. La introducción de las convoluciones más sencillas y de funciones de pesos más sofisticadas permitió ya en los setenta mejoras sustanciales sobre la adaptabilidad, pero siempre con una única función objetivo fija para cada red neuronal.

Desde los años ochenta se han desarrollado refinamientos adaptativos en diferentes contextos vinculados a Algoritmos Genéticos (Genetic Algorithms o GAs), Programación Evolutiva (Evolutionary Programming o EP) y Mapas Auto-Organizantes (Self-Organizing Maps ó SOM) desarrollaron estrategias alternativas más eficientes. Los avances recientes llevados a cabo sobre Redes Neuronales Convolucionales (CNN) y las mejoras en estrategias recurrentes (RNN) sobre arquitecturas hardware distribuidas mucho más eficientes han dado lugar a un crecimiento exponencial de contribuciones en los últimos diez años bajo la etiqueta de Deep Learning.

Los *objetivos principales* de este TFG consiste en reformular de forma rigurosa los modelos geométricos y topológicos subyacentes para mejorar el rendimiento de redes adaptando algunas de las ideas de SOM al contexto Deep Learning.

Estrategias propuestas:

Para alcanzar el objetivo descrito, se adaptan modelos y herramientas procedentes de la topología algebraica, la geometría y topología diferencial para un modelado más eficiente de espacios y , en menor medida, de procesos de aprendizaje.

La *Topología Algebraica* se utiliza para describir *trayectorias* descritas por diferentes agentes como “multi-caminos” $\Gamma = (\gamma_1, \dots, \gamma_k)$, así como para el PL¹-modelado tanto de objetos (en términos de grupos de cadenas) como de PL-funcionales (grupos de co-cadenas) definidos sobre dichos objetos. Aunque el marco más habitual es el simplicial (o su extensión al caso singular), en este trabajo se introduce el cuboidal, pues facilita la transición hacia modelos parametrizados típicos de la Geometría y la Topología Diferencial.

La *Geometría Diferencial* se utiliza para propiedades formales de los diferentes tipos de campos (escalares, vectoriales, tensoriales). Los “eventos” relativos a la (des)aparición de agentes o cambios en las apariencias o atributos de dichas apariencias (color, p.e.) se abordan en términos de singularidades de funciones, lugares críticos de distribuciones de campos o de sistemas de formas. Todos ellos se agrupan con más generalidad en distribuciones de campos tensoriales de bigrado (r, s) “bajo”, es decir, con $r + s \leq 4$.

La *Topología Diferencial* se utiliza para describir las “variaciones” en la topología de las hipersuperficies de nivel asociadas a campos (lugar crítico de una función en el caso escalar) o de distribuciones (lugar discriminante de una aplicación). Para ello, se adaptan resultados de la Teoría Clásica de Morse; su reinterpretación en términos de la (co)homología de las superficies de nivel extiende el uso de herramientas de Topología Algebraica a la (co)homología persistente.

Nuestras contribuciones:

Las contribuciones más relevantes desde el punto de vista matemático conciernen a un tratamiento unificado de los diferentes tipos de campos para el modelado de objetos y el aprendizaje de multi-tareas en entornos distribuidos; el carácter distribuido en Matemáticas se representa como la coexistencia e interacción entre diferentes tipos de campos. En

¹PL \equiv "Piecewise Linear"

cualquier caso, los campos se interpretan como secciones de fibrados o, con más generalidad, fibraciones. Este tratamiento unificado proporciona una versión global en términos de (funcionales sobre) fibraciones topológicas como extensión de los fibrados vectoriales de la Geometría Diferencial. Asimismo, se introducen estimadores robustos para formas eventualmente cambiantes y adaptativos para los procesos vinculados al reconocimiento de objetos y de la función que desempeñan. Esta idea permite incorporar tanto las funciones objetivo-coste de procesos de optimización convencionales, como las herramientas de control robusto-adaptativos a objetos y tareas.

Una aportación relevante es la introducción de elementos centrales para el reconocimiento (detectores, descriptores, clasificadores) en términos de objetos habituales en Fibrados Vectoriales o, con más generalidad, Fibraciones. Esta introducción proporciona un marco común para unificar la inmensa casuística presente en Sistemas Expertos y sus aplicaciones al Reconocimiento.

Con ello, se espera mejorar el rendimiento de Sistemas Expertos en relación con la aparición (y/o desaparición) de eventos. Desde el punto de vista matemático, la clave consiste en reformular las distribuciones \mathcal{D} de campos vectoriales y de sistemas \mathcal{S} de formas diferenciales, no necesariamente regulares. Nuestra propuesta consiste en desarrollar herramientas de control adaptativo que se formulan en términos de filtraciones graduadas cambiantes asociadas al procesamiento de la información parcial de tipo visual.

Índice general

1. Elementos básicos	9
1.1. Deformación y pegado de caminos múltiples.	9
1.1.1. Teoría básica de trenzas.	9
1.1.2. Teoría básica de nudos.	13
1.2. Estructuras y deformaciones PL-superpuestas.	15
1.2.1. Homología cuboidal.	16
1.2.2. Homología relativa cuboidal.	19
1.2.3. Subdivisiones y Escisión en complejos cuboidales.	21
1.2.4. Cohomología cuboidal.	22
1.3. Teoría básica de Morse.	24
1.3.1. Funciones de Morse.	24
1.3.2. Homotopía en términos de valores críticos.	26
1.3.3. Desigualdades de Morse.	27
1.3.4. Reconocimiento bajo información incompleta.	28
1.4. Elementos de fibrados vectoriales.	28
1.4.1. Nociones básicas de Fibraciones.	29
1.4.2. Enfoque funtorial.	32
1.4.3. Algebra lineal de fibrados vectoriales.	34
1.5. Flujos tensoriales en Matemáticas e Ingeniería.	35
1.5.1. Tensores Cartesianos.	36
1.5.2. Tensores Cartesianos para representar deformaciones.	39
1.5.3. Campos Tensoriales.	42
1.5.4. Contracción y expansión.	43
2. Contribuciones teóricas	45
2.1. Topología Algebraica Computacional.	45
2.1.1. Mallas en PL-formas.	46
2.1.2. Las α -formas.	47
2.1.3. Colapso y Expansión.	51
2.1.4. Representaciones simbólicas.	54
2.2. Topología Diferencial Computacional.	57
2.2.1. Auto-intersecciones.	58
2.2.2. Persistencia.	60
2.2.3. Estabilidad.	65
2.3. Sistemas Expertos y fibraciones.	70
2.3.1. Un ejemplo ilustrativo: Imágenes Digitales.	70
2.3.2. Una introducción a las Redes Neuronales.	72
2.3.3. Algunas extensiones de las ANN.	79
2.3.4. Aplicaciones Auto-Organizantes de Kohonen.	82
3. Un enfoque computacional en Reconocimiento	85
3.1. Métodos topológicos en Reconocimiento.	86
3.1.1. Sobre el concepto de “forma”.	87
3.1.2. Descriptores usados en Visión Computacional.	89
3.1.3. Una interpretación en secuencias de video.	91
3.1.4. Modelo tensorial para aspectos funcionales del ciclo PAC.	94
3.2. Una reformulación de Sistemas Expertos.	96
3.2.1. Detectores.	97
3.2.2. Descriptores.	98

3.2.3.	Clasificadores.	101
3.2.4.	Una posible extensión a reconocimiento en vídeo.	103
3.3.	Desde SOM a Deep Learning.	104
3.3.1.	Redes neuronales convolucionales.	104
3.3.2.	Incorporando la topología a las CNN.	105
3.3.3.	Memoria y aprendizaje.	106
3.3.4.	¿Qué red elegir?.	108
3.4.	Flujos tensoriales en Variedades y TensorFlow en ANN.	109
3.4.1.	Cálculo Tensorial en el marco suave.	109
3.4.2.	Conservando el grado total.	111
3.4.3.	Incorporando pesos variables.	112
3.4.4.	Contracción y expansión.	113
4.	Conclusiones	117
	Bibliografía	119

Capítulo 1

Elementos básicos

1.1. Deformación y pegado de caminos múltiples.

En esta sección se introducen algunas nociones básicas sobre caminos múltiples o multi-caminos que permiten describir las diferentes trayectorias llevadas a cabo por un número finito de agentes que se presentan en escenas reales. También se pueden aplicar a otros objetos a reconocer como por ejemplo un nudo de ADN ó estructuras filamentosas del aparato circulatorio o del sistema nervioso, p.e..

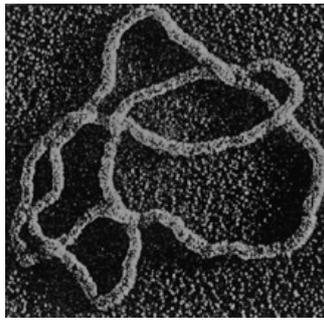


Figura 1.1: Nudo de ADN

Para ello, se presentan algunas ideas básicas de la teoría de trenzas y nudos que facilita un tratamiento topológico de las deformaciones en este espacio.

Definición 1.1 Sea $\gamma : [0, 1] \rightarrow X$ un camino abierto y $\gamma : \mathbb{S}^1 \rightarrow X$ un camino cerrado. Un *multi-camino* se define como una colección finita de caminos abiertos ó cerrados que denotamos mediante

$$\Gamma_{ab} = \{\gamma_i : [0, 1] \rightarrow X, \text{ para todo } i \in \mathbb{N}\}$$

$$\Gamma_{ce} = \{\gamma_i : \mathbb{S}^1 \rightarrow X, \text{ para todo } i \in \mathbb{N}\}.$$

Denotamos mediante Γ al espacio de multi-caminos o caminos múltiples $(\gamma_1, \dots, \gamma_k)$ donde cada camino γ_i puede ser tanto abierto como cerrado, es decir, topológicamente equivalente a $[0, 1]$ como a \mathbb{S}^1 .

En Topología Algebraica se describen las deformaciones de un camino en otro mediante homotopías. Aquí se presenta una situación más complicada porque las deformaciones a estudiar dan lugar a trenzas que no deberían reconectarse para poder mantener un control de la topología. Asimismo, la comparación (incluso cuando no se reanudan) presenta una mayor complejidad en el espacio.

1.1.1. Teoría básica de trenzas.

Comencemos dando una definición de lo que es una *trenza* desde el punto de vista matemático.

Definición 1.2 Sea \mathbb{D} el cubo unidad $[0, 1]^3$ en \mathbb{R}^3 . Consideremos una colección ordenada de n puntos en la cara superior p_1, \dots, p_n (puntos finales) y otra colección ordenada de n puntos p'_1, \dots, p'_n (puntos iniciales) de \mathbb{D} . A continuación unimos un par de puntos mediante un arco/segmento d_i que cumple las siguientes condiciones:

1. d_1, d_2, \dots, d_n son disjuntos.
2. Cada d_i conecta un p_j con un p'_k , donde j y k pueden o no ser iguales, pero d_i no puede unir un p_j con un p_k (o p'_j con p'_k).
3. Cada plano de nivel E_s (planos horizontales), con $0 \leq s \leq 1$, interseca a cada d_i en un único punto.

Al arco d_i se le llama el i -ésimo filamento de la trenza. Al resultado de esta construcción le llamamos una n -Trenza (Braid, Tresse), o una trenza con n filamentos.

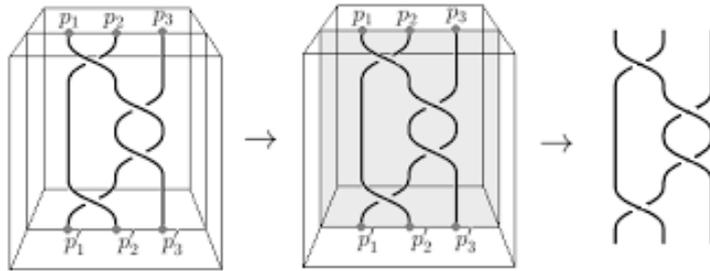


Figura 1.2: Ejemplo 3-Trenza

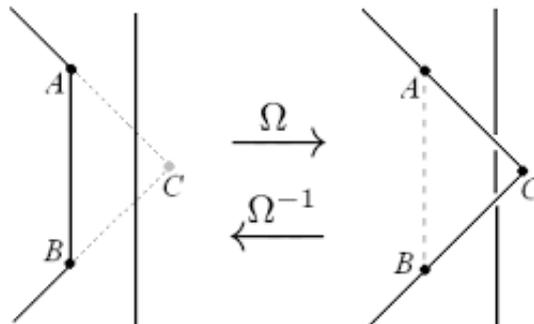
Denotamos al conjunto de trenzas de n filamentos por \mathfrak{B}_n (braids). De forma análoga al caso clásico, se obtiene una estructura de grupo para las (clases de) trenzas con respecto un producto interno. Veamos esto con mas detalle.

La primera cuestión que se plantea es la equivalencia entre dos trenzas. Para que dos trenzas sean equivalentes debe ocurrir que si tomamos una de las dos trenzas y aplicamos movimientos elementales sobre ella podamos llegar a obtener la otra trenza. Para formalizar esta idea definimos los movimientos elementales que podemos aplicar a las trenzas.

Definición 1.3 Sea AB un trozo del filamento d . Sea C un punto en \mathbb{D} tal que el triángulo $\triangle ABC$ solo interseca AB y no interseca con ningún otro filamento. Supongamos además que:

1. $AC \cup CB$ interseca a E_s entre A y B en un único punto.
2. Entonces, podemos reemplazar AB por los dos segmentos $AC \cup CB$.

A esta acción, denotada por Ω , la definimos como *movimiento elemental en una trenza*. La operación inversa sería revertir esta acción, Ω^{-1} que se considera también como movimiento elemental.



De esta manera podemos formalizar el concepto de equivalencia.

Definición 1.4 Una n -Trenza α se dice que es *igual o equivalente* a otra n -Trenza α' , y lo denotamos por $\alpha \sim \alpha'$, si α puede ser deformada en α' por la aplicación de una serie finita de movimientos elementales.

Es inmediato comprobar que la relación anterior es una relación de equivalencia.

En el contexto matemático de esta teoría, pasando al cociente módulo dicha relación, las trenzas equivalentes se representan mediante un único elemento. Denotamos por \mathbf{B}_n al conjunto de n -Trenzas no-equivalentes (es decir, $\mathbf{B}_n = \mathfrak{B}_n / \sim$).

Supongamos que tenemos dos trenzas en las que se presentan un número distinto de cruzamientos (por ejemplo, en la primera se produce un número p de cruzamientos y en la segunda $q \neq p$); entonces esas dos trenzas no pueden ser equivalentes una de la otra, es decir, $\alpha_p \not\sim \alpha'_q$. De este modo, se obtiene un primer criterio tosco que permite distinguir una trenza de otra.

A continuación, desde el punto de vista del algebra abstracto, daremos al conjunto de n -Trenzas una estructura de grupo. Para ello debemos describir el producto interno en este conjunto junto con el elemento neutro e inverso. De esta manera damos una *presentación algebraica del grupo de trenzas*.

Definición 1.5 Sean α_1 y α_2 dos n -Trenzas en \mathfrak{B}_n . Podemos crear una tercera n -Trenza a partir de estas dos que definimos como el *producto* de ellas que denotaremos por $\alpha_1\alpha_2$ y que consiste en hacer corresponder los puntos p_i^1 con los puntos p_i^2 .

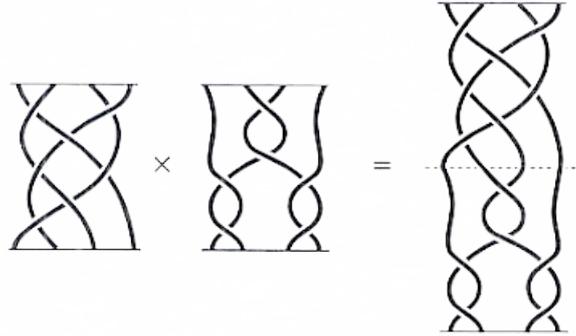


Figura 1.3: Ejemplo ilustrativo

Proposición 1.1 Supongamos que α , α' , $\bar{\alpha}$, $\bar{\alpha}'$ tal que $\alpha \sim \alpha'$ y $\bar{\alpha} \sim \bar{\alpha}'$. Entonces,

$$\alpha\bar{\alpha} \sim \alpha'\bar{\alpha}'$$

Demostración: Como $\alpha \sim \alpha'$ sabemos que existe una secuencia finita,

$$\alpha = \alpha_0 \xrightarrow{\Omega^{\pm 1}} \alpha_1 \xrightarrow{\Omega^{\pm 1}} \dots \xrightarrow{\Omega^{\pm 1}} \alpha_m = \alpha'.$$

Esta secuencia induce a su vez otra secuencia,

$$\alpha\bar{\alpha} = \alpha_0\bar{\alpha} \xrightarrow{\Omega^{\pm 1}} \alpha_1\bar{\alpha} \xrightarrow{\Omega^{\pm 1}} \dots \xrightarrow{\Omega^{\pm 1}} \alpha_m\bar{\alpha} = \alpha'\bar{\alpha}.$$

Por tanto, tenemos que $\alpha\bar{\alpha} \sim \alpha'\bar{\alpha}$.

De la misma manera vemos que $\alpha'\bar{\alpha} \sim \alpha'\bar{\alpha}'$ y finalmente, por la transitividad de la relación, tenemos que $\alpha\bar{\alpha} \sim \alpha'\bar{\alpha} \sim \alpha'\bar{\alpha}'$.

□

Proposición 1.2 El producto de trenzas es asociativo, es decir,

$$(\alpha_1\alpha_2)\alpha_3 \sim \alpha_1(\alpha_2\alpha_3).$$

A pesar de que el producto es una operación asociativa, no es un producto conmutativo, es decir, no tiene por qué ocurrir que $\alpha_1\alpha_2 \sim \alpha_2\alpha_1$.

Proposición 1.3 Sea α_e la trenza que se muestra en Figura 1.4. Entonces, para toda n -Trenza α ,

$$\alpha\alpha_e \sim \alpha \text{ y } \alpha \sim \alpha\alpha_e.$$

Llamamos a esta trenza α_e la trenza identidad (ó trivial) y la denotaremos por $\mathbf{1}_n$.

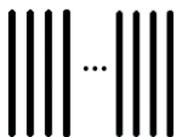


Figura 1.4: n -Trenza trivial.

Proposición 1.4 Para cada n -Trenza α existe una n -Trenza $\bar{\alpha}$ tal que

$$\alpha\bar{\alpha} \sim \mathbf{1}_n \text{ y } \bar{\alpha}\alpha \sim \mathbf{1}_n.$$

Las demostraciones se ilustran mediante construcciones gráficas en [1].

Para definir una estructura de grupo, debemos considerar las clases de equivalencia de n -Trenzas, \mathbf{B}_n . Como consecuencia de la **Proposición 1.1** tenemos que el producto de dos elementos $[\alpha_1]$ y $[\alpha_2]$ en \mathbf{B}_n está bien definido, pues $[\alpha_1][\alpha_2] = [\alpha_1\alpha_2]$. Con los elementos descritos podemos dotar a \mathbf{B}_n con una estructura de grupo no-conmutativo:

Teorema 1.1 El conjunto de las clases de equivalencia de n -Trenzas, \mathbf{B}_n , forma un grupo. A este grupo se le llama grupo de n -Trenzas de Artin.

Demostración: El producto (operación binaria) está dado por la **Definición 1.5**; la asociatividad es una consecuencia de la **Proposición 1.2**. El elemento identidad es $\mathbf{1}_n$ en virtud de la **Proposición 1.3**. Por último, el elemento inverso de $[\alpha]$, denotado por $[\alpha]^{-1}$, es $[\alpha^{-1}]$, como consecuencia de la **Proposición 1.4**. □

Si consideramos una aplicación f del conjunto de todas las trenzas \mathfrak{B}_n en algún objeto algebraico que cumple la propiedad:

$$\alpha \sim \alpha' \Rightarrow f(\alpha) = f(\alpha'),$$

decimos que f es un invariante topológico de \mathfrak{B}_n o más precisamente un *invariante de trenza*. Su recíproco proporciona otro criterio para saber si 2 trenzas cualesquiera son equivalentes.

Un ejemplo de la construcción descrita consiste en considerar la aplicación π de permutación y dos trenzas α_1, α_2 tales que

$$\pi(\alpha_1) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \text{ y } \pi(\alpha_2) = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}.$$

Es claro que $\pi(\alpha_1) \neq \pi(\alpha_2)$, por lo que $\alpha_1 \not\sim \alpha_2$.

Si tenemos una n -Trenza tal que

$$\pi(\alpha) = \begin{pmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{pmatrix},$$

llamamos a este tipo de trenza una n -Trenza Pura y denotamos mediante \mathbf{P}_n , con $n \geq 1$, al conjunto de este tipo de trenzas.

Con la notación anterior damos una presentación para el grupo de las trenzas (de la misma forma que en el Seminario N.Bourbaki dado por Bennequin sobre el enfoque desarrollado por V.I.Arnold's).

Denotamos g_i como la operación en la que cruzamos el i -ésimo filamento sobre el $(i+1)$ -ésimo y al operador g_i^{-1} como el cruzamiento del $(i+1)$ -ésimo filamento sobre el i -ésimo filamento.

Proposición 1.5 *Cualquier n -Trenza α (en \mathbf{B}_n) se puede escribir como el producto de los elementos del conjunto $\{g_i^{\pm 1}\}$, es decir,*

$$\alpha = g_{i_1}^{\varepsilon_1} g_{i_2}^{\varepsilon_2} \dots g_{i_k}^{\varepsilon_k},$$

donde $1 \leq i_1, i_2, \dots, i_k \leq n-1$ y $\varepsilon_i = \pm 1$ para $i = 1, \dots, k$.

A continuación damos la presentación del grupo de trenzas:

Teorema 1.2 *Para todo $n \geq 1$ el grupo de n -Trenzas \mathbf{B}_n tiene la siguiente presentación,*

$$\mathbf{B}_n = \langle g_1, g_2, \dots, g_{n-1} \mid \begin{array}{l} g_i g_j = g_j g_i \text{ para } |i - j| > 1 \\ g_i g_{i+1} g_i = g_{i+1} g_i g_{i+1} \text{ para } i = 1, 2, \dots, n-2 \end{array} \rangle.$$

Un invariante algebraico asociado a las propiedades algebraicas de las trenzas es el siguiente:

Proposición 1.6 *Sea α una n -Trenza dada por $\alpha = g_{i_1}^{\varepsilon_1} g_{i_2}^{\varepsilon_2} \dots g_{i_k}^{\varepsilon_k}$, donde $\varepsilon_i = \pm 1$. Definimos la suma de los exponentes de α (en términos de g_i), denotada por $\exp(\alpha)$, como*

$$\exp(\alpha) = \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_k.$$

Entonces, $\exp(\alpha)$ es un invariante del grupo de trenzas, es decir,

$$\alpha_1 \sim \alpha_2 \Rightarrow \exp(\alpha_1) = \exp(\alpha_2).$$

Otro de los aspectos importantes en esta teoría es la relación que hay entre el grupo de trenzas y el grupo de simetría \mathbf{S}_n . El objetivo aquí es encontrar una relación o un conjunto de relaciones para las cuales se pueda pasar de mi grupo infinito de trenzas a un grupo cociente finito. La solución de a este problema motiva que debemos incluir más relaciones en el grupo de presentación de nuestro grupo de trenzas. Como las trenzas pueden ser interpretadas como objetos geométricos una primera intuición que tenemos es buscar una relación geométrica.

Posiblemente la más sencilla sea lo que en esta teoría se conoce como Δ -operador (o como la acción “desanudar”) que consiste básicamente en intercambiar la superposición de un filamento en el cruce con otro filamento.

El Δ -operador se puede interpretar como una transformación del operador g_i en el g_i^{-1} (o viceversa). Dicho de otra forma, el Δ -operador es una transformación de g_i^2 en $\mathbf{1}$. Por tanto, consideraremos esta última relación en nuestro grupo \mathbf{B}_n . Esta idea permite enunciar la siguiente proposición.

Proposición 1.7 *Sea $\mathbf{B}_n(2)$ el grupo cuya presentación es la misma que \mathbf{B}_n con la relación extra $g_i^2 = \mathbf{1}$, para todo $i = 1, 2, \dots, n-1$. Se tiene que $\mathbf{B}_n(2)$ es isomorfo a \mathbf{S}_n y por consiguiente $|\mathbf{B}_n(2)| = |\mathbf{S}_n| = n!$.*

Usando el resultado anterior, podemos dar una presentación de algún subgrupo de \mathbf{S}_n . Para $n \geq 2$, uno de los subgrupos más importantes del grupo de simetrías es el llamado *grupo alternado*, \mathbf{A}_n . En la referencia [1] puede encontrarse una presentación del grupo \mathbf{A}_n dado a partir de unas modificaciones de la presentación del grupo de simetría.

1.1.2. Teoría básica de nudos.

Para enlazar la teoría de nudos con la teoría de trenzas nos apoyamos en el teorema de Alexander. Antes de enunciar este teorema daremos algunos resultados para que no haya confusión con ciertos términos como por ejemplo *nudo* y *enlace*. Una referencia para esta parte es [13], en la que encontramos un estudio más a fondo sobre esta parte de las Matemáticas. En este trabajo suponemos que los nudos son curvas cerradas, suaves y sin autointersecciones, tal y como se muestra en la siguiente figura.

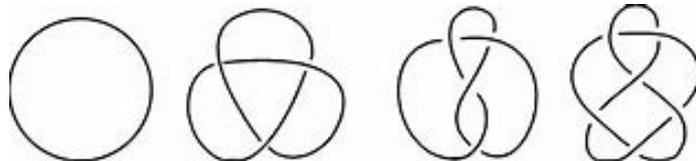


Figura 1.5: Nudos en \mathbb{R}^3

Una definición más formal de nudo desde el punto de vista matemático es la siguiente:

Definición 1.6 Un *nudo* K es embebimiento de S^1 en alguna 3-variedad M .

Un ejemplo especial de nudo es el conocido *nudo tórico*. Es un tipo especial de nudo que se encuentra en la superficie de un toro T^2 en \mathbb{R}^3 . Cada nudo tórico se especifica por un par de números enteros p y q tales que $\text{mcd}(p, q) = 1$. Un *enlace tórico* está caracterizado por la condición $\text{mcd}(p, q) \neq 1$, en cuyo caso, será el número de componentes. Un *nudo tórico* es *trivial* si y sólo si p o q es igual a ± 1 . El ejemplo más sencillo no trivial es el nudo $(2, 3)$, también conocido como nudo de trébol.

El nudo (p, q) se puede describir por la parametrización:

$$x = r \cos(p\phi); \quad y = r \sin(p\phi); \quad z = -\sin(q\phi)$$

donde $r = \cos(q\phi) + 2$ y $0 < \phi < 2\pi$.

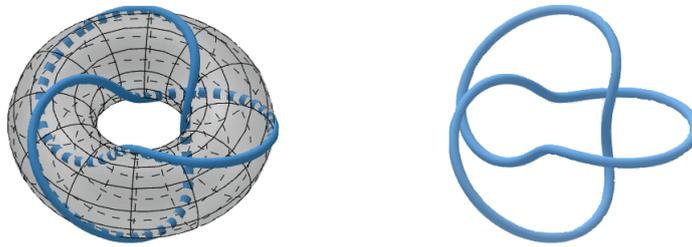
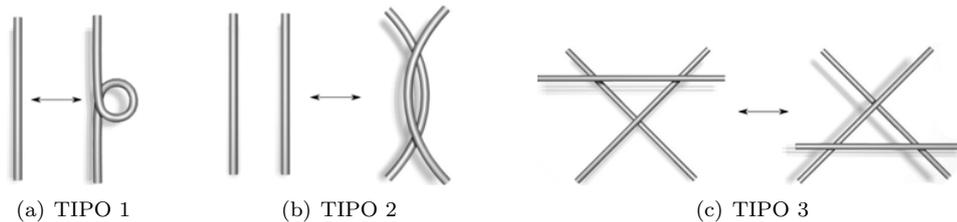


Figura 1.6: Nudo tórico $(2,3)$

La equivalencia entre dos nudos es similar a la formulada en teoría de trenzas: *Dos nudos K y K' son equivalentes si mediante una serie de movimientos aplicados a K se puede obtener K'* . Los movimientos que consideramos en esta teoría son los conocidos como *movimientos de Reidemeister* que constan de tres tipos diferentes:

- **Tipo 1:** Girar y desenroscar en cualquier dirección.
- **Tipo 2:** Superponer un filamento con otro.
- **Tipo 3:** Mover un filamento por debajo o encima de otro/s.



Es fácil darse cuenta de que estos movimientos se pueden obtener a través de movimientos elementales Ω aplicados sobre los filamentos de los nudos como hemos descrito en la teoría de trenzas del principio.

Análogamente al enfoque de la Topología conjuntista, para realizar la suma conexa de dos nudos se retira un segmento de un filamento de cada uno de ellos y a continuación se pegan los extremos que quedan libres (en Topología para obtener la suma conexa de dos toros se recorta un disco \mathbb{D}^2 de cada uno y se identifican los bordes de los espacios vacíos que quedan).

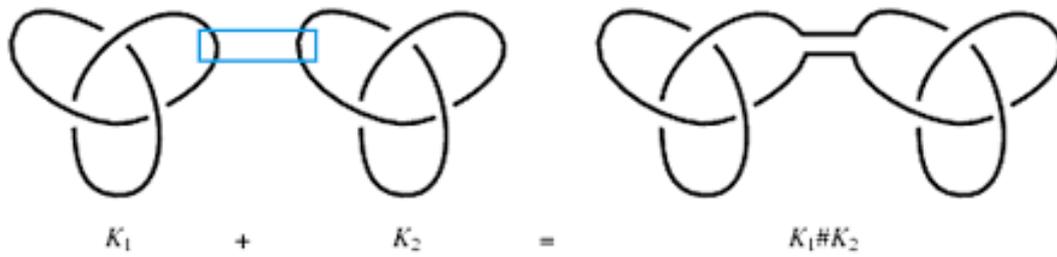


Figura 1.7: Sumas conexas de K_1 y K_2

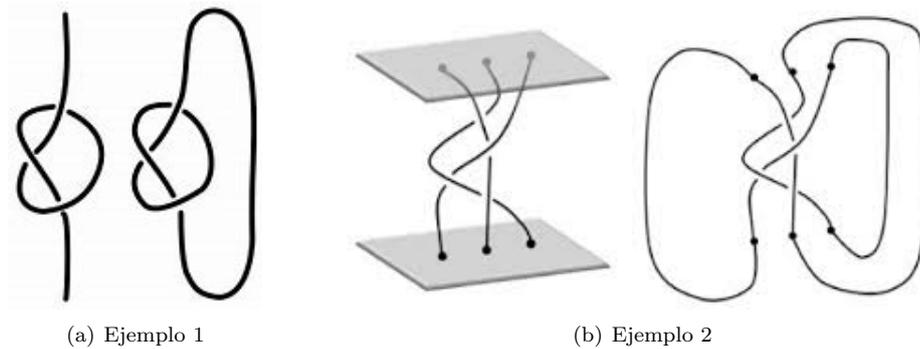
En presencia de una colección ordenada de nudos, K_1, \dots, K_n con $n > 1$, se obtiene una n -componente o enlace. La equivalencia entre dos enlaces L_1 y L_2 se da cuando cada nudo K_1^i es equivalente a algún nudo K_2^j .

Para finalizar esta parte enunciaremos un resultado muy importante que permite relacionar la teoría de trenzas y la teoría de nudos.

Teorema 1.3 (Teorema de Alexander) *Cualquier nudo o enlace se puede representar como una trenza cerrada.*

La demostración puede consultarse en [1].

Una primera idea de como podemos ver un nudo como una trenza sería pegando los puntos p_i con los p'_i de la trenza.



Este resultado no implica que dos trenzas distintas representen dos nudos o enlaces distintos. Puede ocurrir en la teoría de nudos estudiada a través de la teoría de trenzas que dos trenzas distintas den lugar a nudos equivalentes (la propiedad de conmutatividad sí es cierta en los nudos). La respuesta a este problema es el famoso teorema de Markov junto con los dos tipos de movimientos de Markov.

1.2. Estructuras y deformaciones PL-superpuestas.

La homología consiste en la asignación de una colección finita de grupos y homomorfismos de grupos a un espacio topológico. Esto se puede realizar de diversas formas, lo que da origen a diferentes Teorías de Homología que proporcionan los mismos invariantes para espacios con descomposiciones celulares.

En \mathbb{R}^n los símlices, los cubos y las células n -dimensionales son homeomorfos entre sí; sin embargo, dependiendo del marco teórico o de las aplicaciones consideradas a otras áreas una homología puede ser más intuitiva (simplicial), ordenada (cuboidal) o flexible (celular).

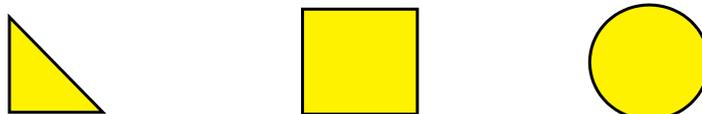


Figura 1.8: Símplex, cubo y célula 2-dimensional, respectivamente

En esta sección se presta más atención a la (co)homología cuboidal [4]. Como ya se comentó en el **Prefacio**, esta homología facilita la transición hacia modelos parametrizados típicos de la Geometría y la Topología Diferencial. Además, permite parametrizar los objetos que se desee reconocer, pudiendo recubrirlos con mallas cuboidales reparametrizables usando argumentos sencillos de subdivisión regular.

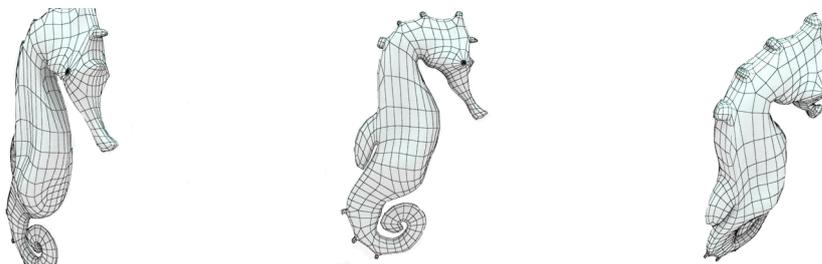


Figura 1.9: Objeto recubierto por una malla cuboidal

En ocasiones, estas estructuras pueden presentarse de forma superpuestas unas con otras. Además y de forma similar a lo que ocurre en algunas aplicaciones a Geometría Compleja Singular, pueden aparecer fenómenos de colapso o, por el contrario, de “emergencia” de ciclos reales (fenómenos de “suspensión”). Estos fenómenos aparecen asociados a “eventos” correspondientes a componentes u objetos que aparecen o desaparecen en los modelos dependiendo de la localización de un observador eventualmente móvil (estos conceptos serán tratados más adelante).

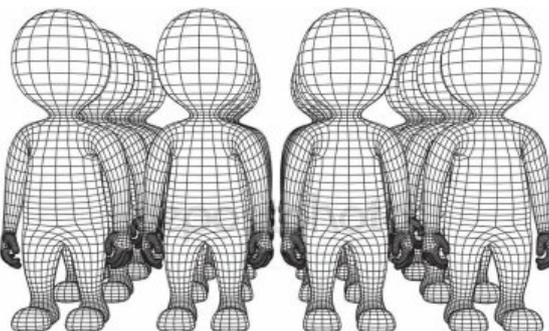


Figura 1.10: Estructuras PL-superpuestas

1.2.1. Homología cuboidal.

Empezamos definiendo la noción de n -cubo, que es la generalización natural del intervalo unidad $I = [0, 1]$ ó del cuadrado unidad $I \times I$. Esto nos permite considerar subconjuntos de varias dimensiones en un espacio topológico.

Definición 1.7 Llamamos n -cubo, y lo denotamos por I^n , al conjunto $[0, 1]^n = [0, 1] \times \dots \times [0, 1]$ para todo $n \geq 0$.

Para el caso $n = 0$ se considera el conjunto como un único punto.

Definición 1.8 Un n -cubo singular en un espacio topológico X es una aplicación continua $T : I^n \rightarrow X$ para todo $n \geq 0$.

En ocasiones, se identifica un cubo singular con su imagen en X .

Denotamos por $Q_n(X)$ al grupo abeliano libre generado por el conjunto de todos los n -cubos singulares en X . Cualquier elemento de este grupo tiene una única representación como una combinación lineal con coeficientes enteros de n -cubos singulares en X (si $u \in Q_n(X)$, u tiene una única representación de la forma $u = \sum_i n_i T_i$).

Definición 1.9 Un n -cubo singular T es *degenerado* si existe un entero i , $1 \leq i \leq n$, tal que $T(x_1, \dots, x_n)$ no depende de x_i .

Sea $D_n(X)$ el subgrupo de $Q_n(X)$ generado por los n -cubos singulares degenerados, y sea $C_n(X)$ el grupo cociente $Q_n(X)/D_n(X)$ llamado el grupo de n -cadenas cuboidales singulares en X o n -cadenas en X .

Un concepto fundamental para la definición de homología es el de la aplicación borde pero antes de pasar a este concepto tenemos que dar un breve conocimiento de lo que son las “caras” de un n -cubo singular.

Definición 1.10 Sea T un n -cubo singular. Definimos, para todo i , $1 \leq i \leq n$, $(n-1)$ -cubos singulares

$$A_i T, B_i T : I^{n-1} \rightarrow X$$

por las fórmulas

$$A_i T(x_1, \dots, x_{n-1}) = T(x_1, \dots, x_{i-1}, 0, x_i, \dots, x_{n-1}),$$

$$B_i T(x_1, \dots, x_{n-1}) = T(x_1, \dots, x_{i-1}, 1, x_i, \dots, x_{n-1}).$$

Llamamos a $A_i T$ la i -cara frontal de T y a $B_i T$ la i -cara trasera de T .

Estos operadores satisfacen las siguientes identidades para $1 \leq i < j \leq n$:

$$A_i A_j(T) = A_{j-1} A_i(T); \quad B_i B_j(T) = B_{j-1} B_i(T);$$

$$A_i B_j(T) = B_{j-1} A_i(T); \quad B_i A_j(T) = A_{j-1} B_i(T).$$

Definición 1.11 El *operador borde de orden n* , para $n \geq 1$, es el homomorfismo $\partial_n : Q_n(X) \rightarrow Q_{n-1}(X)$ definido por

$$\partial_n(T) = \sum_{i=1}^n (-1)^i [A_i T - B_i T].$$

Este operador dota a los cubos singulares de una dimensión menos de una orientación y cumple dos propiedades muy importantes:

$$\partial_{n-1}(\partial_n(T)) = 0 \quad [n > 1],$$

$$\partial_n(D_n(X)) \subset D_{n-1}(X) \quad [n \geq 0].$$

A consecuencia de la segunda, ∂_n induce un homomorfismo $C_n(X) \rightarrow C_{n-1}(X)$, que lo denotaremos por la misma notación que el operador borde.

Entonces tenemos una secuencia de grupos de homomorfismos

$$\dots \rightarrow C_n(X) \xrightarrow{\partial_n} C_{n-1}(X) \xrightarrow{\partial_{n-1}} \dots \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \xrightarrow{\partial_0} 0$$

donde, por definición, $\partial_0 = 0$. Llamamos a esta secuencia *complejo de cadenas*.

A continuación definimos $Z_n(X) = \text{Ker } \partial_n$ (el conjunto de los *ciclos*) y $B_n(X) = \text{Im } \partial_{n+1}$ (el conjunto de los *bordes*). Por la primera propiedad del operador borde podemos afirmar que $B_n(X) \subseteq Z_n(X)$, lo que nos permite dar la siguiente definición.

Definición 1.12 El *n -ésimo grupo de homología de un espacio topológico X* es el grupo cociente $H_n(X) = Z_n(X)/B_n(X)$.

Proposición 1.8 Si X es un conjunto formado por un único punto, entonces

$$H_n(X) = \begin{cases} \mathbb{Z} & \text{si } n = 0, \\ 0 & \text{si } n \neq 0. \end{cases}$$

Proposición 1.9 Si X es conexo por arcos, entonces $H_0(X) \approx \mathbb{Z}$.

Definición 1.13 Se considera la secuencia

$$\dots \rightarrow C_n(X) \xrightarrow{\partial_n} C_{n-1}(X) \xrightarrow{\partial_{n-1}} \dots \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \xrightarrow{\varepsilon} \mathbb{Z} \rightarrow 0,$$

donde ε es la aplicación aumento definida por

$$\sum n_i T_i \mapsto \sum n_i.$$

Los grupos de homología $\tilde{H}_n(X)$ de este complejo de cadenas se llaman los *grupos de homología reducida de X* y se tiene que $H_n(X) \approx \tilde{H}_n(X)$, para $n > 0$, y $H_0(X) \approx \tilde{H}_0(X) \oplus \mathbb{Z}$.

Si X es homeomorfo a Y , entonces es inmediato que $H_n(X)$ es isomorfo a $H_n(Y)$ (las cadenas en Y son las composiciones de las cadenas en X con el homeomorfismo).

Proposición 1.10 *Sea $X = \sqcup_i X_i$ la unión disjunta de los espacios topológicos X_i , $i \in I$. Entonces se tiene un isomorfismo $H_n(X) \approx \oplus_i H_n(X_i)$.*

Historicamente, el grupo fundamental apareció después de los grupos de homología, en la tentativa de obtener un invariante más fuerte que el primer grupo de homología. Estos dos invariantes están relacionados por el siguiente teorema.

Teorema 1.4 *Sea X un espacio topológico conexo por arcos. Entonces $H_1(X)$ es isomorfo a la abelianización de $\pi_1(X)$.*

Cada aplicación continua $f : X \rightarrow Y$ induce un homeomorfismo f_* entre los grupos de homología de X e Y . El objetivo ahora es como definimos este homomorfismo f_* .

Lo primero que hacemos es definir el homomorfismo $f_{\#} : Q_n(X) \rightarrow Q_n(Y)$ de manera que $f_{\#}(T) = f \circ T$. Este homomorfismo tiene las siguientes propiedades:

- Si T es degenerado, $f_{\#}(T)$ también lo es, por lo que $f_{\#}$ induce un homomorfismo de $C_n(X)$ en $C_n(Y)$ que denotamos por la misma $f_{\#}$.
- El siguiente diagrama es conmutativo:

$$\begin{array}{ccc} Q_n(X) & \xrightarrow{f_{\#}} & Q_n(Y) \\ \partial_n \downarrow & & \downarrow \partial_n \\ Q_{n-1}(X) & \xrightarrow{f_{\#}} & Q_{n-1}(Y) \end{array}$$

La prueba de esto en darse cuenta de que $f_{\#}(A_i T) = A_i(f_{\#} T)$ y que $f_{\#}(B_i T) = B_i(f_{\#} T)$.

De esto se sigue que el siguiente diagrama también es conmutativo:

$$\begin{array}{ccc} C_n(X) & \xrightarrow{f_{\#}} & C_n(Y) \\ \partial_n \downarrow & & \downarrow \partial_n \\ C_{n-1}(X) & \xrightarrow{f_{\#}} & C_{n-1}(Y) \end{array}$$

Como $f_{\#}$ lleva $Z_n(X)$ en $Z_n(Y)$ y $B_n(X)$ en $B_n(Y)$ para todo $n \geq 0$, induce un homomorfismo de los grupos cocientes denotado por

$$f_* : H_n(X) \rightarrow H_n(Y).$$

A esta aplicación se la conoce como *homomorfismo inducido* y cumple las propiedades formales habituales que permiten calcular los grupos de homología para un espacio topológico X :

Proposición 1.11 *El homomorfismo inducido posee las siguientes propiedades:*

1. $id_* = id$.
2. $(f \circ g)_* = f_* \circ g_*$.

Proposición 1.12 *Si dos aplicaciones continuas $f, g : X \rightarrow Y$ son homotópicas, entonces $f_* = g_*$.*

Corolario 1.1 *Si $f : X \rightarrow Y$ es una equivalencia homotópica, entonces $f_* : H_n(X) \rightarrow H_n(Y)$ es un isomorfismo.*

1.2.2. Homología relativa cuboidal.

A continuación se presenta la homología relativa cuboidal, donde la teoría sobre homomorfismos inducidos que acabamos de estudiar funciona de la misma manera. Para ello, se selecciona un subconjunto A de un espacio topológico de X y se identifican todas las n -cadenas que tenemos en A como un solo elemento.

Definición 1.14 Sea A un subespacio de un espacio topológico X . Se define el *grupo de las cadenas relativas* $C_n(X, A) = C_n(X)/C_n(A)$. Como el operador borde verifica que $\partial_n(C_n(A)) \subset C_{n-1}(A)$ podemos definir un *operador borde relativo* $\partial_n : C_n(X, A) \rightarrow C_{n-1}(X, A)$. Sea $Z_n(X, A) = \text{Ker } \partial_n$ y $B_n(X, A) = \text{Im } \partial_{n+1}$. Como se cumple que $\partial_n \circ \partial_{n+1} = 0$, se define como el *n -ésimo grupo de homología de X relativa a A* al grupo cociente $H_n(X, A) = Z_n(X, A)/B_n(X, A)$.

No es cierto que $H_n(X, A) = H_n(X)/H_n(A)$, ya que esta igualdad solo es cierta cuando A es un retracto por deformación de X .

A continuación introducimos el concepto de *secuencia exacta*, que es una propiedad de un complejo de cadenas que ayuda demostrar el teorema de Mayer-Vietoris que enunciaremos posteriormente.

Definición 1.15 Una *secuencia*

$$\cdots \rightarrow G_{n+1} \xrightarrow{\varphi_{n+1}} G_n \xrightarrow{\varphi_n} G_{n-1} \rightarrow \cdots$$

es *exacta* si $\text{Ker } \varphi_n = \text{Im } \varphi_{n+1}$, $\forall n$.

Una secuencia exacta $0 \rightarrow G \xrightarrow{\varphi} H \xrightarrow{\Psi} K \rightarrow 0$ se dice que es una *secuencia exacta corta*. Nótese que la secuencia $0 \rightarrow C_n(A) \xrightarrow{i} C_n(X) \xrightarrow{j} C_n(X, A) \rightarrow 0$ es exacta, donde i es la aplicación inclusión y j es la proyección sobre el cociente, por lo que es sencillo ver que el siguiente diagrama es conmutativo:

$$\begin{array}{ccccccc} & & 0 & & 0 & & 0 \\ & & \downarrow & & \downarrow & & \downarrow \\ \cdots & \longrightarrow & C_{k+1}(A) & \xrightarrow{\partial_{k+1}} & C_k(A) & \xrightarrow{\partial_k} & C_{k-1}(A) & \longrightarrow & \cdots \\ & & \downarrow i & & \downarrow i & & \downarrow i & & \\ \cdots & \longrightarrow & C_{k+1}(X) & \xrightarrow{\partial_{k+1}} & C_k(X) & \xrightarrow{\partial_k} & C_{k-1}(X) & \longrightarrow & \cdots \\ & & \downarrow j & & \downarrow j & & \downarrow j & & \\ \cdots & \longrightarrow & C_{k+1}(X, A) & \xrightarrow{\partial_{k+1}} & C_k(X, A) & \xrightarrow{\partial_k} & C_{k-1}(X, A) & \longrightarrow & \cdots \\ & & \downarrow & & \downarrow & & \downarrow & & \\ & & 0 & & 0 & & 0 & & \end{array}$$

Con esto, se tiene el siguiente resultado.

Proposición 1.13 La aplicación $\partial_* : H_n(X, A) \rightarrow H_{n-1}(A)$ definida por $\partial_*[z] = [\partial_n(w)]$, con $j(w) = z$, es un homomorfismo que llamamos *homomorfismo de conexión*.

Es fácil demostrar que la secuencia

$$\cdots \rightarrow H_n(A) \xrightarrow{i_*} H_n(X) \xrightarrow{j_*} H_n(X, A) \xrightarrow{\partial_*} H_{n-1}(A) \rightarrow \cdots$$

es exacta (con i_* y j_* las aplicaciones de inclusión y de proyección, respectivamente, sobre los grupos de homología). Tan solo debemos estudiar la exactitud de cada grupo del complejo de cadenas y aplicar lo que acabamos de ver sobre homología relativa. Esto también es cierto para cuando se consideran grupos de homología reducida.

El teorema de Van Kampen da el grupo fundamental de $A \cup B$ en términos de los grupos fundamentales de A , B y $A \cap B$. La secuencia de Mayer-Vietoris es análogo a este teorema para la homología.

Llamamos *escisión* a la aplicación inclusión $(X \setminus U, A \setminus U) \rightarrow (X, A)$ que induce un isomorfismo

$$H_n(X \setminus U, A \setminus U) \rightarrow H_n(X, A), \quad \forall n.$$

Consideramos la terna (X, X_1, X_2) ordenada de espacios donde X_1 y X_2 son subespacios de X . Tenemos las funciones de inclusión

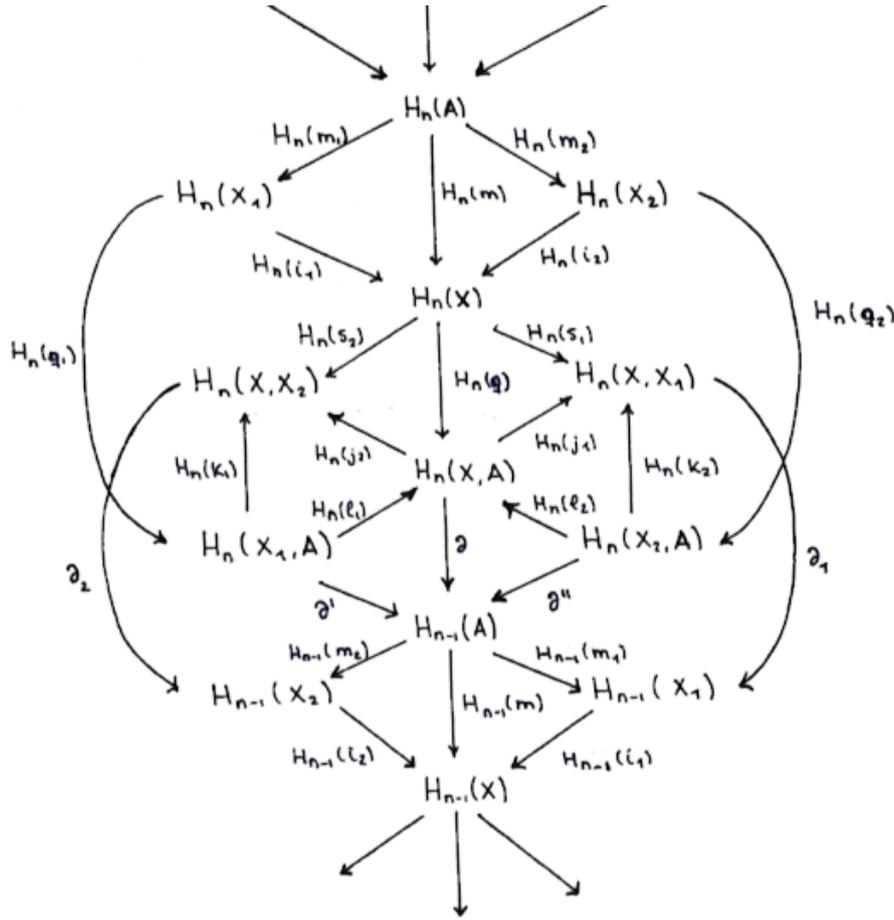
$$K_i : (X_i, X_1 \cap X_2) \rightarrow (X_1 \cup X_2, X_j), \quad \forall i, j \in \{1, 2\} \text{ con } i \neq j,$$

obtenidas por la escisión de $X_2 \setminus (X_1 \cap X_2)$, $X_1 \setminus (X_1 \cap X_2)$ de $X_1 \cup X_2$. Si ambas aplicaciones K_1 y K_2 son escisiones, se dice que la terna es *exacta* (o *propia*). Por tanto

$$H_n(K_i) : H_n(X_i, X_1 \cap X_2) \rightarrow H_n(X_1 \cup X_2, X_j) \quad \forall i, j \in \{1, 2\} \text{ con } i \neq j,$$

es un isomorfismo para todo n .

Supongamos que $X = X_1 \cup X_2$ y $A = X_1 \cap X_2$. Entonces, las diferentes aplicaciones de inclusión inducen el siguiente diagrama de homomorfismos, donde los triángulos y rectángulos son conmutativos.



El siguiente lema es puramente algebraico, usando sólo la conmutatividad y propiedades de exactitud del diagrama.

Lema 1.1 (Lema Hexagonal) *Asumiendo que la terna (X, X_1, X_2) es exacta, tenemos que*

$$0 = \partial' H_n(k_1)^{-1} H_n(s_2) + \partial'' H_n(k_2)^{-1} H_n(s_1)$$

Este lema, conocido como *lema hexagonal*, permite construir la *secuencia de Mayer-Vietoris*:

$$\dots \rightarrow H_n(A) \xrightarrow{\Psi} H_n(X_1) \oplus H_n(X_2) \xrightarrow{\Phi} H_n(X) \xrightarrow{\Delta} H_{n-1}(A) \rightarrow \dots$$

donde

$$\begin{aligned} \Psi(a) &= (H_n(m_1)(a), -H_n(m_2)(a)), \\ \Phi(x_1, x_2) &= H_n(i_1)(x_1) + H_n(i_2)(x_2), \\ \Delta &= \partial'' H_n(k_2)^{-1} H_n(s_1) = -\partial' H_n(k_1)^{-1} H_n(s_2). \end{aligned}$$

Teorema 1.5 *Si (X, X_1, X_2) es una terna exacta, entonces la secuencia de Mayer-Vietoris es exacta.*

La demostración de este teorema, así como el lema anterior, pueden encontrarse en [3]. Pero esta no es la única forma de demostrarlo. Otra manera de demostrar el Teorema de Mayer-Vietoris es a través del *Lema de Barratt-Whitehead* o del *Lema de la serpiente* (ambos lemas pueden consultarse en [19] y [22], respectivamente).

1.2.3. Subdivisiones y Escisión en complejos cuboidales.

En esta sección definimos como se forman los complejos cuboidales y como podemos subdividirlos. Después enunciamos el teorema de escisión.

Complejos cuboidales.

La construcción de estos complejos se basa en el “pegado de los bordes de unos cubos con otros”. Por ejemplo, el cubo unidad I^3 se forma a partir del producto de cuatro 2-cubos I^2 .

Para pegar n -cubos entre si usamos el álgebra de Boole con operaciones de unión e intersección teniendo así que:

1. La unión de dos cubos se lleva a cabo sobre “caras k -dimensionales enteras compartida”.
2. La intersección de dos cubos es “trivial” (vacía o él mismo), a lo largo de una cara común, a lo largo de una arista común o es un vértice común.

Como sabemos, cada n -cubo hereda una orientación inducida por la orientación positiva (en el sentido contrario a las agujas del reloj) con referencia a una base ortonormal ordenada de la esfera n -dimensional a través del homeomorfismo natural. El pegado de cubos orientados a lo largo de una cara k -dimensional o arista común presenta dos casos:

- (a) Si la cara o arista común tiene orientaciones opuestas, el pegado es orientable.
- (b) Si la cara o arista común tiene orientaciones comunes, el pegado es no-orientable.

Definición 1.16 *Un complejo cuboidal K es el resultado del pegado de cubos con las condiciones anteriores relacionadas con su unión e intersección.*

La característica de Euler-Poincaré tendrá un papel muy importante como invariante topológico. Se define de la siguiente manera:

Definición 1.17 *El i -ésimo número de Betti b_i de un complejo cuboidal K es el rango de la parte libre de $H_i(X)$.*

Definición 1.18 *La característica de Euler-Poincaré $\chi(K)$ de un complejo cuboidal K es la suma alternada de los números de Betti*

$$\chi(K) = \sum_{i=0}^n (-1)^i b_i$$

Subdivisiones y escisión.

Es importante introducir técnicas de subdivisión de cubos singulares para obtener un refinamiento de las mallas; ello permite una mejor adaptación a la forma de objetos complejos. Para ello, se procede de forma similar al caso simplicial, pero ahora introduciendo la subdivisión de un cubo. La idea más simple es subdividir el cubo I^n en 2^n cubos de lado $1/2$ mediante los hiperplanos mediatrices $x_i = 1/2$, con $i = 1, \dots, n$.

Definición 1.19 *Sea $\nu_n = \{\underline{e} \in \mathbb{R}^n | e_i = 0 \text{ o } 1 \forall i\}$ el conjunto de todos los vértices del I^n cubo. Para todo n -cubo singular T y $\underline{e} \in \nu_n$ se define la aplicación*

$$F_{\underline{e}}(T) : I^n \rightarrow X$$

por

$$(F_{\underline{e}}T)(\underline{x}) = T(1/2(\underline{x} + \underline{e})), \quad \forall \underline{x} \in I^n.$$

Definimos así $Sd_n : Q_n(X) \rightarrow Q_n(Y)$ el *operador de subdivisión* por

$$Sd_n(T) = \sum_{\underline{e} \in \nu_n} F_{\underline{e}}(T), \quad \forall n \geq 1.$$

Si T es el 0-cubo singular, $Sd_0(T) = T$. Este operador que acabamos de definir, junto con algunas propiedades que cumple, proporciona una herramienta fundamental para demostrar el *teorema de escisión* que enunciamos a continuación:

Teorema 1.6 *Sea A un subespacio de un espacio topológico X y sea U tal que $\bar{U} \subset \text{int } A$. Entonces la inclusión $(X \setminus U, A \setminus U) \rightarrow (X, A)$ induce un isomorfismo*

$$H_n(X \setminus U, A \setminus U) \rightarrow H_n(X, A), \quad \forall n.$$

1.2.4. Cohomología cuboidal.

Antes de enunciar la cohomología cuboidal introducimos elementos básicos de la teoría de categorías para facilitar las relaciones entre diferentes enfoques y su posterior unificación.

La teoría de categorías trata de axiomatizar diversas estructuras como una sola a través del uso de objetos y morfismo u homomorfismo.

Definición 1.20 *Decimos que \mathcal{A} es una *categoría* si consta de:*

- a) *Una clase de objetos de \mathcal{A} , que denotamos por $Ob(\mathcal{A})$.*
- b) *$\forall A, B \in Ob(\mathcal{A})$, un conjunto cuyos elementos son los morfismo (u homomorfismos) de $f : A \rightarrow B$. es el conjunto que se denota por $Mor(A, B)$ (o $Hom(A, B)$).*
- c) *Una operación binaria que es la composición de morfismos u homomorfismos que satisface dos propiedades:*
 1. *Asociatividad: $(f \circ g) \circ h = f \circ (g \circ h)$.*
 2. *Existencia de identidad: Un elemento de $Hom(A, A)$ tal que*

$$1_A \circ g = g; \quad f \circ 1_A = f,$$

con $g : C \rightarrow A$ y $f : A \rightarrow B$.

Una categoría se puede relacionar con otra mediante *funtores*. Un *functor* asocia a cada objeto de una categoría un objeto de la otra, y a cada aplicación (o morfismo) de la primera una aplicación (o morfismo) de la segunda. Recordemos que los morfismos (u homomorfismos) son las aplicaciones entre estructuras matemáticas que preservan la estructura interna.

Los grupos de homología con coeficientes en el grupo R se obtienen de la siguiente manera:

1. Tenemos el grupo de cadenas cuboidales $C(X) = \{C_n(X), \partial_n\}$.
2. Aplicamos el funtor $\otimes R$ para obtener así un nuevo grupo de cadenas:

$$C(X) \otimes R = C(X; R).$$

3. Construimos así los grupos de homología de este grupo de cadenas:

$$H_n(X; R) = H_n(C(X; R)).$$

En caso de que no se especifique el grupo R de la homología, se toma el grupo de los números enteros \mathbb{Z} .

Los *grupos de cohomología de X con coeficientes en R* se construye de una manera similar a lo que acabamos de hacer, solo que en el segundo paso aplicamos el funtor $\otimes Hom(\quad, R)$ en lugar de $\otimes R$. De esta manera, definimos el grupo de las *co-cadenas n -dimensionales* como

$$C^n(X) := \text{Hom}(C_n(X), R) = \text{Hom}(C_n(X), \mathbb{Z}) \otimes_{\mathbb{Z}} R.$$

Entonces, si cada n -cadena tenía una representación única de la forma $u = \sum_i n_i T_i$, una co-cadena también de manera que si $\varphi \in C^n(X)$, $\varphi(u) = \sum_i n_i \varphi(T_i)$.

En el ambiente que estamos trabajando exponemos dos resultados sin demostración que nos facilitan ciertos aspectos.

Proposición 1.14 *Para toda variedad compacta, orientable y conexa todas las teorías de (co)homología coinciden.*

Proposición 1.15 *Para toda variedad compacta, orientable y conexa la homología es dual de la cohomología en grado complementario.*

Las demostraciones de estos resultados pueden encontrarse en [5].

Supongamos $f : X \rightarrow Y$ induce una aplicación $C_n(X) \rightarrow C_n(Y)$. Por dualidad se construye f^* tal que

$$\varphi \in \text{Hom}(Y, R) \mapsto f^* \varphi := \varphi \circ f \in \text{Hom}(X, R).$$

Por tanto, f induce una aplicación $f : C^n(Y) \rightarrow C^n(X)$. Entonces $\forall \varphi \in C^n(Y)$ con $\varphi : C_n(Y) \rightarrow R$ le corresponde $C^n(\varphi) := \varphi \circ C_n(f)$ una aplicación lineal entre grupo de n -cocadenas.

Dualizando el operador borde ∂_n obtenemos el operador coborde, δ_n , definido de la siguiente manera:

Consideramos el siguiente complejo de cocadenas

$$\dots \rightarrow C^{n-1}(X) \xrightarrow{\delta_{n-1}} C^n(X) \xrightarrow{\delta_n} C^{n+1}(X) \rightarrow \dots$$

el operador coborde se define como

$$\delta_n := \varphi \circ \partial_{n+1}, \quad \varphi \in C^n(X).$$

En el marco del Análisis este operador se conoce como “operador adjunto”, ∂^* , mientras que en el marco de la Geometría se conoce como “operador diferencial”, d .

Es fácil comprobar que el operador coborde verifica las mismas propiedades que el operador borde gracias a las cuales hemos definido los grupos de homología. De forma paralela, estas propiedades, desde el punto de vista de la cocadenas, permiten definir los grupos de cohomología.

Definición 1.21 *Sea $B^n(X) := \text{Im } \varphi_{n-1}$ los n -cobordes de X y $Z^n(X) := \text{Ker } \varphi_n$ los n -cociclos de X . Teniendo que $B^n(X) \subseteq Z^n(X)$, definimos el n -ésimo grupo de cohomología de X como el grupo cociente*

$$H^n(X, R) := \frac{Z^n(X, R)}{B^n(X, R)}.$$

Como $H_n(X, R)$ es un invariante topológico de la clase de homotopía de X , $H^n(X, R)$ también lo es.

De forma paralela a la teoría de la homología, si $f : A \rightarrow B$ entonces induce una aplicación entre los grupos de cohomología $H^n(f) : H^n(Y) \rightarrow H^n(X)$.

La cohomología permite identificar la solubilidad o la obstrucción a resolver cualquier sistema de ecuaciones relativas a cualquier tipo de campos (escalares, vectoriales o tensoriales) gracias a la dualidad entre ciclos y cociclos. La clave para esta dualidad está en la estructura multiplicativa de la cohomología que se expresa en términos de productos (exterior, interior, etc...). En nuestro contexto, se formalizan el producto exterior y el interior mediante el cup-producto, \cup , y el cap-producto, \cap , respectivamente. Estos dos operadores nos ayudarán para formalizar los procesos de contracción y de expansión mencionados en diferentes lugares de la Memoria.

Definición 1.22 *Dados dos co-cubos $c \in C^p(X)$ y $d \in C^q(X)$, definimos el **cup-producto**, $c \cup d$, de la forma siguiente*

$$[c \cup d, T^{p+q}] := [c, i_p T^{p+q}][d, i_q T^{p+q}]$$

donde $i_p T^{p+q}$ ($i_q T^{p+q}$, resp.) es el p -cubo (q -cubo, resp.) correspondiente a la tapa inferior (superior, resp.) correspondiente a los vértices A_0, \dots, A_p (A_p, \dots, A_{p+q} , resp.).

Otra manera de definir el cup-producto sería de la manera siguiente a través de la aplicación

$$\begin{aligned} C^p(X) \times C^q(X) &\xrightarrow{\cup} C^{p+q}(X) \\ (c, d) &\longmapsto c \cup d : C_{p+q}(X) \rightarrow R \end{aligned}$$

con $(c \cup d)(T_{p+q}) := c(i_{0, \dots, p} T_{p+q})d(i_{p, \dots, p+q} T_{p+q})$ donde el primer elemento del producto genera un p -cubo y el otro genera un q -cubo.

El cup-producto dota a $C^*(X)$ de estructura multiplicativa compatible con el operador coborde, verificando que δ es una anti-derivación (en el sentido de Leibnitz), es decir,

$$\delta(c \cup d) = (\delta c) \cup d + (-1)^p c \cup \delta d.$$

Este producto no es conmutativo. Se verifica que

$$c \cup d = (-1)^{pq} d \cup c.$$

Definición 1.23 *Definimos el **cap-producto** como la aplicación*

$$\begin{aligned} C_{p+q}(X) \times C^p(X) &\xrightarrow{\cap} C_q(X) \\ (z, c) &\longmapsto z \cap c \end{aligned}$$

En el contexto topológico, el cap-producto se interpreta como la contracción de c a lo largo de las “direcciones de z ”. En el contexto diferencial o analítico corresponden a la integración de una p -forma con respecto a las variables que aparecen en $c \in C^p(X)$. Además, el cap-producto es el operador adjunto del cup-producto, que se define asimismo como

$$[z \cap c, d] := [c \cup d, z].$$

1.3. Teoría básica de Morse.

La Teoría de Morse nos ayudará para construir los complejos celulares a través de funciones que llamamos *funciones de Morse* [2]. Las funciones de Morse que utilizamos para construir un complejo celular son completamente independientes, en el sentido de que la homología celular es isomorfa a la homología cuboidal. Un resultado importante en esta teoría y que será una herramienta muy importante en nuestro trabajo son las *desigualdades de Morse* (capítulo 3 de [15]) que veremos al final de esta parte ya que establecen una relación entre la forma del objeto que estamos intentando reconocer y las funciones en el objeto.

1.3.1. Funciones de Morse.

Denotamos por M una variedad diferenciable, es decir, de clase C^∞ . Si $p \in M$, siendo p un punto, denotamos por $T_p(M)$ al *espacio tangente en el punto p de la variedad M* . Si $f : M \rightarrow \mathbb{R}$ es una aplicación diferenciable (donde \mathbb{R} es otra variedad diferenciable), es decir, $f \in C^\infty$, tenemos que f induce una aplicación lineal entre los espacios tangentes $\partial f / \partial x : T_p(M) \rightarrow T_{f(p)}(\mathbb{R})$.

Definición 1.24 *Sea f una función real diferenciable en M . Decimos que un punto $p \in M$ es un **punto crítico** de f si la aplicación $\partial f / \partial x : T_p(M) \rightarrow \mathbb{R}$ es nula. Llamamos al valor $f(p)$ **valor crítico** de f .*

Supondremos que la aplicación $\phi : U \rightarrow \mathbb{R}^n$; $\phi(x) = (x_1, \dots, x_n)$ es una carta local, es decir

$$\frac{\partial}{\partial x_i} (f \circ \phi^{-1})(\phi(p)) = 0, \quad \forall i \in \{1, \dots, n\}.$$

Definición 1.25 Se dice que un punto crítico $p \in M$ es *no-degenerado* si la matriz Hessiana asociada a la función f en el punto p , $H_p(f)$, es no singular.

Definición 1.26 Toda función f diferenciable en una variedad M cuyos puntos críticos son no-degenerados es una *función de Morse*.

Si tenemos $u, v \in T_p(M)$, siendo p un punto crítico, entonces u y v tienen extensiones u' y v' al campo de vectores ya que podemos interpretar $H_p(f)$ como un funcional bilineal simétrico

$$H_p(f) : T_p(M) \times T_p(M) \rightarrow \mathbb{R}.$$

Definición 1.27 Se define *índice de p de un funcional bilineal F en un espacio de vectores V* como la dimensión del subespacio de V en el que F es definida negativa, es decir, el número de autovalores negativos de F .

Definición 1.28 Se define *nulidad de un funcional bilineal F en un espacio de vectores V* como la dimensión del subespacio de V que consiste en todos los $u \in V$ tales que $F(u, v) = 0$, $\forall v \in V$.

El lema de Morse nos permite describir el comportamiento de la función f en p a través de su índice.

Lema 1.2 (Lema de Morse) Sea p un punto crítico de f no-degenerado. Sea $\phi(x) = (x_1, \dots, x_n)$ una carta local en un entorno ξ de p con $x_i(p) = 0$, $\forall i \in \{1, \dots, n\}$. En estas condiciones se tiene que

$$(f \circ \phi^{-1})(x_1, \dots, x_n) = f(p) - x_1^2 - \dots - x_\lambda^2 + x_{\lambda+1}^2 + \dots + x_n^2,$$

donde λ es el índice de p .

Terminamos esta sección dando un resultado que nos ayudará en las construcciones teóricas futuras.

Definición 1.29 El *grupo uniparamétrico de difeomorfismos de una variedad M* induce una función diferenciable

$$\Upsilon_t : M \rightarrow M, \quad \forall t \in \mathbb{R}$$

que cumple:

1. La función $\Upsilon_t(p) = \Upsilon(t, p)$ es un difeomorfismo.
2. $\Upsilon_{m+n} = \Upsilon_m \circ \Upsilon_n$, $\forall m, n \in \mathbb{R}$.

Con esta función Υ_t definimos un campo de vectores \mathbb{X} en M de la siguiente manera.

Definición 1.30 Para cada función $f : M \rightarrow \mathbb{R}$ de clase C^∞ definimos un *campo de vectores*

$$\mathbb{X}_p(f) := \lim_{t \rightarrow 0} \frac{(f \circ \Upsilon_t)(p) - f(p)}{t}.$$

Se suele decir que el campo de vectores \mathbb{X} está generado por el grupo Υ .

Lema 1.3 Un campo de vectores en M , el cual contiene un compacto K , genera un único grupo uniparamétrico de difeomorfismos de M .

Este resultado motiva la introducción a la noción de “flujo de una variedad” (abuso terminológico):

Definición 1.31 El *flujo de una variedad M* es una aplicación diferenciable

$$\varphi : M \times \mathbb{R} \rightarrow M$$

que satisface:

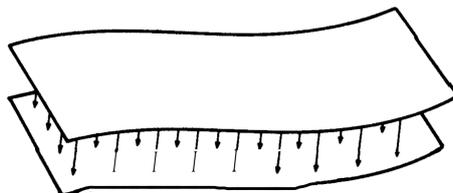
1. $\varphi(p, 0) = p$, $\forall p \in M$
2. $\varphi_{m+n} = \varphi_m \circ \varphi_n$, $\forall m, n \in \mathbb{R}$

1.3.2. Homotopía en términos de valores críticos.

De ahora en adelante definimos $M_a := \{p \in M \mid f(p) \leq a\}$ donde $f : M \rightarrow \mathbb{R}$.

Lema 1.4 Sean $a < b$ números reales tales que f no tiene valores críticos en el intervalo $[a, b]$. Entonces M_a es difeomorfo a M_b . Además, M_a es un retracto por deformación de M_b , por lo que la aplicación inclusión $M_a \hookrightarrow M_b$ es una equivalencia homotópica.

La idea de esta demostración consiste en “pegar” M_b con M_a a lo largo de las trayectorias ortogonales a las hipersuperficies $f = k \in \mathbb{R}$.



El siguiente resultado es central en esta teoría, pues justifica los procedimientos de pegado en la variedad para pasar de un tipo de homotopía a otra conservando la estructura interna.

Teorema 1.7 Sea $f : M \rightarrow \mathbb{R}$ una función de Morse y sea λ el índice de $p \in M$ con $f(p) = c$. Supongamos que $f^{-1}[c - \epsilon, c + \epsilon]$ es compacto y no contiene ningún otro punto crítico distinto de p . Por tanto se tiene que el tipo de homotopía de $M_{c+\epsilon}$ y de $M_{c-\epsilon}$ con una λ -célula es el mismo.

Demostración: Una idea de la demostración de este teorema está ilustrada en la Figura 1.11. La región $M_{c-\epsilon}$ es la más sombreada. Introducimos una nueva función $F : M \rightarrow \mathbb{R}$ que coincide con la función f , salvo en un pequeño entorno de p donde $F < f$. Entonces la región $F^{-1}(-\infty, c - \epsilon]$ consiste en la región $M_{c-\epsilon}$ junto con la región H cerca de p . Eligiendo una célula adecuada $e^n \subset H$, un argumento directo (contrayendo a lo de las líneas horizontales) muestra que $M_{c-\epsilon} \cup e^n$ es un retracto por deformación de $M_{c-\epsilon} \cup H$. Finalmente, aplicando el **Lema 1.4** a la función F y a la región $F^{-1}[c - \epsilon, c + \epsilon]$ vemos que $M_{c-\epsilon} \cup H$ es un retracto por deformación de $M_{c-\epsilon}$.

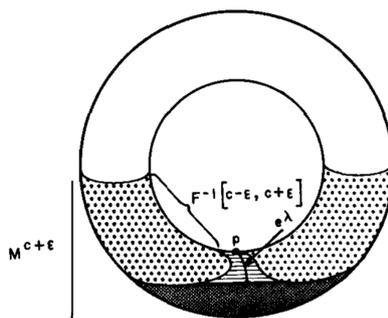


Figura 1.11: Toro \mathbb{T}^2

□

Podemos explicar el pegado de una n -célula como sigue:

Sea X un espacio topológico, y sea $e^n := \{\underline{x} \in \mathbb{R}^n \mid \|\underline{x}\| \leq 1\}$ una n -célula. Denotamos al borde de esta célula mediante $\dot{e}^n = \mathbb{S}^{n-1} := \{\underline{x} \in \mathbb{R}^n \mid \|\underline{x}\| = 1\}$. Si $g : \mathbb{S}^{n-1} \rightarrow X$ es una función continua, denotamos por $X \cup_g e^n$ al resultado de identificar cada $\underline{x} \in \mathbb{R}^n$ con $g(\underline{x}) \in X$. La idea consiste en “pegar” a lo largo del borde de la n -célula a X obteniendo así una nueva estructura. Por ello, a la función g se conoce por *función de pegado*. Notese que para el caso de $n = 0$ se obtiene la unión disjunta de X con un punto, ya que $\mathbb{S}^{-1} = \emptyset$.



Figura 1.12: Pegado de una 1-célula

A continuación se dan dos lemas que sirven para demostrar el **Teorema 1.8** que se enuncia al final de esta parte.

Lema 1.5 (Whitehead) Sean g_1 y g_2 funciones homotópicas de la esfera \mathbb{S}^{n-1} en X . La función identidad de X se extiende a la equivalencia homotópica

$$K : X \cup_{g_1} e^n \rightarrow X \cup_{g_2} e^n.$$

Lema 1.6 Sea $g : \mathbb{S}^{n-1} \rightarrow X$ una función de pegado. Cualquier equivalencia homotópica $f : X \rightarrow Y$ se extiende a la equivalencia homotópica

$$F : X \cup_g e^n \rightarrow Y \cup_{f \circ g} e^n.$$

Aplicando los dos resultados precedentes se tiene:

Teorema 1.8 Si $f : M \rightarrow \mathbb{R}$ es una función de Morse y cada M_α es compacto entonces se tiene que M tiene el mismo tipo de homotopía de un complejo celular con una λ -célula por cada punto crítico de índice λ .

1.3.3. Desigualdades de Morse.

Los grupos de homología $H_n(X)$ de un complejo celular son grupos abelianos, por lo que se pueden expresar como

$$H_n(X) \simeq \mathbb{Z} \oplus \cdots \oplus \mathbb{Z} \oplus \mathbb{T},$$

donde \mathbb{T} es un grupo abeliano que contiene algunos elementos finitos y es conocido como la torsión de $H_n(X)$. El resto de la expresión es conocida como la parte libre de $H_n(X)$. El número de copias de \mathbb{Z} corresponde al rango de $H_n(X)$ que es lo que hemos definido anteriormente como el n -ésimo número de Betti y en esta parte juega un papel muy importante.

Teorema 1.9 Sea M una variedad diferenciable y $f : M \rightarrow \mathbb{R}$ una función de Morse. Sean c_n el número de puntos críticos de índice n y b_n el n -dimensional número de Betti de M . Entonces se tiene que

$$c_n \geq b_n.$$

Esta desigualdad es conocida como *Desigualdad Débil de Morse*. En particular, si $b_n > 0$ entonces la función de Morse tiene por lo menos un punto crítico de índice n . Nótese que el número de Betti no depende de la función de Morse.

Las siguientes desigualdades son conocidas como *Desigualdades fuertes de Morse* o simplemente *Desigualdades de Morse* y dan una mayor relación entre las funciones que aplicamos en M y la forma de M .

Teorema 1.10 Para cualquier función de Morse $f : M \rightarrow \mathbb{R}$ donde M es una variedad diferenciable de dimensión k , se tiene que

$$1. \sum_{i=0}^N (-1)^{i+N} c_i \geq \sum_{i=0}^N (-1)^{i+N} b_i, \quad \forall N = 0, \dots, k.$$

$$2. \sum_{i=0}^k (-1)^i c_i = \sum_{i=0}^k (-1)^i b_i$$

Corolario 1.2 Si $c_{n+1} = c_{n-1} = 0$, entonces $b_n = c_n$ y $b_{n+1} = b_{n-1} = 0$.

Este último resultado es útil cuando se desea calcular los grupos de homología del espacio proyectivo complejo sin recurrir al **Teorema 1.8** [2].

1.3.4. Reconocimiento bajo información incompleta.

En las aplicaciones prácticas, habitualmente se desconoce la naturaleza de la variedad M . Sólo se tiene información parcial sobre la parte visible o detectable a partir de los sensores. Además esta información puede aumentar o disminuir en términos de oclusiones parciales debidas a la localización (posición y orientación) de los dispositivos de captura de información con respecto a los objetos (modelables inicialmente como PS-variedades) o tareas a reconocer (modelables en términos de campos).

En una primera aproximación, los objetos se modelan como PS-variedades (aproximables mediante PL-espacios topológicos), mientras que las tareas se modelan en términos de PS-campos (escalares, vectoriales o tensoriales). En esta sección, sólo se abordan PS-modelos M con funciones de Morse $f : M \rightarrow \mathbb{R}$ definidas sobre los modelos.

A diferencia de la Teoría de Morse Clásica donde se supone que M es conocida, aquí se construye una representación a partir de la parte visible. Por ello, aparecen singularidades asociadas al lugar discriminante $\Delta\pi$ de la proyección π desde la localización de la cámara. Dicho lugar se define como el lugar de anulación de π y de $d\pi$. El soporte $\text{supp}(\Delta\pi)$ es la intersección del cono tangente desde la localización de la cámara con el objeto, en función del campo de visibilidad para el dispositivo de captura. Así, p.e., para una superficie S el lugar discriminante está dado por una curva alabeada llamada *contorno aparente* que se proyecta en el plano de imagen sobre una silueta cambiante con el movimiento de la cámara.

Por ello, la representación puede incluir aparición y desaparición de componentes, dependiendo de la geometría del cono de visibilidad. Estos fenómenos modifican la descomposición celular del objeto en función de las singularidades de la aplicación proyección dada en este caso por la profundidad de cada objeto con respecto a la cámara. Para adaptar este enfoque a la Teoría Clásica de Morse, sólo consideramos un número finito de puntos críticos (extremos con respecto a coordenadas píxel en la imagen, p.e.) situados sobre cada componente de la imagen del lugar discriminante.

Las desigualdades de Morse presentadas en el apartado anterior permiten relacionar los caracteres diferenciales asociados al número de puntos críticos con invariantes topológicos de las descomposiciones celulares. En la segunda sección del **Capítulo 2** se introducen los invariantes homológicos para controlar la variabilidad de los complejos celulares asociados.

1.4. Elementos de fibrados vectoriales.

Los fibrados vectoriales (y sus generalizaciones a fibraciones) proporcionan estructuras sencillas (desde el punto de vista topológico) que permiten “pegar” datos locales de estructuras para obtener un objeto global. Localmente, son isomorfos a un producto cartesiano de un abierto U de la variedad base por una fibra F ; a dichos abiertos se les llama abiertos de trivialización.

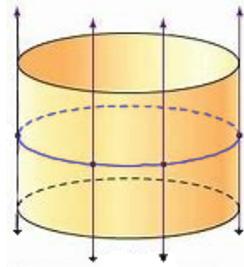
Un primer ejemplo de estructuras superpuestas a una variedad M está dado por el “pegado” de espacios tangentes $T_p(M)$ para cada $p \in M$ que da lugar al fibrado tangente $\tau_M = (TM, \pi, M, F)$; los campos vectoriales tangentes dados localmente por $\sum f_i \partial/\partial x_i$ son las secciones locales $s : U \rightarrow TU$ de dicho fibrado tangente verificando que $\pi \circ s = 1_U$. Los fibrados vectoriales proporcionan una “linealización global” que facilita el estudio de las propiedades de las variedades M . Una referencia clásica sobre esta teoría es [14].

Las propiedades fundamentales de los fibrados (o con más generalidad, las fibraciones), son el carácter localmente trivial (expresión topológica como producto), es decir, la C^r -equivlaencia $\pi^{-1}(U) \sim_{C^r} U \times F$ que se restringe a un isomorfismos entre la fibra genérica F y la fibra específica $F_p := \pi^{-1}(p)$ en cada punto $p \in M$. En el caso más simple del fibrado vectorial tangente, se tiene un isomorfismo como espacios vectoriales entre $T_p(M)$ y \mathbb{R}^m . Este isomorfismos permite describir y comparar datos lineales en fibras en términos de grupos estructurales de fibrados. La situación más frecuente corresponde a grupos clásicos que son subgrupos o grupos cociente del grupo lineal general $GL(m; \mathbb{R})$ que conservan una forma cuadrática (asociada a una métrica), una forma bilineal (asociada a un producto interior) o, con más generalidad, una forma multilineal.

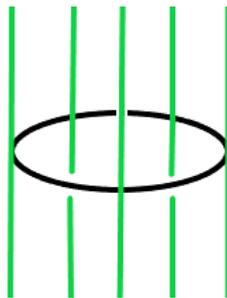
En esta sección, antes de comenzar las nociones básicas sobre fibrados vectoriales, comenzamos presentando la idea de fibrado para mejorar la comprensión de esta parte.

1.4.1. Nociones básicas de Fibraciones.

En esta subsección se muestra una versión geométrica de la noción de fibración que es de utilidad desde el punto de vista de la Geometría Diferencial. Para ello comenzamos con un sencillo ejemplo que utilizamos para definir la noción de fibración: Consideremos el cilindro infinito que tiene como base la circunferencia \mathbb{S}^1 con su embebimiento natural como se muestra en la siguiente figura.



Cada punto p del cilindro se representa mediante las coordenadas $p = (a, b)$ con $a \in \mathbb{S}^1$ y $b \in \mathbb{R}$. Este cilindro es una variedad diferenciable que se puede construir mediante el pegado de rectas paralelas reales \mathbb{R} en los puntos de de la circunferencia \mathbb{S}^1 . Por ello podemos denotar al cilindro como el producto cartesiano $E = \mathbb{S}^1 \times \mathbb{R}$ que constituye el espacio total de la variedad. Entonces podemos presentar el cilindro por tres elementos: $E \equiv$ “Espacio total”, $\mathbb{S}^1 \equiv$ “Espacio base” y $\mathbb{R} \equiv$ “Fibra”.



En la definición posterior de fibración se extiende la fibra sobre $B = \mathbb{S}^1$ a espacios más generales que la recta \mathbb{R} . Además consideramos el homomorfismo de proyección

$$\begin{aligned} \pi : E &\longrightarrow \mathbb{S}^1 \\ p = (a, b) &\longmapsto a \end{aligned}$$

Notese que, si tenemos otro punto $q = (c, d) \in E$, las imágenes inversas $\pi^{-1}(a) \simeq \mathbb{R}_a$ y $\pi^{-1}(c) \simeq \mathbb{R}_c$ son diferentes entre sí, pero ambas son isomorfas a \mathbb{R} .

La descripción precedente proporciona un ejemplo sencillo geométrico de la noción de producto cartesiano y motiva la siguiente definición.

Definición 1.32 Llamamos *fibración localmente trivial* ó simplemente *fibración* a una 5-upla $\xi = (B, F, E, \pi, G)$ con las siguientes características:

1. Un espacio topológico conexo B (en ocasiones una variedad) llamado “Espacio base”.
2. Un espacio topológico F (en ocasiones una variedad) llamado “Fibra general”.
3. Un espacio topológico E (en ocasiones una variedad) llamado “Espacio total”.
4. Una aplicación continua y suprayectiva $\pi : E \rightarrow B$ que es localmente trivial. A la imagen inversa $\pi^{-1}(p)$ de $p \in B$ respecto de B se le llama “Fibra” de ξ en p , se denota también mediante E_p ¹ y se tiene un difeomorfismo $\pi^{-1}(p) \simeq F$ con la fibra genérica.

¹En algunos libros se denota mediante E a la fibración, dando lugar a una confusión terminológica

5. Un recubrimiento $\mathcal{U} = \{U_i\}_{i \in \Lambda}$ por abiertos de B verificando que para cada $i \in \Lambda$ existe un difeomorfismo

$$\phi_i : U_i \times F \rightarrow \pi^{-1}(U_i)$$

que se llama “trivialización local” de ξ (suele denotarse también por $(\pi, \psi_i)^{-1}$ donde en este caso $\pi : \pi^{-1}(U_i) \rightarrow U_i$ y $\psi_i : \pi^{-1}(U_i) \rightarrow F$) tal que $(\pi \circ \phi_i)(p, f) = p$ para cualquier $p \in U_i \subset B$ y $f \in E_p$.

6. Sobre cada intersección $U_{ij} = U_i \cap U_j \neq \emptyset$, si $\psi_{(i,p)} : \pi^{-1}(p) \rightarrow F$ denota la restricción de la aplicación de trivialización a la fibra E_p relativa a U_i , entonces cada aplicación

$$g_{ij}(f) := (\psi_{(j,p)} \circ \psi_{(i,p)}^{-1})(f)$$

es un difeomorfismo en F , es decir, es un elemento del grupo de C^r -equivalencias (difeomorfismos $\text{Diff}(F)$ en el caso diferenciable) de F . Por ello, ψ_i y ψ_j están relacionadas por

$$\psi_i(p, f) = \psi_j(p, g_{ij}(f))$$

y expresan el cambio en la expresión de un elemento f de la fibra $\pi^{-1}(p)$ debido al cambio de carta en la base. A dichas aplicaciones se les llaman “Funciones de transición” de la fibración.

7. Los elementos de $\text{Diff}(F)$ se pueden expresar como composición de las funciones de transición de una fibración topológica. Ello motiva la necesidad de dotar a este conjunto con una estructura de grupo dando lugar a la noción de “Grupo estructural” G ; en la categoría diferenciable, el grupo estructural G es equivalente a $\text{Diff}(F)$. Por ello, en ocasiones la fibra puede ser el grupo G ó un subgrupo suyo (en el ejemplo anterior del cilindro infinito, $G = \{1\}$).

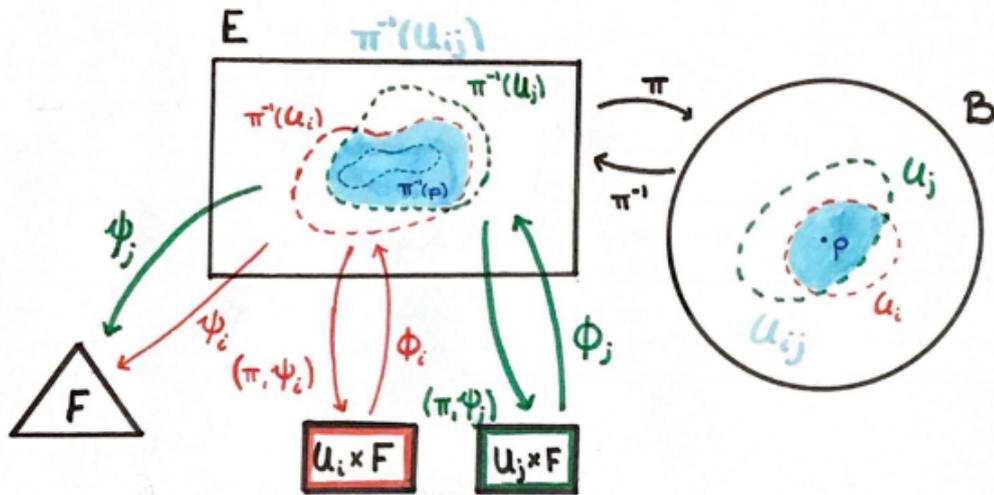


Figura 1.13: Diagrama de fibración

Los *fibrados vectoriales* son fibraciones en los que la fibra es un espacio vectorial n -dimensional sobre un cuerpo \mathbb{K} sobre el que actúa el grupo lineal general $G = GL(n; \mathbb{K})$ de forma natural. Nótese que la aplicación π dota a E con la topología cociente, es decir, $\tau_E = \{\pi^{-1}(U_i) | U_i \in \tau_B\}$.

El conjunto $\{(U_i, \phi_i)\}_{i \in \Lambda}$ se llama *Atlas de la fibración*.

Las *fibraciones* no tienen porqué tener una fibra continua; en el caso discreto pueden tener una cantidad infinita de puntos.

Algunos ejemplos de fibraciones son:

- La aplicación proyección $\pi_1 : B \times F \rightarrow B$ sobre la primera componente del producto que es una fibración globalmente trivial (como el ejemplo del principio del todo).

- La aplicación $f : \mathbb{S}^1 \rightarrow \mathbb{S}^1$ dada por $f_n(z) = z^n$ define una fibración cuya fibra en cada punto es el conjunto de raíces n -ésimas de la unidad. Nótese que esta fibración presenta puntos de ramificación correspondientes a los valores en los que se “solapan” varias raíces.

- La aplicación exponencial

$$\exp : \mathbb{R} \longrightarrow \mathbb{S}^1;$$

$$t \longmapsto e^{2\pi it},$$

es una fibración localmente trivial cuya fibra en cada punto es isomorfa a \mathbb{Z} .

- Consideramos la banda de Möbius $\mathcal{M} = [0, 1] \times [0, 1] / \sim$, donde $(t, 0) \sim (1 - t, 1)$. Sea $C = \{(1/2, s) \in \mathcal{M}\}$ la “circunferencia central” y consideramos la proyección

$$\pi : \mathcal{M} \longrightarrow C;$$

$$(t, s) \longmapsto (1/2, s).$$

Esta construcción proporciona el primer ejemplo de *fibración no-trivial* con fibra $F = [0, 1]$ y grupo estructural $G = \{1, -1\}$.

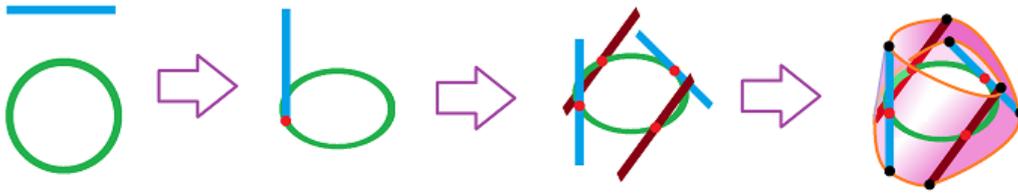


Figura 1.14: Construcción de la Banda de Möbius extendida como fibración

- Los espacios recubridores son otro ejemplo de fibración de uso común en Topología y en Geometría Algebraica.

Definición 1.33 Sean ξ y ξ' dos fibraciones. Llamamos *morfismos fibrados* a cualquier par de aplicaciones (α, β) con $\alpha : E \rightarrow E'$ y $\beta : B \rightarrow B'$ de manera que $\pi' \circ \alpha = \beta \circ \pi$, es decir, hace que el siguiente diagrama sea conmutativo

$$\begin{array}{ccc} E & \xrightarrow{\alpha} & E' \\ \pi \downarrow & & \downarrow \pi' \\ B & \xrightarrow{\beta} & B' \end{array}$$

Definición 1.34 Sea ξ una fibración y $U_i \in \mathcal{U}$ un abierto del espacio base de la fibración. El conjunto $\Gamma(U_i, E)$ de *secciones de E sobre U_i* es el conjunto de aplicaciones continuas $s : U_i \rightarrow E$ tales que $(\pi \circ s)(p) = p$, $\forall p \in U_i$, es decir, la composición de ambas funciones es equivalente a la aplicación identidad.

De este modo se establece una equivalencia entre el espacio base y su imagen a través de la aplicación s . Así, por ejemplo, si E está dada globalmente como una variedad producto $\mathbb{M} \times F$, entonces es posible describir un número de secciones funcionalmente independientes (es decir, con diferenciales linealmente independientes) igual a $\dim(F)$.

Las secciones de una fibración pueden tener un comportamiento bastante más complicado que el correspondiente a las secciones de un fibrado vectorial. Un ejemplo de esto es la fibración de Hopf $H : \mathbb{S}^3 \rightarrow \mathbb{S}^2$ con fibra \mathbb{S}^1 .

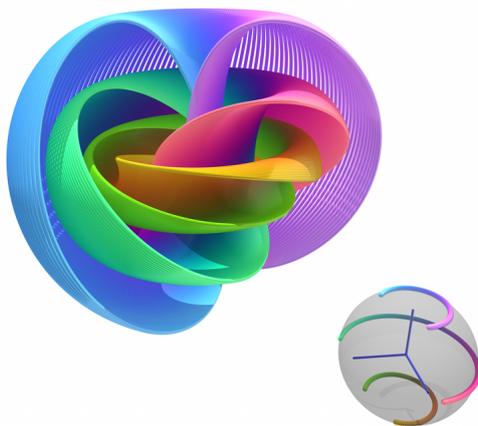


Figura 1.15: Fibración de Hopf asociada a un corte de la fibración

Las funciones de transición permiten “pegar datos” utilizando transformaciones sobre la fibra de cada punto $p \in B$, fibra que se visualiza de diferentes formas dependiendo del abierto U_i al que pertenece p . En el caso de fibrados vectoriales, como la fibra es un e.v. de dimensión finita $r = \text{rang}(\xi)$, las diferentes formas de visualizar la fibra están dadas por transformaciones regulares sobre $\pi^{-1}(p) \simeq \mathbb{R}^n$, es decir, por automorfismos de \mathbb{R}^n que se representan mediante elementos del grupo lineal general $GL(n; \mathbb{R})$. Por ello, las funciones de transición están dadas localmente por aplicaciones

$$g_{ij} : U_{ij} \rightarrow GL(n; \mathbb{R}),$$

es decir, no son propiamente funciones, pues toman valores en \mathbb{R}^n , sino en un grupo. Esta noción se extiende de forma natural a otros grupos diferentes del grupo lineal general y permite el “pegado” de datos locales utilizando transformaciones sobre la fibra.

Proposición 1.16 *Las funciones de transición de una fibración ξ verifican:*

1. $g_{ij} \circ g_{jk} = g_{ik}$, $\forall i, j, k \in \Lambda$, con $U_{ijk} \neq \emptyset$ (Transitividad).
2. $g_{ii} = 1_G$, $\forall i \in \Lambda$ (Identidad).

Las propiedades anteriores permiten comparar y clasificar fibraciones.

Corolario 1.3 *Si $i, j \in \Lambda$ con $U_{ij} \neq \emptyset$, entonces $g_{ij} = g_{ji}^{-1}$.*

Asimismo, las funciones de transición proporcionan herramientas para el “pegado” de datos locales sobre una estructura superpuesta a una variedad.

1.4.2. Enfoque funtorial.

De una forma intuitiva podría decirse que el enfoque funtorial consiste en estudiar de forma simultánea los objetos y los morfismos entre objetos. En el caso de esta sección, los objetos son fibrados vectoriales y los morfismos son aplicaciones fibradas.

El enfoque funtorial tiene interés para identificar propiedades formales que facilitan la clasificación salvo isomorfismo de los fibrados vectoriales (K -Teoría) y el cálculo de invariantes topológicos de los fibrados vectoriales. Estos invariantes (clase de cohomología, por ejemplo) sólo dependen de la clase de homotopía del morfismo, lo cual proporciona una motivación adicional para el estudio de aplicaciones clasificantes sobre espacios homogéneos (como la Grassmanniana, p.e.).

Definición 1.35 *Llamamos **fibrado vectorial sobre una variedad** M a una fibración sobre M tal que la fibra F es un \mathbb{K} -espacio vectorial r -dimensional. Llamamos **rango** y lo denotamos como $\text{rang}(\xi)$ o simplemente $r(\xi)$ de un fibrado vectorial ξ a la dimensión de la fibra general F como espacio vectorial. Diremos que un fibrado vectorial $\xi = (B, F, E, \pi, G)$ es trivial si E es equivalente a $M \times F$.*

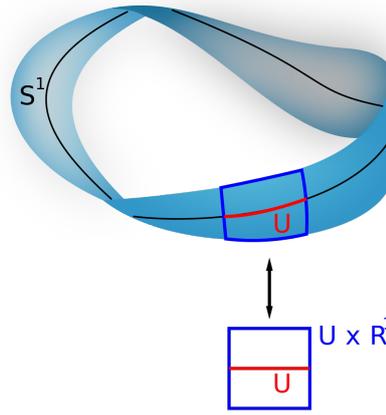


Figura 1.16: Banda de Möbius como fibrado vectorial

Definición 1.36 Llamamos *sección local del fibrado vectorial* ξ a cualquier aplicación $s : U_i \rightarrow E$ tal que $\pi \circ s = id_M$. Denotamos mediante $\Gamma(U_i, E)$ al conjunto de secciones locales definidas sobre U_i y mediante $\Gamma(M, E)$ a las secciones globales.

Definición 1.37 Dados dos fibrados vectoriales ξ y ξ' y una aplicación $\sigma : B \rightarrow B'$, llamamos *homomorfismos del fibrado ξ en ξ'* a cualquier aplicación $\Theta : E \rightarrow E'$ que hace el siguiente diagrama conmutativo

$$\begin{array}{ccc} E & \xrightarrow{\Theta} & E' \\ \pi \downarrow & & \downarrow \pi' \\ B & \xrightarrow{\sigma} & B' \end{array}$$

y cuya restricción a las fibras respectivas es un \mathbb{K} -homomorfismo, siendo \mathbb{K} el cuerpo base del espacio vectorial isomorfo a la fibra.

Dados dos fibrados vectoriales ξ y ξ' sobre un mismo espacio base B , denotamos al conjunto de homomorfismos de ξ en ξ' por $HOM(\xi, \xi')$. Este conjunto tiene asimismo estructura de fibrado vectorial con fibra isomorfa a $Hom_{\mathbb{K}}(F, F')$.

Frecuentemente denotaremos a los elementos de $HOM(\xi, \xi')$ mediante pares de la forma (Θ, σ) donde $\Theta : E \rightarrow E'$, $\sigma : B \rightarrow B'$ de modo que $\pi' \circ \Theta = \sigma \circ \pi$ y la restricción de Θ a cada fibra induce una aplicación lineal entre espacios vectoriales, es decir, $\Theta_p : \pi^{-1}(p) \rightarrow \pi'^{-1}(\sigma(p))$, ó si se prefiere, $\Theta | \pi^{-1}(p) \in Hom(\xi_p, \xi'_{\sigma(p)})$, $\forall p \in B$.

Definición 1.38 Dados dos fibrados vectoriales ξ y ξ' sobre un mismo espacio base B , diremos que ξ es isomorfo a ξ' si existe un homeomorfismo $E \rightarrow E'$ entre los espacios totales, cuya restricción a cada cifra induce un isomorfismo $\pi^{-1}(p) \simeq \pi'^{-1}$ como espacios vectoriales para cualquier $p \in B$.

Ejemplos importantes de fibrado vectorial de uso común en Geometría Diferencial están dados por el *fibrado tangente* y el *fibrado cotangente*.

Definición 1.39 El *Fibrado Tangente* τ_M de una variedad real m -dimensional M es una 4-tupla (TM, π, M, F) cuyo espacio total TM es la unión disjunta de los espacios tangentes $T_p(M)$ a M en cada $p \in M$, con espacio base M , aplicación proyección $\pi : TM \rightarrow M$ con $\pi(T_p(M)) = p$ y fibra genérica $F = \mathbb{R}^m$ isomorfa de forma no-canónica a $\pi^{-1}(p) := T_p(M)$. Dicha 4-tupla verifica que

- Para cada $p \in M$ existe un abierto de trivialización $U \subset M$ tal que $\pi^{-1}(U)$ es difeomorfo (de forma no-canónica) a $U \times \mathbb{R}^m$;
- El difeomorfismo anterior se restringe a un isomorfismo no-canónico entre la fibra $\pi^{-1}(p) := T_p(M)$ y la fibra genérica \mathbb{R}^m es isomorfa como espacio vectorial a $\mathbb{R} \langle \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_m} \rangle$ (evaluación de las derivaciones en el punto $p \in M$).

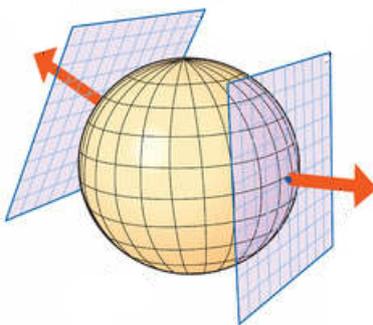


Figura 1.17: Ejemplo de fibrado tangente con espacio base \mathbb{S}^2 y fibras $T_p(\mathbb{S}^2)$

Definición 1.40 El Fibrado Cotangente Ω_M^1 de una variedad real m -dimensional M es una 4-tupla (T^*M, π, M, F) definido como el dual $\text{HOM}(\tau_M, \varepsilon_M^1)$ del fibrado tangente, donde $\varepsilon_M^1 := (M \times \mathbb{R}, \pi, M, \mathbb{R})$ el fibrado trivial de rango 1 sobre M . En este caso, la fibra genérica \mathbb{R}^m es isomorfa como espacio vectorial a $\mathbb{R} \langle dx_1, \dots, dx_m \rangle$ (evaluación de las diferenciales en el punto $p \in M$).

Las secciones del fibrado tangente τ_M son campos vectoriales sobre M , mientras que las secciones del fibrado cotangente Ω_M^1 son 1-formas diferenciales sobre M . Ambas son dos casos particularmente simples de tensores $t^{r,s}M$ de tipo (r, s) sobre M que se definen como las secciones del producto tensorial $\otimes^r \tau_M^* \otimes \otimes^s \tau_M$. En particular, los tensores de tipo $(0, 1)$ son campos vectoriales, los tensores de tipo $(1, 0)$ son 1-formas diferenciales y los de tipo $(2, 0)$ son métricas sobre M . La teoría sobre el campo de tensores se desarrollará con más detalle en la siguiente sección.

1.4.3. Álgebra lineal de fibrados vectoriales.

Las operaciones algebraicas que se efectúan en álgebra lineal con espacios vectoriales y homomorfismos se pueden definir también en fibrados vectoriales y homomorfismos fibrados, procediendo en cada punto de la base con las fibras.

Definición 1.41 Dada una aplicación $\sigma : B' \rightarrow B$ y un fibrado vectorial ξ , definimos el producto fibrado de B' y ξ como el conjunto de pares

$$\{(p, z) \in B' \times E \mid \sigma(p) = \pi(z)\}$$

al que denotamos por E_σ , dotado con la aplicación proyección natural π_σ que a cada par $(p, f) \in E_\sigma$ le asocia la primera componente $p \in B'$.

Haciendo $\Theta(p, z) := (\sigma(p), z)$ se obtiene un isomorfismo entre las fibras $\pi_\sigma^{-1}(p) \simeq \pi^{-1}(\sigma(p))$ como espacios vectoriales sobre el cuerpo base. Por último, sea V un abierto de trivialización local de ξ (de modo que $\pi^{-1}(V)$ es equivalente a $V \times F$) y hagamos $U := \sigma^{-1}(V)$ (para ello f debe ser al menos continua). Definimos entonces

$$\phi_U : U \times F_\sigma \rightarrow \pi_\sigma^{-1}(U) \mid \phi_U(p, z) := (p, \phi_V(\sigma(p), z))$$

Las aplicaciones así definidas dan un atlas de trivialización para σ , por lo que la imagen recíproca σ es un fibrado vectorial sobre B' .

Las nociones de orientabilidad y de orientación son asimismo importantes. Para analizar la orientabilidad de un fibrado vectorial se introduce una orientación sobre cada fibra y se evalúa si al desplazarse continuamente sobre la base sobre caminos cerrados (lazos), la orientación de las fibras correspondientes se mantiene o no al cerrar el lazo.

Definición 1.42 Diremos que un fibrado vectorial ξ es *orientable* si para cada punto $p \in U_{ij}$ (donde U_i y U_j son abiertos de trivialización de ξ) la restricción a $\pi^{-1}(U_{ij})$ del cambio de carta $\varphi_j \circ \varphi_i^{-1}$ induce una transformación sobre la fibra que conserva la orientación (como espacio vectorial).

El cambio de carta $\varphi_j \circ \varphi_i^{-1}$ en cada punto $p \in U_{ij}$ está dado por el valor de la función de transición $g_{ij}(p)$. Por ello, un fibrado vectorial es orientable si y sólo si $\det(g_{\alpha\beta}(p)) > 0$ para

cualquier $p \in B$. En particular, una variedad M es orientable si y sólo si su fibrado tangente τ_M es orientable. Si una variedad M es orientable y tiene k componentes conexas, entonces admite 2^k posibles orientaciones.

Definición 1.43 Llamamos *orientación de un fibrado orientable* ξ a una de las clases de equivalencia determinadas por la orientabilidad. Dicha clase recibe el nombre de clase de Euler del fibrado y se denota mediante $e(\xi)$; es no nula para cualquier fibrado orientable.

El fibrado tangente $\tau_{\mathbb{S}^m}$ de la esfera m -dimensional \mathbb{S}^m es orientable, mientras que la banda de Moebius extendida presentada más arriba es no-orientable.

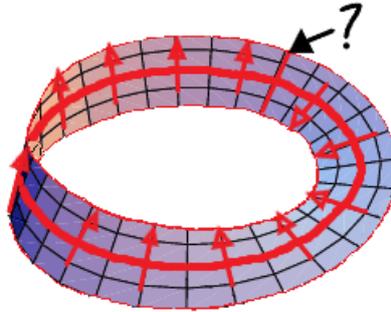


Figura 1.18: Fibrado vectorial no-orientable

Por último damos la definición de la *suma de Whitney* de fibrados vectoriales.

Definición 1.44 Dados dos fibrados vectoriales ξ y ξ' sobre un mismo espacio base B , se define el *fibrado suma de Whitney* y se denota mediante $\xi \oplus \xi'$ como el fibrado cuya fibra en cada punto base $b \in B$ está dado por la suma directa $\pi^{-1}(b) \oplus \pi'^{-1}(b)$ de las fibras de ξ y ξ' en $b \in B$

Dados dos fibrados vectoriales ξ y ξ' sobre un mismo espacio base B , construimos el espacio

$$E(\xi \oplus \xi') := \{(z, z') \in E \times E' \mid \pi(z) = \pi'(z')\}.$$

Haciendo $\pi_{\xi \oplus \xi'}(z, z') := \pi(z) = \pi'(z')$, se tiene que la 4-upla $(E(\xi \oplus \xi'), \pi_{\xi \oplus \xi'}, B, F_{\xi \oplus \xi'})$ es un fibrado vectorial al que llamamos *suma de Whitney* de ξ y ξ' , y denotamos mediante $\xi \oplus \xi'$.

Las funciones de transición de la suma de Whitney $\xi \oplus \xi'$ respecto a dicha trivialización están dadas por aplicaciones a valores en matrices diagonales por bloques:

$$g_{ij} : U_{ij} \rightarrow GL(r + r'; \mathbb{R}) \quad | \quad g_{ij}(p) = \begin{pmatrix} h_{ij} & 0 \\ 0 & k_{ij} \end{pmatrix} \quad \forall p \in U_{ij},$$

donde h_{ij} (respectivamente, k_{ij}) denota un sistema de funciones de transición para el fibrado vectorial ξ (respectivamente, ξ'). Esta descripción nos permite dar asimismo una expresión explícita para el isomorfismo natural $\xi \oplus \xi' \simeq \xi' \oplus \xi$.

1.5. Flujos tensoriales en Matemáticas e Ingeniería.

Un tensor de tipo (r, s) sobre una variedad diferenciable M es una sección del fibrado dado por el producto tensorial de r copias del fibrado cotangente $\tau_M^* = \Omega_M^1$ y s copias del fibrado tangente τ_M . Por ello, permite representar de forma simultánea la evolución espacio-temporal de una distribución de s vectores y evaluar el comportamiento de r covectores (formas diferenciales). Esta descripción intuitiva permite justificar la ubicuidad de los tensores en casi todas las ramas de Matemáticas, Física e Ingeniería.

Formalmente, el álgebra tensorial es un álgebra bigraduada con la diferencial exterior d y derivada parcial ∂ como operadores diferenciales. En la segunda subsección se revisan algunas propiedades formales básicas asociadas a la contracción y expansión de tensores. Estas operaciones son significativas para relacionar propiedades de objetos asociadas a diferentes

dimensiones, así como para gestionar “eventos” vinculados a la (des)aparición de elementos significativos, tal y como ocurre en secuencias de vídeo debido a la entrada-salida de elementos o bien a oclusiones parciales, p.e..

Desde el punto de vista discreto, el *producto tensorial* se lleva a cabo en términos de convoluciones que se interpretan simplemente como productos ordinarios (componente a componente) con “pesos” asociados a “cantidades” (vectores o covectores) variables sobre una variedad o, con más generalidad, un espacio topológico. En particular, todos los filtros utilizados para el procesamiento de cualquier tipo de señal se expresan en términos de convoluciones que no son otra cosa que la expresión discreta de un producto tensorial. La adaptación al marco de Redes Neuronales Artificiales (ANN) es inmediata en el marco de las CNN (Convolutional Neural Networks).

En esta sección se revisan algunos aspectos básicos de los tensores que serán utilizados en el último capítulo. Aunque los tensores pueden tener un tipo $r + s$ arbitrario Para fijar ideas, nos restringimos a tensores de orden “bajo”, es decir, con $r + s \leq 4$ de forma análoga al enfoque que se lleva a cabo en la Geometría Riemanniana Clásica de finales del S.XIX.

El cálculo tensorial es un instrumento que, si bien se utilizan coordenadas, las reglas operatorias son tales que siempre dan lugar a propiedades independientes del sistema utilizado, es decir, sus operaciones y resultados son invariantes por cambios de coordenadas. Luego el cálculo tensorial nos ofrece un aspecto invariante.

Comenzaremos tratando los *Tensores Cartesianos* y posteriormente los *Campos tensoriales*, ya que estos primeros sirven para interpretar la teoría básica del marco y como un entrenamiento para el caso general. Los Tensores cartesianos se refieren a cambios de coordenadas cartesianas ortogonales² en otras del mismo tipo. Para cambios de coordenadas más generales (coordenadas oblicuas o curvilíneas), la definición de tensor debe generalizarse.

1.5.1. Tensores Cartesianos.

En el cálculo tensorial aparece con mucha frecuencia sumatorios que en matemáticas denotamos por Σ . En el cálculo tensorial se “suprime” esta notación por comodidad y abreviatura. Esta convención se la conoce como *Convención de Einstein* y establece que *cuando en una expresión monomía figuren dos índices repetidos, se entenderá que se trata de una suma en la que los índices repetidos van sumados de 1 a n*.

De esta manera se escribirá simplemente

$$\begin{aligned} a_i b_i &= a_1 b_1 + \cdots + a_n b_n; \\ a_i b_j c_i &= a_1 b_j c_1 + \cdots + a_n b_j c_n. \end{aligned}$$

Una observación importante sobre esta “notación” es que los índices repetidos, puesto que únicamente indican que deben tomar los valores de 1 a n y sumarse, pueden representarse con cualquier letra, sin que la suma cambie. Se tiene así que

$$a_i b_i = a_j b_j$$

ambos miembros representan la suma $a_1 b_1 + \cdots + a_n b_n$.

A continuación consideramos el espacio ordinario de tres dimensiones, referido a un sistema de coordenadas cartesianas ortogonales. Representaremos las coordenadas de un punto en el espacio por x_1, x_2, x_3 .

Un cambio de coordenadas x_i a las coordenadas x'_i de otro sistema también cartesiano ortogonal con el mismo origen viene dado por

$$x'_i = a_{ih} x_h \quad (i = 1, 2, 3).$$

En matemáticas definimos un *vector* como el conjunto de tres componentes u_i que por un cambio de coordenadas (igual que en la educación anterior y que será el mismo en adelante) se transforman en

$$u'_i = a_{ih} u_h$$

²Son los que se cumple la expresión $d^2 = (y - x)^t (y - x)$ para la distancia entre dos puntos.

donde en el segundo miembro, según el convenio establecido anteriormente, el índice h va sumado de 1 a 3.

Sea v_j las componentes de otro vector las cuales, por el mismo cambio de coordenadas se transformarán en

$$v'_j = a_{jk}v_k.$$

Consideramos los nueve productos

$$u_i \otimes v_j = (u_i v_j) = \begin{pmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{pmatrix}$$

que constituye un nuevo ente geométrico que se conoce como *producto tensorial* de los vectores u_i y v_j . La ley de transformación induce a establecer la siguiente definición.

Definición 1.45 Dadas las 9 cantidades anteriores, t_{ij} , se dice que son componentes de un *tensor cartesiano de segundo orden* cuando por un cambio de coordenadas se transforman según la ley

$$t'_{ij} = a_{ih}a_{jk}t_{hk}.$$

Según esta definición, el producto tensorial de dos vectores es un tensor, *pero no todos los tensores de segundo orden son el producto tensorial de dos vectores*. Un ejemplo básico de un tensor de segundo orden es el *Tensor de Kronecker*, el cual se define a partir de el símbolo de Kronecker δ_{ij} .

Análogamente, el producto tensorial de tres vectores u_i, v_j, w_k es el conjunto de los 27 productos $u_i v_j w_k$. Por un cambio de coordenadas estas componentes se transformarán según la ley

$$u'_i v'_j w'_k = a_{ih}a_{jk}a_{km}u_h v_l w_m.$$

Este producto tensorial es un ejemplo de tensor cartesiano de tercer orden. En analogía sobre lo escrito, dadas las 27 cantidades t_{ijk} , que por un cambio de coordenadas se transformen según la ley $t'_{ijk} = a_{ih}a_{jl}a_{km}t_{hlm}$, diremos que ellas son componentes de un tensor de tercer orden.

Con esto podemos dar una definición general.

Definición 1.46 Dadas 3^p cantidades t_{i_1, \dots, i_p} se dice que son componentes de un *tensor cartesiano de orden p* si por un cambio de coordenadas se transforman en

$$t'_{i_1, \dots, i_p} = a_{i_1 h_1} \dots a_{i_p h_p} t_{h_1, \dots, h_p}$$

Tantos los índices i como h pueden tomar los valores 1, 2 y 3.

Según esta definición los vectores son tensores de primer orden. Además, dadas las componentes de un tensor en un sistema de coordenadas, la igualdad en la **Definición 1.46** permite calcular las mismas en cualquier otro sistema.

La propiedad fundamental de los tensores, de la cual deriva su importancia en la geometría y la física, es que siendo las fórmulas de transformación lineales y homogéneas, *si las componentes de un tensor se anulan en un sistema de coordenadas, se anularán también en cualquier otro sistema*³.

El convenio de Einstein hace que todo lo anterior valga exactamente, sin modificaciones, para el espacio euclidiano n -dimensional (para fijar ideas hemos estado suponiendo $n = 3$).

Definición 1.47 Dado el espacio euclidiano n -dimensional, llamaremos *tensor cartesiano de orden p* al conjunto de n^p componentes t_{i_1, \dots, i_p} .

Es fácil comprobar por inducción que los tensores cumplen la propiedad de transitividad (es necesaria para que la definición sea admisible), es decir, que por nuevos cambios de coordenadas por ley de transformación se conserva [24].

³La anulación de un tensor da siempre una propiedad intrínseca, independiente del sistema de coordenadas.

Definición 1.48 *Un invariante o tensor de orden 0 es una expresión que toma la misma forma en cualquier sistema de coordenadas.*

Por ejemplo, el producto escalar de dos vectores es un invariante.

A continuación describimos las operaciones con tensores cartesianos:

- **Suma/Resta:** Esta operación solo está definida para tensores del mismo orden. El *tensor suma o diferencia* de otros es el nuevo tensor que tiene por componentes la suma o diferencia de las componentes respectivas.
- **Multiplicación por un escalar:** Para multiplicar un escalar por un tensor se multiplica por el escalar cada componente del tensor.
- **Producto de tensores:** El producto de un tensor de orden p por otro tensor de orden q , es el tensor de orden $p + q$ cuyas componentes son los productos de las componentes del primero por las del segundo.
- **Contracción de índices:** Es una operación que permite obtener tensores de menor orden a partir de un tensor dado de orden $p \geq 2$. Consiste en lo siguiente: *Dado un tensor de orden $p \geq 2$, igualando dos índices y sumando respecto del índice igualado, el resultado es un nuevo tensor en el que han desaparecido estos dos índices.*

Supongamos, por ejemplo, un tensor t_{ijk} . Consideremos las componentes

$$u_k = t_{ijk}$$

donde en el segundo miembro el índice i , repetido, va sumando de 1 a n . Por un cambio de coordenadas es

$$u'_k = t'_{iik} = a_{ih}a_{il}a_{km}t_{hlm}$$

es decir

$$u'_k = \delta_{hl}a_{km}t_{hlm} = a_{km}t_{hhm} - a_{km}u_m.$$

Es decir, las u_k son componentes de un vector (tensor de orden uno), que se llama *tensor contraído* del t_{ijk} respecto de los dos primeros índices.

A partir del tensor t_{ijk} todavía se pueden formar otros dos tensores contraídos ($v_j = t_{iji}$; $w_i = t_{ijj}$).

Mediante esta operación de contracción, un tensor de segundo orden t_{ij} puede considerarse en cierto modo como un operador que a un vector le hace corresponder otro vector. Así, al vector v_i le hace corresponder el vector

$$u_i = t_{ij}v_j.$$

- **Expansión de índices:** Supongamos, para fijar las ideas, un tensor de segundo orden t_{ij} cuyas componentes sean funciones de las coordenadas de x_i , es decir, un *tensor función de punto*. Lo mismo sería para un tensor de cualquier orden. Se dice entonces que constituyen un *campos de tensores*.

En la ley de transformación

$$t'_{ij} = a_{ih}a_{jk}t_{hk}$$

los coeficientes a_{ih} son independientes del punto considerado, de manera que al derivar ambos miembros respecto de x'_m queda

$$t_{ij,m} = a_{ih}a_{jk}t_{hk,s} \frac{\partial x_s}{\partial x'_m}$$

donde

$$t'_{ij} = \frac{\partial t'_{ij}}{\partial x'_m} \quad t_{hk} = \frac{\partial t_{hk}}{\partial x_s}.$$

Según las fórmulas de transformación, la relación anterior se puede escribir de la siguiente manera

$$t'_{ij,m} = a_{ih}a_{jk}a_{ms}t_{hk,s},$$

lo cual nos dice que las derivadas parciales $t_{hk,s}$ son componentes de un vector de tercer orden.

La demostración es general para cualquier tensor de manera que: *Las derivadas parciales de las componentes de un tensor cartesiano respecto de las coordenadas son componentes de un nuevo tensor de un orden superior en una unidad.* A este nuevo tensor se le conoce como *tensor derivado del primero.*

- **Permutación de índices:** A partir de un tensor, permutando los índices de cada una de sus componentes, se obtiene otro tensor.

En [24] se muestra un ejemplo claro de que no vale con permutar las componentes para obtener un nuevo vector.

Definición 1.49 Un tensor de segundo orden t_{ij} se llama *simétrico* si se verifica

$$t_{ij} - t_{ji} = 0$$

y *antisimétrico* si

$$t_{ij} + t_{ji} = 0.$$

Al igual que para las matrices, se cumple que todo tensor de segundo orden es suma de un tensor simétrico y otro antisimétrico

$$t_{ij} = \frac{1}{2}(t_{ij} + t_{ji}) + \frac{1}{2}(t_{ij} - t_{ji}).$$

1.5.2. Tensores Cartesianos para representar deformaciones.

Desde mediados del S.XX, los tensores han sido utilizados en Ingeniería para representar aspectos vinculados a la deformación de objetos (elasticidad y viscosidad, p.e.) y la propagación de ondas, incluyendo fenómenos de interacción con la materia que pueden dar lugar a dislocaciones o fracturas (debidas a la aparición de vibraciones). Una referencia clásica es [18].

Las mallas triangulares o cuadrangulares proporcionan una aproximación topológica discreta a los fenómenos de deformación. Los Métodos de Elementos Finitos (FEM) proporcionan la versión computacional para representar los efectos representados por los tensores. La estructura graduada de los complejos simpliciales y cuboidales proporcionan el soporte para modelar los fenómenos observados (tal como se explico en la subsección anterior).

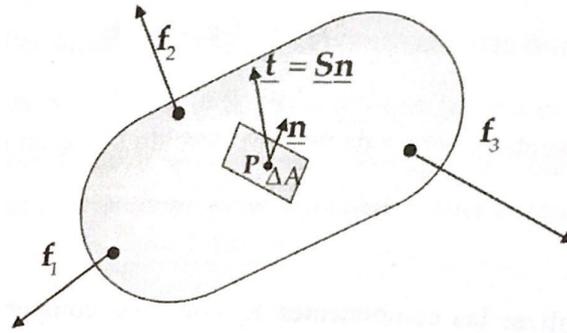
En esta parte hablaremos de los tensores de tensiones y del tensor de deformaciones, que son de orden dos (Capítulo 7 de [25]). En esta parte, al tratarse de orden 2, los tensores describen una aplicación lineal que llevan vectores de \mathbb{R}^3 en \mathbb{R}^3 ya que pueden representarse por matrices 3×3 , es decir, elementos de $\text{lin}(\mathbb{R}^3, \mathbb{R}^3)$ (que tiene estructura de espacio vectorial).

El tensor de tensiones.

Supongamos que tenemos un sólido o un fluido continuo que está sometido a un sistema de fuerzas. La fuerza por unidad de área debida a la interacción entre partes internas del continuo inducidas por el sistema de fuerzas externo se llama *vector de tensión* o *vector de tracción*, $\underline{\mathbf{t}}$.

Para visualizar $\underline{\mathbf{t}}$, imaginemos que en el interior del medio continuo realizamos en el punto P un pequeño corte separando la materia en dos partes. Este corte genera una pequeña superficie de área ΔA y el vector unitario $\underline{\mathbf{n}}$, perpendicular a la superficie. En este caso, $\underline{\mathbf{t}} \Delta A$

es la fuerza que ejerce la parte de materia a donde apunta \underline{n} sobre el elemento de superficie $\underline{n} \triangle A$.



El primer postulado de Cauchy establece que el vector de tracciones \underline{t} depende del punto P y de la normal \underline{n} del elemento de superficie, pero no de la forma del elemento de superficie.

El teorema de Cauchy sobre las tensiones en un cuerpo establece que las tracciones \underline{t} dependen linealmente de la orientación \underline{n} del elemento de superficie, es decir, existe un tensor de segundo orden $\underline{\underline{S}}$ de forma que

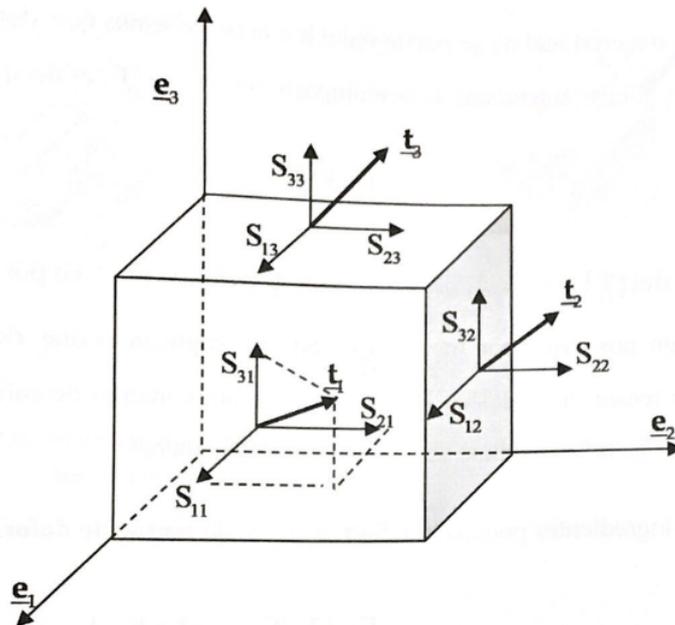
$$\underline{t} = \underline{\underline{S}} \underline{n}.$$

Al tensor $\underline{\underline{S}}$ se le conoce como *tensor de tensiones de Cauchy* que envía cualquier vector \underline{n} al vector de tensión \underline{t} que actúa sobre el elemento de superficie de normal \underline{n} . Se puede demostrar que el tensor de tensiones de Cauchy es simétrico, imponiendo el balance del momento angular.

Si elegimos un sistema de coordenadas cartesiano anclado a un punto del cuerpo, podemos obtener las componentes cartesianas de $\underline{\underline{S}}$ en ese punto a través

$$S_{ij} = \underline{e}_i \cdot \underline{\underline{S}} \underline{e}_j = \underline{e}_i \cdot \underline{t}_j.$$

Podemos visualizar las componentes S_{ij} como las componentes de vectores de tracción actuando en las caras de un pequeño cubo de aristas paralelas a los ejes de coordenadas.



Las componentes S_{11} , S_{22} y S_{33} se denotan por *tensiones normales*, mientras que el resto son *tensiones tangenciales*. De forma esquemática se puede decir que al tirar o empujar un objeto se crean tensiones normales mientras que al retorcer un objeto se crean tensiones tangenciales.

Tensor de deformaciones.

Este tipo de tensores se ven dentro de la cinemática de un medio continuo. En otras palabras, supongamos que cada partícula de un sólido o fluido se desplace de su posición inicial a $\underline{\mathbf{x}}$ a una posición final dada por $\underline{\mathbf{T}} \underline{\mathbf{x}}$, siendo $\underline{\mathbf{T}}$ un tensor de segundo orden. Como el volumen de un objeto real no se puede reducir a cero, sabemos que $\det(\underline{\mathbf{T}}) > 0$, por lo que podemos aplicar el teorema de descomposición polar a $\underline{\mathbf{T}}$ el cual nos dice que todo tensor de segundo orden no singular puede ponerse como composición de un tensor ortogonal y un tensor simétrico definido positivo, es decir,

$$\underline{\mathbf{T}} = \underline{\mathbf{S}} \underline{\mathbf{R}}.$$

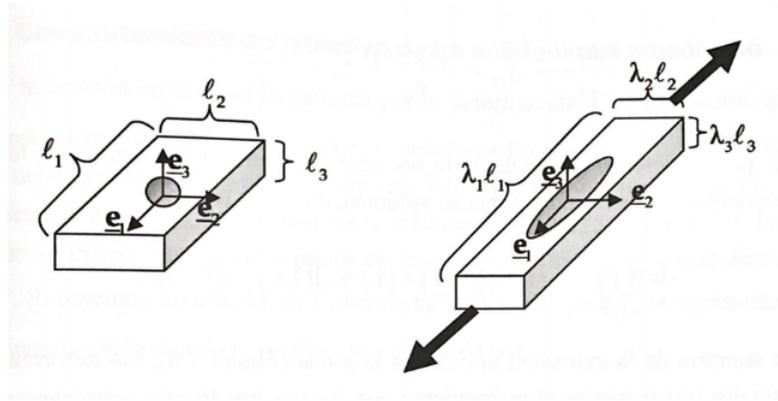
Como $\det(\underline{\mathbf{T}}) = \det(\underline{\mathbf{S}})\det(\underline{\mathbf{R}})$ y $\underline{\mathbf{S}}$ es definido positivo⁴ tenemos que $\det(\underline{\mathbf{S}}) > 0$ implicando que $\det(\underline{\mathbf{R}}) = 1$, es decir, $\underline{\mathbf{R}}$ es un tensor de rotación. $\underline{\mathbf{R}}$ representa un movimiento sólido rígido del continuo, preservando la distancia de dos puntos cualesquiera.

Con esto podemos definir el siguiente *tensor de deformación*

$$\underline{\underline{\Gamma}} = (\underline{\mathbf{T}} \cdot \underline{\mathbf{T}}^t)^{\frac{1}{2}} - \underline{\mathbf{I}} = \underline{\mathbf{S}} - \underline{\mathbf{I}}.$$

El tensor de segundo orden $\underline{\underline{\Gamma}}$ es una medida de la cantidad de distorsión que tiene lugar en un proceso de deformación. Si denotamos a las componentes cartesianas de $\underline{\underline{\Gamma}}$ por γ_{ij} (es decir, $\gamma_{ij} = \mathbf{e}_i \cdot \underline{\underline{\Gamma}} \mathbf{e}_j$) las componentes γ_{11} , γ_{22} y γ_{33} se denominan *deformaciones normales*, de extensión o longitudinales, mientras que el resto son las *deformaciones de corte* o transversales.

A modo de ejemplo, supongamos que queremos deformar una goma elástica de forma que la goma pase de ser un sólido rectangular de longitud de aristas (l_1, l_2, l_3) a otro sólido rectangular de longitud de aristas $(\lambda_1 l_1, \lambda_2 l_2, \lambda_3 l_3)$.



Una partícula que estaba inicialmente en la posición $\underline{\mathbf{x}} = x_i \mathbf{e}_i$ en la goma sin deformar pasa a la posición

$$\underline{\mathbf{T}} \underline{\mathbf{x}} = x_i \underline{\mathbf{T}} \mathbf{e}_i = \lambda_i x_i \mathbf{e}_i = \lambda_1 x_1 \mathbf{e}_1 + \lambda_2 x_2 \mathbf{e}_2 + \lambda_3 x_3 \mathbf{e}_3$$

en la goma deformada, por lo que

$$\underline{\mathbf{T}} = \lambda_1 \mathbf{e}_1 \otimes \mathbf{e}_1 + \lambda_2 \mathbf{e}_2 \otimes \mathbf{e}_2 + \lambda_3 \mathbf{e}_3 \otimes \mathbf{e}_3.$$

Como $\underline{\mathbf{T}}$ es simétrico, el tensor $\underline{\mathbf{R}} = \underline{\mathbf{I}}$, por lo que no hay rotación de sólido rígido y el tensor de deformación viene dado por

$$\underline{\underline{\Gamma}} = (\lambda_1 - 1) \mathbf{e}_1 \otimes \mathbf{e}_1 + (\lambda_2 - 1) \mathbf{e}_2 \otimes \mathbf{e}_2 + (\lambda_3 - 1) \mathbf{e}_3 \otimes \mathbf{e}_3$$

es decir, las deformaciones longitudinales corresponden al incremento relativo de las aristas

$$\gamma_{11} = \frac{l_1 \lambda_1 - l_1}{l_1}, \quad \gamma_{22} = \frac{l_2 \lambda_2 - l_2}{l_2}, \quad \gamma_{33} = \frac{l_3 \lambda_3 - l_3}{l_3}$$

⁴ $\mathbf{v}_i \cdot \underline{\mathbf{S}} \mathbf{v}_i = \mathbf{v}_i \cdot (\lambda_i \mathbf{v}_i \otimes \mathbf{v}_i) \mathbf{v}_i = \lambda_i \mathbf{v}_i \cdot (\mathbf{v}_i \cdot \mathbf{v}_i) \mathbf{v}_i = \lambda_i \mathbf{v}_i \cdot \mathbf{v}_i = \lambda_i$, donde $\lambda_i > 0$ son los autovalores de $\underline{\mathbf{T}}$ asociados a los autovectores \mathbf{v}_i .

mientras que las deformaciones de corte son cero. Si además suponemos que la goma es incomprensible, es decir, no cambia su volumen, entonces tenemos

$$\det(\underline{\mathbf{T}}) = 1 = \det(\underline{\mathbf{\Gamma}} + \underline{\mathbf{I}}) = (1 + \gamma_{11})(1 + \gamma_{22})(1 + \gamma_{33}).$$

Por simetría de la extensión aplicada a la goma (como se muestra en la figura anterior), las deformaciones longitudinales transversales al estiramiento son iguales por lo que obtenemos

$$\gamma_{22} = \gamma_{33} = \frac{1}{\sqrt{1 + \gamma_{11}}} - 1.$$

Es decir, si triplicamos la longitud de la goma elástica ($\lambda_1 = 3$), cada una de las dimensiones transversales se reducirá alrededor de un 42%.

1.5.3. Campos Tensoriales.

Con el objetivo de conectar sobre la interpretación que tenemos de un tensor en esta subsección daremos un enfoque general sobre los tensores a partir de las funciones multilineales.

Definición 1.50 Sean V_1, \dots, V_n y W espacios vectoriales. Decimos que una función es *multilineal* cuando es lineal para cada uno de sus argumentos, es decir

$$f(v_1, \dots, \lambda v_i + \mu v'_i, \dots, v_n) = \lambda f(v_1, \dots, v_i, \dots, v_n) + \mu f(v_1, \dots, v'_i, \dots, v_n) \quad \forall i \in \{1, \dots, n\}.$$

Cuando es lineal solo para cierto argumento v_i decimos que es i -lineal.

Definición 1.51 Supongamos que $\tau \in V^*$ y $\theta \in W^*$ (funciones lineales de valor real). Definimos el *producto tensorial* de τ y θ como la función bilineal real $\tau \otimes \theta : V \times W \rightarrow \mathbb{R}$ tal que

$$\tau \otimes \theta(v, w) = (\tau v)(\theta w).$$

Las funciones multilineales pueden multiplicarse por un escalar o se pueden combinar dos funciones multilineales (deben tener el mismo rango y dominio) y el resultado sigue siendo una función multilineal del mismo tipo. Esto nos muestra que el espacio de funciones i -lineales de $V_1 \times \dots \times V_n$ en W forma un espacio vectorial, denotado por $L(V_1, \dots, V_n; W)$.

Sea ahora V un espacio vectorial. Las funciones multilineales reales con argumentos en V (o V^*) serán *tensores sobre V* y el espacio vectorial que forman será el *espacio tensorial sobre V* . El número de variables de V y V^* son lo que se conoce como *grados de un tensor*. El número de V es el grado *covariante*, mientras que el número de V^* es el grado *contravariante*. De esta manera decimos que un tensor en $V^* \times V \times V$ tiene grado $(1, 2)$.

De esta manera un tensor de grado (r, s) tendrá r grados contravariantes y s grados covariantes. Cuando se encuentra una relación en cuanto a la permutación de los dominios de los tensores decimos que tienen una *propiedad de simetría* (estas propiedades son estudiadas desde la sección 2.15 a la 2.19 en [23]).

Al espacio de funciones multilineales en $V^* \times V \times V$ lo denotaremos por

$$V \otimes V^* \otimes V^* = T_2^1(V; \mathbb{R}).$$

El “cambio” en los espacios es intencionado y está justificado por el hecho de que es una generalización del caso de los tensores de grado 1. De hecho, al ser V^* el espacio que consiste en las funciones lineales en V y tener que $V \simeq (V^*)^*$, podemos considerar como $(V^*)^*$, es decir, el espacio de funciones lineales en V^* . De manera general, un tensor de tipo (r, s) forma un espacio vectorial denotado por

$$T_s^r = T^{s,r} = V \otimes \overset{r}{\underbrace{V^* \otimes \dots \otimes V^*}} \otimes \overset{s}{\underbrace{V \otimes \dots \otimes V}} \otimes V^*,$$

que consiste en funciones multilineales sobre

$$V^* \otimes \overset{r}{\underbrace{V^* \otimes \dots \otimes V^*}} \otimes V \otimes \overset{s}{\underbrace{V \otimes \dots \otimes V}} \otimes V.$$

El *producto tensorial*⁵ de un tensor A del tipo (r, s) y un tensor B del tipo (p, q) es un tensor $A \otimes B$ del tipo $(r + p, s + q)$ definido, como una función sobre $(V^*)^{(r+p)} \times V^{(s+q)}$, por

$$A \otimes B = A(\tau^1, \dots, \tau^r, v_1, \dots, v_s)B(\tau^{r+1}, \dots, \tau^{r+p}, v_{s+1}, \dots, v_{s+q}).$$

⁵Cumple la asociatividad y la distributividad respecto de la suma.

De esta manera, un tensor de tipo $(0, 0)$ es un escalar ($T_0^0 = \mathbb{R}$). Un tensor de tipo $(1, 0)$ es llamado *vector contravariante* y uno de tipo $(0, 1)$ es un *vector covariante*. A un tensor del tipo $(r, 0)$ se le llama *tensor contravariante* y uno de tipo $(0, s)$ es un *tensor covariante*.

1.5.4. Contracción y expansión.

En esta subsección al espacio vectorial lo denotaremos por E y el cuerpo de valores por F por tener una notación mas general.

Definición 1.52 Dado un tensor $t \in T_s^r(E; F)$ se define la (k, ℓ) -contracción como una aplicación $C_\ell^k : T_s^r(E; F) \rightarrow T_{s-1}^{r-1}(E; F)$ dada por

$$C_\ell^k(t^I e_I \otimes e^J) := t_{j_1 \dots j_{\ell-1} p j_{\ell+1} \dots j_r}^{i_1 \dots i_{k-1} p i_{k+1} \dots i_s} e_{i_1} \otimes \dots \otimes \hat{e}_{i_k} \otimes \dots \otimes e_{i_s} \otimes e^{j_1} \otimes \dots \otimes \hat{e}^{j_\ell} \otimes \dots \otimes e^{j_r},$$

donde $I = \{i_1, \dots, i_s\}$ y $J = \{j_1, \dots, j_r\}$.

Alteraciones.

Por último, tenemos otras dos operaciones que consisten en *bajar* y *subir* índices respectivamente que (por analogía con la grafía utilizada en la notación musical para indicar modificaciones de semitonos en las notas) representaremos mediante $^\flat$ y $^\sharp$, respectivamente. Más explícitamente, si g_{ij} es la matriz de una métrica con inversa g^{ij} , entonces el producto interior define isomorfismos que se denota mediante

$$\begin{aligned} \flat : E = T_0^1(E) &\rightarrow E^* = T_1^0(E) & | & (x^\flat)_i := g_{ij} x^j; \\ \sharp : E^* = T_1^0(E) &\rightarrow E = T_0^1(E) & | & (\alpha^\sharp)^i := g^{ij} \alpha_j. \end{aligned}$$

A partir de estas dos últimas operaciones, es posible visualizar en términos de cada métrica g sobre E cómo se llevan a cabo los isomorfismos entre tensores de tipo (r, s) con $r + s$ fijo en términos de las bases para el producto tensorial descritas más arriba. En particular, se tiene que $g^{jk} g_{ik} = \delta_i^j$, donde se interpreta la δ de Kronecker como un tensor de tipo $(1, 1)$.

Subida y bajada de índices.

A partir del tensor $t = t_{\ell m}^{ijk} e_i \otimes e_j \otimes e_k \otimes e^\ell \otimes e^m \in T_2^3(E)$, se construyen diferentes tensores asociados mediante las operaciones de subir y bajar índices. Por ejemplo,

$$\begin{aligned} t_{\ell}^{ijk m} &= g^{mp} t_{\ell p}^{ijk} & t^i{}_{jkl}{}^m &= g_{ja} g_{kb} g^{mc} t_{\ell c}^{iab}, \\ t_{ij}{}^{klm} &= g_{ia} g_{jb} g^{\ell c} g^{md} t_{cd}{}^{abk} & t_i{}^j{}_{k\ell}{}^m &= g_{ia} g_{kb} g^{\ell c} t_{cm}{}^{ajb}. \end{aligned}$$

Capítulo 2

Contribuciones teóricas

2.1. Topología Algebraica Computacional.

La Topología Algebraica es la parte de la Topología que usa métodos algebraicos y/o discretos (incluyendo aspectos combinatorios) para identificar y caracterizar propiedades topológicas de espacios topológicos X y aplicaciones $f : X \rightarrow Y$ entre espacios topológicos bajo relaciones de equivalencias dadas por los homeomorfismos. Estos métodos usan propiedades algebraicas para relacionar (clases de equivalencia de) multi-caminos Γ vinculados a puntos de control (*Teoría de la homotopía*), y/o PL-estructuras superpuestas vinculadas a complejos simpliciales o cuboidales.

La generación automática de dichas PL-estructuras a partir de nubes de puntos con densidad irregular es un problema no trivial que se resuelve de forma recursiva a partir de modelos de propagación. De este modo, es posible generar PL-aproximaciones a "formas" que se gestionan en términos de *Teorías de Homología* para complejos graduados.

Las características básicas de los objetos a representar se reformulan en términos de invariantes algebraicos las cuáles son dadas por el grupo fundamental $\pi_1(X)$ y los grupos de homología $H_n(X)$. Esta información debe ser completada con otros "descriptores" que proporcionen información sobre aspectos más finos de los objetos.

A lo largo de este trabajo veremos como la *Topología Algebraica Computacional* proporciona herramientas para generar y comparar PL-modelos con objetos complejos, proporcionando soporte para el análisis de todo tipo de funcionales definidos sobre los PL-complejos.

El tratamiento de múltiples objetos (a veces representados por puntos) y estructuras cambiantes de objetos particulares (representadas por *blobs*¹, por ejemplo) son dos tópicos usuales en el manejo de la información de gran cantidad de datos. Desde el punto de vista matemático, la Topología nos ofrece una colección de modelos y métodos para el tratamiento de objetos en términos de multi-caminos y PL-estructuras superpuestas.

En este trabajo se adaptan las estrategias de la Topología desde un punto de vista computacional; en la práctica podemos encontrarnos con problemas computacionales difíciles de resolver encontrándonos con que alguno de ellos son problemas NP debido al crecimiento exponencial del número de operaciones a realizar para resolverlos.

En el marco de la CAT (*Computational Algebraic Topology*), presentamos las PL-estructuras superpuestas en términos de n -cadenas cuboidales cuya utilización formal da lugar a *complejos cuboidales* sobre las nubes originales de puntos.

El *tipo de homotopía* de los espacios topológicos ofrece un criterio inicial para distinguir un objeto de otro (es decir, no son homeomorfos). El problema está en dar un criterio efectivo para calcular el tipo de homología y otras características más finas. Por este motivo introducimos los conceptos de multi-caminos (representando interpolaciones, deformaciones, trayectorias) y PL-estructuras superpuestas (cuboidal, celular).

Para fijar ideas, las estructuras de datos más simples están dadas inicialmente en términos de listas asociadas a nubes de puntos. A dichas listas se asocian de forma automática PL-estructuras superpuestas (incluyendo PL-caminos y PL-mallas). Las estructuras algebraicas

¹Basic Large Object; elementos utilizados en las bases de datos para almacenar datos de gran tamaño que cambian de forma dinámica.

básicas las cuales pueden estar superpuestas a espacios topológicos están dadas por (multi-)caminos y mallas generalizadas.

2.1.1. Mallas en PL-formas.

Las mallas cuboidales y simpliciales son de uso común para proporcionar un enfoque más sencillo a los objetos complejos. Proporcionan modelos continuos que se pueden adaptar a distribuciones arbitrarias o datos discretos. Los complejos simpliciales y cuboidales ofrecen un marco más general, que puede manipularse en términos de PL- o de transformaciones continuas. Análogamente, la relación entre los objetos se representan inicialmente mediante aplicaciones topológicas $f : X \rightarrow Y$ que permiten asociar PL-deformaciones dadas por PL-caminos entre los complejos (simpliciales o cuboidales).

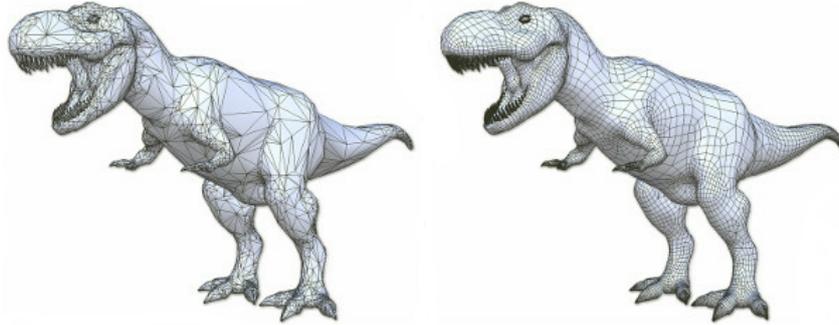


Figura 2.1: Complejo simplicial y cuboidal, respectivamente.

Motivación.

De una forma intuitiva, una malla es una “teselación” eventualmente irregular de un espacio ambiente que se reasigna en objetos o escenas según una colección de funciones. En los casos más simples, se consideran las mallas dadas por las células básicas que son un polígono o un poliedro del mismo tipo.

El caso más simple corresponde a las células básicas dadas por polígonos regulares que corresponden exactamente a un tipo de polígono. Por medio de un simple argumento geométrico (conocido ya por Euclides), se puede probar que la célula básica en el plano es un triángulo equilátero, un cuadrado o un hexágono regular.

Este argumento se extiende a dimensión tres dando lugar a tres tipos básicos (salvo dualidad) de poliedros regulares que proporcionan teselaciones del espacio euclídeo ordinario dadas por tetraedros, hexaedros (duales de los octaedros) y dodecaedros (duales de los icosaedros)².

En este trabajo, se destacan las teselaciones asociadas a cubos debido a una mayor facilidad para el tratamiento computacional. Desde el punto de vista matemático, los cubos generalizados ó *cuboides* están dados por la imagen de una aplicación continua T sobre un n -cubo $[0, 1]^n$.

Teselaciones y deformaciones.

Una deformación genérica de una teselación triangular o cúbica da lugar a mallas simpliciales o cuboidales en el espacio. En particular, la descripción triangular clásica se puede extender a complejos tridimensionales reemplazando las celdas básicas por tetraedros, cubos o prismas hexagonales regulares. En este trabajo la mayoría de las aplicaciones están restringidas a casos de baja dimensión, donde las aplicaciones, transformaciones y/o deformaciones pueden ser más fácilmente controlables. Las teselaciones más fáciles de gestionar desde el punto de vista computacional (en relación con procedimientos de inserción y borrado) están dadas por representaciones cuboidales.

Una malla plana normal viene dada por una colección de mapas continuos definidos en una teselación plana normal sobre un plano euclidiano o en el plano afín asociado. Para el caso 3D, basta con reemplazar “planar” por “volumétrico” y “plano” por “espacio”. Esta

²La prueba ya era conocida, al menos, por los matemáticos griegos en el siglo IV a.C.

descripción permite considerar las mallas normales en objetos planos o volumétricos que verifican las condiciones de compatibilidad en los límites de las celdas básicas.

Aquí utilizamos mallas cuboidales porque el coste computacional para teselaciones triangulares en el caso planar es manejable (complejidad $O(N \log N)$; su extensión al caso 3D presenta una complejidad $O(N^2 \log N)$). En presencia de varios cientos de miles (o incluso millones de puntos capturados con un escaneo láser, por ejemplo) de puntos distribuidos irregularmente, es necesario llevar a cabo un muestreo e implementar procesos distribuidos con algoritmos paralelizables que permitan descomponer y pegar mallas a diferentes LoDs (*Niveles de detalle*). Las diferentes estrategias para trabajar a diferentes resoluciones (LoD: Level of Detail) presentando problemas de descomposición y pegado que ralentizan los cálculos y, por consiguiente, la toma semi-automática de decisiones.

Modelos algebraicos.

Alternativamente, se pueden introducir modelos cuboidales que son fácilmente parametrizables. Los complejos cuboidales proporcionan un enfoque general para los modelos parametrizables emparejando las aplicaciones $f : [0, 1]^n \rightarrow X$. En particular, una “serpiente” de grado d se puede considerar como la imagen de una “pieza ponderada” (pesos relativos) de una curva normal racional de grado d . Por tanto, una snake se puede considerar como una curva localmente parametrizada por la imagen de una subdivisión (no necesariamente regular) del intervalo unidad $[0, 1]$.

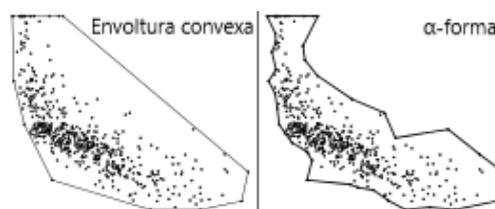
El producto tensorial de dos serpientes de grado d_1 y d_2 da una *superficie spline* S de bigrado (d_1, d_2) de uso común en ingeniería y diseño. Análogamente, un *T-spline* (triple) de trigrado (d_1, d_2, d_3) se define como el producto tensorial de tres serpientes de grados d_1 , d_2 y d_3 . Los productos de snakes se utilizan comúnmente para el modelado 3D de objetos complejos, como por ejemplo los órganos internos del cuerpo humano.

Para simplificar, en este TFG no se consideran T-splines. Para facilitar el tratamiento computacional se supone que los PL-modelos están dados por expresiones algebraicas de grado bajo (habitualmente, menor o igual que cuatro); es decir, por

- *Snakes (serpientes)* para curvas, es decir, curvas racionales de grado bajo d con “pesos” apropiados.
- *B-splines* de bigrado (p, q) para superficies dadas como el producto de dos trozos de curvas racionales de grado p y q , con $p + 1$ y $q + 1$ puntos de control, respectivamente.

2.1.2. Las α -formas.

Las α -formas o formas α son una secuencia de complejos derivados del diagrama de Voronoi y la triangulación de Delaunay (son construcciones fundamentales en la geometría computacional que explicaremos a continuación). Las α -formas fueron definidas por Edelsbrunner, Kirkpatrick y Seidel. Las definieron para un conjunto finito de puntos en el plano [20] y posteriormente Edelsbrunner y Mücke lo generalizaron a dimensiones superiores [21]. Las α -formas asociadas a un conjunto de puntos es una generalización del concepto de envoltura convexa de un conjunto.



Como una breve idea pensemos en que, dependiendo del valor α , podremos describir diferentes envolturas y con ello crear diferentes tipos de complejos.

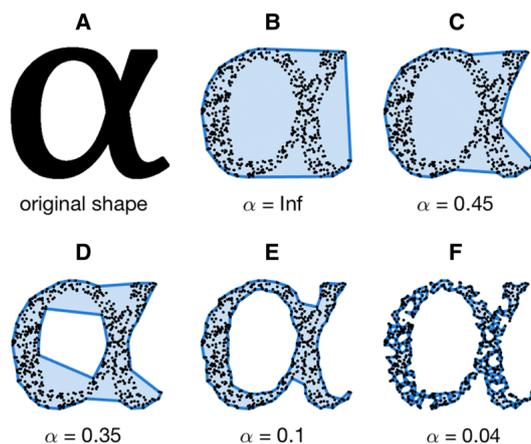


Figura 2.2: Tipos de envolturas en función de α

Para poder describir lo que son las α -formas debemos saber ciertos conocimientos previos.

Definición 2.1 Una *subcomplejo* L de un *complejo cuboidal* K es un *complejo cuboidal* tal que $L \subseteq K$.

Definición 2.2 Una *filtración de un complejo* K es una *secuencia anidada de subcomplejos*, $\emptyset = K^0 \subset K^1 \subset K^2 \subset \dots \subset K^m = K$. Llamamos a un *complejo* K con una *filtración* un *complejo filtrado*.

Nótese que el complejo $K^{i+1} = K^i \cup T^i$, donde T^i es un conjunto de cubos singulares.

Diagrama de Voronoi.

El diagrama, teselación o descomposición de Voronoi, también conocido como teselación de Dirichlet o Polígonos de Thiessen (debido a que estos matemáticos también realizaron estudios sobre esta materia) es una construcción geométrica formalizada por el matemático ruso de origen ucraniano G.F. Voronoi (1868-1908) en [8]. El diagrama de Voronoi es una solución local al problema de caracterización del área de influencia para un conjunto finito S de sitios [7]. Tiene una amplia gama de aplicaciones en geometría computacional en relación con la resolución de problemas de localización-asignación de recursos ó de obtención de rutas óptimas de desplazamiento en navegación automática, por ejemplo. Las triangulaciones de Delaunay son duales de los diagrama de Voronoi en el plano; por ello son igualmente importantes en la geometría computacional y proporcionan un nexo estructural con los complejos simpliciales planares (hablaremos sobre ellos en la siguiente subsección).

Definición 2.3 Sea $P = \{p_1, \dots, p_m\} \subset \mathbb{R}^n$ un conjunto de puntos y d una distancia. Definimos *Célula de Voronoi* o *Región de Voronoi*, $V(p_i)$, de un punto $p_i \in P$ como el conjunto de todos los puntos más cercanos a p_i que cualquier otro punto en P , es decir,

$$V(p_i) = \{x \in \mathbb{R}^n | d(x, p_i) \leq d(x, p_j), \text{ para } j \neq i\}.$$

La colección de estos elementos forman el *diagrama de Voronoi*.

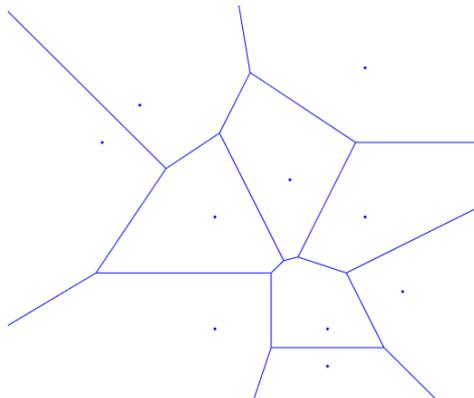


Figura 2.3: Diagrama de Voronoi de un conjunto P de puntos

Proposición 2.1 *El diagrama de Voronoi tiene las siguientes propiedades:*

1. Toda $V(p_i)$ es convexa.
2. $\bigcup_{p \in P} V(p) = \mathbb{R}^n$.
3. Si $V(p_i) \cap V(p_j) \neq \emptyset$ entonces su intersección es un subconjunto del hiperplano que es perpendicular y biseciona a la arista $p_i p_j$.
4. Los interiores de las células de Voronoi son disjuntos.
5. $V(p_i)$ es ilimitada si y solo si p_i se sitúa en la envolvente convexa de P .
6. El centro de la circunferencia que pasa por los puntos de P de $V(p_1)$, $V(p_2)$ y $V(p_3)$ es precisamente el vértice en el que se juntan las tres regiones.
7. El interior de la circunferencia definida anteriormente descrita no contiene ningún punto de P .
8. El punto vecino más cercano a p_i dentro del conjunto P , al que se denomina p_j define una arista de una célula de Voronoi.
9. Un diagrama de Voronoi de N puntos tiene a lo sumo $2N - 5$ vértices y $3N - 6$ aristas.

La demostración de las propiedades se puede encontrar en [6] y [7].

Existe una amplia variedad de algoritmos para el cálculo del diagrama de Voronoi de un conjunto de puntos. Dos de los métodos de cómputo del diagrama más importantes son el método de la *intersección de semiplanos* (uno de los métodos pioneros) y el *algoritmo de Fortune* (mejoró sensiblemente la complejidad del método de la intersección de semiplanos).

Triangulación de Delaunay.

Si en un diagrama de Voronoi se conectan todos los pares de puntos $p \in P$ de células adyacentes el conjunto de segmentos resultante forma una triangulación del conjunto, llamada triangulación de Delaunay, una construcción dual del diagrama de Voronoi de gran utilidad en el modelado de la superficie de terrenos.

Definición 2.4 *El complejo Delaunay o Triangulación de Delaunay, τ , es definido como el dual geométrico del diagrama de Voronoi.*

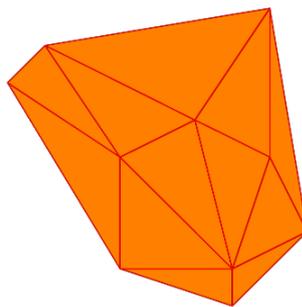


Figura 2.4: Triangulación de Delaunay asociada al diagrama de la Figura 2.4

El dual geométrico es una construcción similar al nervio de la envolvente: Si las células $V(p_0), \dots, V(p_m)$ tienen intersección no vacía, entonces la envoltura convexa de p_0, \dots, p_m es su dual.

En una triangulación de Delaunay, por definición cada vértice corresponderá a un punto $p \in P$ de Voronoi, y por lo tanto la superficie de cada triángulo corresponderá igualmente a un único vértice del diagrama de Voronoi. Además se da la circunstancia de que la triangulación de Delaunay forma un grafo, ya que ninguna de las aristas de los triángulos se cruza con otra.

Los α -complejos.

A continuación describimos cómo Edelsbrunner introduce un parámetro α en el diagrama de Voronoi y la triangulación de Delaunay. Esta operación genera una secuencia de complejos que triangulan los α -vecinos del conjunto de datos P .

Definición 2.5 El cerrado α -vecinos, P_α , es la unión de las bolas cerradas de radio α con centro $p \in P$, es decir,

$$P_\alpha = \bigcup_{p \in P} B_\alpha(p), \text{ donde } B_\alpha(p) = \{x | d(x, p) \leq \alpha\}.$$

Estas α -bolas además forman una envoltura de P_α .

Consideramos ahora la intersección de las α -bolas con las células de Voronoi:

$$V_\alpha(p) = V(p) \cap B_\alpha(p).$$

Esto nos proporciona una envoltura de P_α de regiones cerradas y convexas, llamada α -diagrama:

$$P_\alpha = \bigcup_{p \in P} V_\alpha(p).$$

Si los puntos de P están en una posición general, entonces el nervio del α -diagrama es un subconjunto de la triangulación de Delaunay, conocido como α -complejo, C_α . Las construcción de los α -complejos es muy útil en el desarrollo de algoritmos eficientes.

En primer lugar, ya que $C_\alpha \subset \tau$, podemos ver que si $\alpha < \alpha'$, entonces $C_\alpha \subset C_{\alpha'}$. Además, si la triangulación de Delaunay es finita, existe un número finito (distintos entre si) de α -complejos

$$\emptyset = C_{\alpha_0}, \dots, C_{\alpha_n} = \tau.$$

Estos están ordenados por el incremento del parámetro α^3 . A partir de estos complejos nosotros formamos las α -formas que son muy útiles en el reconocimiento de objetos.

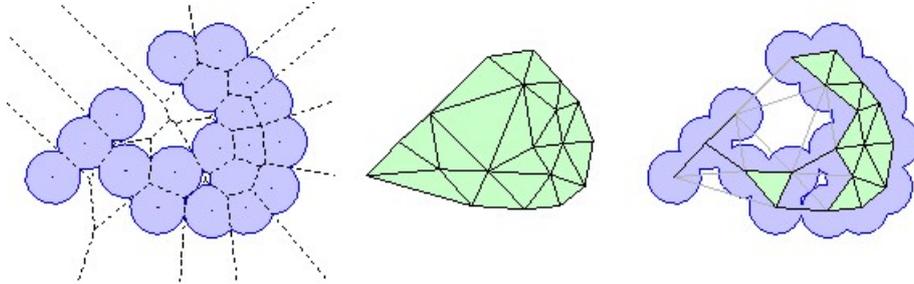


Figura 2.5: Diagrama de Voronoi, Triangulación de Delaunay y α -forma asociada a un conjunto de datos

Utilidad de los α -complejos.

Los α -complejos permiten reconstruir objetos con un LoD que varía según el parámetro α que testeamos. Si razonamos desde el punto de vista dual, podemos considerar funcionales lineales definidos sobre cada α -complejo para cada grado. De este modo, se obtiene una PL-aproximación pesada a diferentes LoD para campos definidos sobre el complejo de las α -formas.

El ajuste a una “forma” se puede entender como un proceso de optimización en el que se minimiza la diferencia entre la forma que se desea alcanzar y las aproximaciones dadas por α -formas. De manera similar, el ajuste (incluyendo el aprendizaje) a una tarea entendida como un (multi-)camino en el espacio de funcionales se puede entender asimismo como el problema dual del ajuste a una forma; en otras palabras, es un problema de control para el que existen asimismo diferentes opciones dependiendo del tipo de campo a considerar.

Estas ideas tan simples sugieren que las α -formas deberían adaptarse bien a procesos de tipo recurrente como los utilizados en sistemas inteligentes que se abordan en el **Capítulo**

³Por convenio tomamos $\alpha_0 = 0$ y $C_0 = \emptyset$.

3. Sin embargo, aparentemente, este enfoque aún no ha sido desarrollado en relación con el aprendizaje de una forma o de una tarea.

2.1.3. Colapso y Expansión.

En el problema del Reconocimiento Automático de objetos modelados como unión de variedades topológicas en una escena, puede ocurrir que alguno de estos objetos esté “tapando” otro objeto de la escena; su re-aparición da lugar a cambios bruscos (discontinuidades) que se traducen en una “expansión” para las listas de elementos básicos. Alternativamente, puede ocurrir que, en lugar de surgir nuevas variedades topológicas, algunos objetos desaparezcan de la escena; etiquetamos como “colapso” a este fenómeno, en los que disminuye el número de objetos. Una referencia para esta parte es el capítulo 4 de [17].

Para simplificar, supondremos inicialmente que los objetos que (des)aparecen no tienen agujeros en ninguna dimensión⁴. Esta hipótesis está justificada por la elección de envolventes cuboidales para el tratamiento computacional en tiempo real de los objetos capturados. En una terminología más matemática, un *colapso* reduce un complejo cuboidal (o, mas generalmente, un CW-complejo) a un subcomplejo equivalente por homotopía. Estas aplicaciones dentro del marco de la homología computacional fueron introducidas y estudiadas inicialmente por J.H.C.Whitehead.

Interesa desarrollar modelos matemáticos y herramientas computacionales (algoritmos) que permitan conectar de manera semi-automática (selección interactiva de opciones) bordes de regiones planares o volumétricas atendiendo a puntos de control en el borde de las regiones y, eventualmente, puntos de control en el interior de dichas regiones. La dificultad del proceso está vinculada a la (des)aparición de elementos cuboidales, dando lugar a contracciones o expansiones de células básicas.

Representación simbólica.

Para tratar de explicar este tipo de proceso realizamos una reformulación de los complejos cuboidales.

Definición 2.6 *Un conjunto $X \subset \mathbb{R}^n$ es **cúbico** si X puede ser escrito como la unión finita de n -cubos.*

Si $X \subset \mathbb{R}^n$ es un conjunto cúbico, entonces adoptamos la siguiente notación:

$$K(X) := \{C(X) \in K \mid C(X) \subset X\}$$

y

$$K_n(X) := \{C(X) \in K \mid \dim(C(X)) = n\}$$

para referirnos a los diferentes niveles de cadenas cuboidales del complejo cuboidal que forma el conjunto cúbico X . Es decir, los elementos de $K_0(X)$ son los vértices, los de $K_1(X)$ son los segmentos, etc. . .

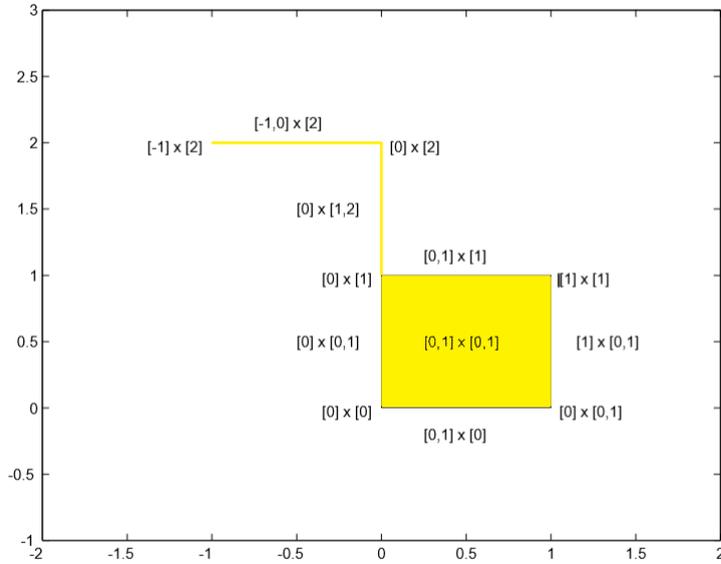
Definición 2.7 *A cualquier $C \in K(X)$ se le llama una **cara de X** y se le denota mediante $C \preceq X$. Decimos que C es una **cara propia de X** , $C \prec X$, si existe $P \in K(X)$ tal que $P \neq C$ y $C \preceq K(P)$. Si C no es una cara propia, entonces es una **cara maximal**. Denotamos por $K_{max}(X)$ al conjunto de caras maximales del complejo. Una cara propia formada por un n -cubo elemental se dice que es una **cara libre**.*

Ejemplo: Sea $X = [0, 1]^3$. Entonces, $K_0(X) \cup K_1(X) \cup K_2(X)$ es el conjunto de las caras propias. El conjunto de caras libres está dado por $K_2(X)$.

Ejemplo: Refiriéndonos al conjunto cúbico de la Figura 2.6, los siguientes cubos elementales son caras libres:

$$[-1] \times [2], [0, 1] \times [0], [0, 1] \times [1], [0] \times [0, 1], [1] \times [0, 1]$$

⁴Esta hipótesis permite aplicar argumentos de retracción

Figura 2.6: Cubos elementales en \mathbb{R}^2 **Simplificando la representación.**

Como se ha mostrado más arriba, incluso los conjuntos cúbicos muy simples contiene un gran número de cubos elementales. Precisamos reducir este número de cubos que es necesario para computar la homología del conjunto.

Lema 2.1 *Sea X un conjunto cúbico. Sea $C \in K(X)$ una cara libre tal que $C \prec P$. Entonces, P no es la cara propia de otro cubo en $K(X)$ y $\dim(C) = \dim(P) - 1$.*

Definición 2.8 *Sea C una cara libre en $K(X)$ y sea P una cara propia de C . Entonces definimos*

$$K'(X) := K(X) \setminus \{C, P\}.$$

donde

$$X' := \bigcup_{R \in K'(X)} R.$$

Entonces X' es un espacio cúbico obtenido a partir de $K(X)$ a través de lo que llamamos un *colapsamiento de C sobre P* .

Ejemplo: Sea $X = [0, 1]^2 \subset \mathbb{R}^2$. Entonces

$$K_2(X) = \{[0, 1] \times [0, 1]\},$$

$$K_1(X) = \{[0] \times [0, 1], [1] \times [0, 1], [0, 1] \times [0], [0, 1] \times [1]\},$$

$$K_0(X) = \{[0] \times [0], [0] \times [1], [1] \times [0], [1] \times [1]\}.$$

En total hay 4 caras libres que se corresponden a los elementos de $K_1(X)$. Sea $C = [0, 1] \times [1]$, entonces $C \prec P = [0, 1] \times [0, 1]$. Si tomamos X' el espacio cúbico obtenido a partir de $K(X)$ a través del colapsamiento de C sobre P se tiene que $X' = [0] \times [0, 1] \cup [1] \times [0, 1] \cup [0, 1] \times [0]$ y

$$K_1(X') = \{[0] \times [0, 1], [1] \times [0, 1], [0, 1] \times [0]\},$$

$$K_0(X') = \{[0] \times [0], [0] \times [1], [1] \times [0], [1] \times [1]\}$$

Nótese que las caras libres de $K(X')$ son diferentes de las de $K(X)$. En particular, $[0] \times [1]$ y $[1] \times [1]$ son caras libres con $[0] \times [1] \prec [0] \times [0, 1]$. Sea X'' el espacio por el colapsamiento de $[0] \times [1]$ sobre $[0] \times [0, 1]$. Entonces,

$$K_1(X'') = \{[1] \times [0, 1], [0, 1] \times [0]\},$$

$$K_0(X'') = \{[0] \times [0], [1] \times [0], [1] \times [1]\}.$$

En $K(X'')$ podemos colapsar $[1] \times [1]$ sobre $[1] \times [0, 1]$ obteniendo así X''' donde

$$K_1(X''') = \{[0, 1] \times [0]\},$$

$$K_0(X''') = \{[0] \times [0], [1] \times [0]\}.$$

Y finalmente podemos colapsar $[1] \times [0]$ sobre $[0, 1] \times [0]$ obteniendo así el punto $X'''' = [0] \times [0]$. Este ejemplo se ilustra en la figura de abajo y muestra que en ciertos espacios podemos reducir un conjunto cúbico a un punto; a este tipo de conjuntos cúbicos los llamamos *plegables*.

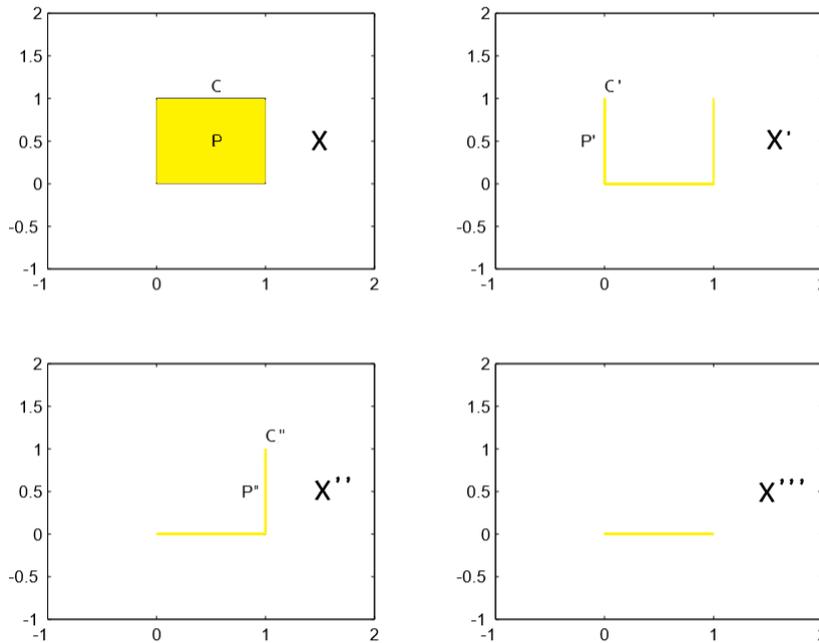


Figura 2.7: Secuencia de colapsamientos de $[0, 1] \times [0, 1]$

Teorema 2.1 Sea X' el conjunto cúbico obtenido a partir de X (conj. cúbico) a través del colapsamiento de un cubo C sobre otro P . Se verifica que

$$H_n(X') \simeq H_n(X).$$

El caso relativo. Invariantes.

Este tipo de fenómenos se pueden describir en también en términos de aplicaciones entre diferentes complejos cuboidales.

Definición 2.9 Una *aplicación cuboidal* $f_K : K(X) \rightarrow K(Y)$ entre dos complejos cuboidales es una aplicación cuya restricción a cada cubo singular es una aplicación entre cubos singulares.

De esta manera, al igual que en la Topología General, se obtiene un criterio de comparación entre complejos cuboidales donde uno guarda información del otro. A partir de ahora denotamos por \hat{K} al complejo cuboidal sobre el que se ha aplicado “colapsamientos”.

Expansión.

El proceso inverso al colapsamiento se conoce como *suspensión* o *extrusión*. Es el fenómeno en el cuál ocurre todo lo contrario, es decir, o bien el número de variedades en la escena aumenta o en los complejos cuboidales pasamos a “expandirlos” e una dirección con la aparición de nuevos cubos en el complejo.

Este proceso se puede describir de varias formas dependiendo del contexto. Las construcciones de tipo prismático se pueden describir en términos de PL-aplicaciones:

Definición 2.10 Diremos que $\varphi : K_1 \rightarrow K_2$ es una *PL-aplicación* si existe una aplicación cuboidal $\hat{\varphi} : \hat{K}_1 \rightarrow \hat{K}_2$.

La condición de PL-aplicación se puede reemplazar por PQ-aplicación (cuadrática a trozos) para disminuir el número de elementos básicos a considerar y mantener baja la complejidad en las operaciones de pegado. Algunos casos particulares importantes corresponden a cilindroides o conoides, de uso común en procesos de comparación de estructuras o de funcionales lineales sobre dichas estructuras en Topología Algebraica.

2.1.4. Representaciones simbólicas.

Grafos de aspecto.

Los *Grafos de aspecto*, \mathcal{G}_s en el marco de la topología, son una representación gráfica desde distintos puntos de vista característicos de un objeto que ha sido ampliamente desarrollada. El término *Aspecto* se refiere al aspecto topológico de un objeto; es decir, desde un punto de vista específico donde podemos observar una partición del espacio (*espacio de partición del aspecto*) en regiones que van alterandose en base a eventos de suspensión y de colapso; en particular las vistas “singulares” corresponden a subcomplejos minimales de la máxima dimensión.

La descripción del aspecto cambiante de subcomplejos maximales pero no coincidentes con el complejo total se conoce también en otras áreas como *Potencia visual* o *Vista característica* (por esta razón, en el marco de la ingeniería, a los grafos de aspecto también se les suele llamar *Grafos de visibilidad*, \mathcal{G}_v).

Los grafos de aspecto se pueden describir para poliedros (no) convexos, una escena de poliedros y objetos no-poliédricos como el toro T^2 . Tienen importantes aplicaciones en Visión por computador (Reconocimiento de poliedros), Procesos de visibilidad (Representación de escenas de poliedros), Imágenes bases de representación, etc. . . . Una aplicación a priori de los grafos de aspecto sería las diferentes formas que puede adoptar un objeto desde diferentes ángulos.

En visión por computador, para el reconocimiento de objetos 3D, el input está dado por una imagen digital que se interpreta en términos de agrupamiento de píxeles con características geométricas o radiométricas similares. Tras la supresión del ruido y la corrección automática de distorsiones, el programa realiza un agrupamiento identificando segmentos “largos” y juntas donde confluyen aristas largas, en relación con las vistas características ó genéricas.

La idea básica para el Reconocimiento de la forma consiste en comparar los datos obtenidos (resultado del agrupamiento) con a vista característica almacenada más próxima. Asimismo, en una fase posterior se puede calcular la transformación que lleva la vista obtenida sobre la vista almacenada más próxima. En ausencia de deformaciones del objeto y presencia sólo de deformaciones aparentes (debidas a la localización relativa de la cámara), esta comparación se lleva a cabo mediante transformaciones afines, habitualmente⁵.

El caso general correspondiente a objetos deformables (personas, p.e.) presenta una complejidad bastante mayor, pues las PL-transformaciones son insuficientes para modelos más finos de reconocimiento de postura, p.e. La diversidad de eventos crece con la complejidad de los objetos a analizar; puede requerir tomar en consideración regiones no-equivalentes desde el punto de vista topológico, debido a oclusiones parciales, presencia de incertidumbre o información incompleta; por ello, el método basado en información procedente de una sola cámara es insuficiente para proporcionar el reconocimiento de un objeto).

Los ejemplos más sencillos de grafos de aspecto corresponden al tetraedro y al cubo que son las piezas básicas de los complejos tetraedrales y cuboidales. Si aplicamos giros a un tetraedro obtenemos diferentes puntos de vista característicos.

⁵El procesamiento de esta información para comparar formas relacionadas se puede paralelizar, lo cual permite acelerar el proceso de comparación de unas formas con respecto a otras.

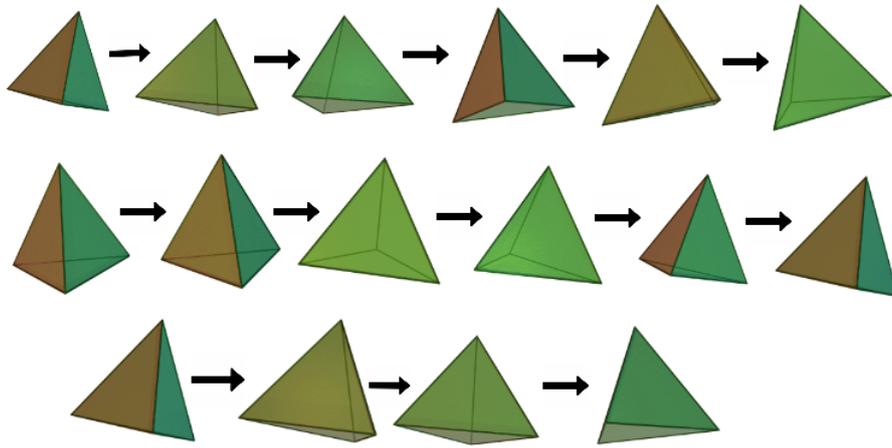


Figura 2.8: Giros sobre el tetraedro

donde es fácil observar los *eventos de colapsamiento* correspondientes a una única cara visible.

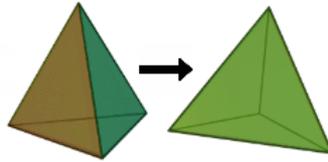


Figura 2.9: Dejamos de ver una de las dos caras

y *eventos de suspensión*

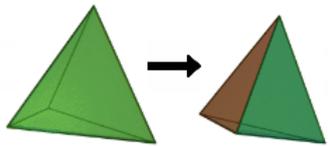


Figura 2.10: Aparece una de las caras que no veíamos

Todos los grafos de aspectos del tetraedro son elementos del siguiente grafo que conocemos como *Grafo de adyacencia* (son grafos sin auto-intersecciones entre las aristas que denotamos por \mathcal{G}_a). De forma más general, se cumple que $\mathcal{G}_s \subset \mathcal{G}_a$.

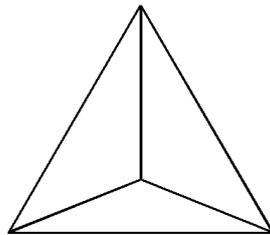
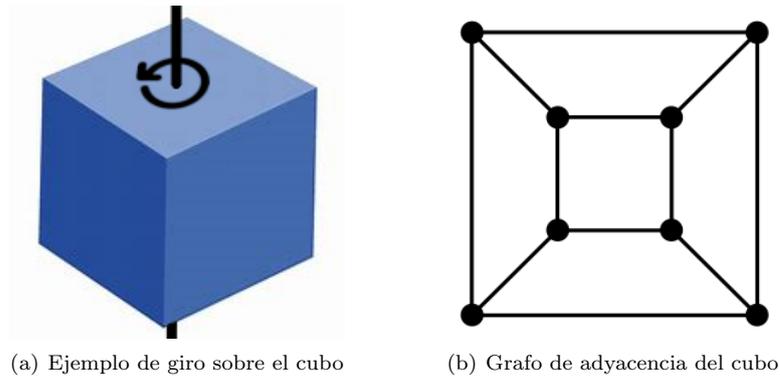


Figura 2.11: Grafo de adyacencia del Tetraedro

El ejemplo de los correspondientes grafos para el cubo es similar al ejemplo del tetraedro aplicando giros al cubo donde se producen los mismo eventos.



Obviamente, en la práctica se presentan fenómenos mucho más complejos debido a las adyacencias entre diferentes elementos (tetraedrales o cuboidales) del PL-complejo utilizado para aproximar los objetos reales. Por ello, es necesario utilizar estrategias de descomposición en unidades más simples que están guiadas por multicaminos interpretables como “cortes virtuales” de la PL-estructura subyacente.

Como *conclusión topológica* para los modelos básicos vemos que los invariantes homológicos de los grafos de aspectos \mathcal{G}_s y \mathcal{G}_a se mantienen al ser los grafos de aspectos transformaciones homeomorfas de los grafos de adyacencia.

En el reconocimiento de objetos más complejos podemos encontrarnos con estructuras poliédricas que dan lugar a *oclusiones* (eventualmente, auto-oclusiones) como las que aparecen en la siguiente figura.

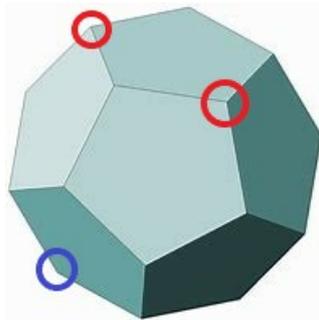


Figura 2.12: Los círculos indican las diferentes oclusiones

El círculo azul señala un punto doble (oclusión en forma de **L**) que son fáciles de describir como confluencia de dos aristas. Los círculos rojos señalan puntos triples (oclusión en forma de **Y** y otra en forma de flecha \rightarrow). El método que se sigue para diferenciar una oclusión de otra es el siguiente: Prolongamos las aristas visibles de forma consecutiva y a continuación aplicamos un barrido rotacional de forma que, girando alrededor del nodo, se identifica las aristas (visibles o prolongadas) que vayamos encontrando. Contando el número de aristas visibles consecutivas es posible diferenciar una juntura triple con respecto a otra⁶.



Grafos de Reeb.

La C^r -estructura de una función continua definida sobre una variedad M se puede hacer explícita visualizando la evolución de las componentes de los conjuntos de nivel. La evolución de los conjuntos de nivel se representa simbólicamente mediante el grafo de Reeb de la función. Tiene aplicaciones en imágenes médicas y otras áreas de ciencia e ingeniería.

⁶Este método se usa en oclusiones de grado mayor.

Los *grafos de Reeb* son un ejemplo muy importante en relación con los grafos de aspecto de una variedad en el espacio; esta descripción se aplica también en dimensiones más altas. La idea es la siguiente:

Consideremos una función de Morse $f : X \rightarrow M$ sobre una 2-variedad como por ejemplo la función altura descrita en apartados anteriores. Los puntos críticos de la función f permiten descomponer nuestro objeto en diferentes regiones obteniéndose así una filtración; esta idea aparecerá más adelante en la parte de “Persistencia” de nuestro trabajo). Los puntos críticos son conocidos como *nodos de Reeb* y, a través de una asignación entre diferentes nodos, nos permite representar un grafo de nuestra variedad.

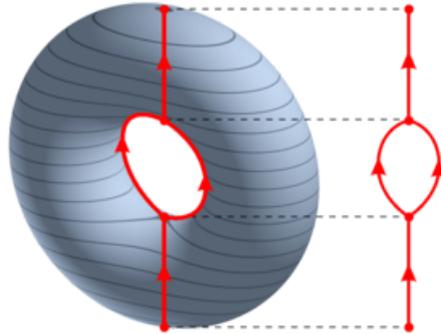


Figura 2.13: Ejemplo de los conjuntos de nivel de una 2-variedad con puntos críticos y la formación de su correspondiente grafo de Reeb.

Para una explicación más detallada y más precisa sobre como funcionan los grafos de Reeb y algunos resultados sobre esta teoría puede consultarse en [11].

2.2. Topología Diferencial Computacional.

La Topología Diferencial Computacional (CDT, *Computational Differential Topology*) es un campo emergente en las técnicas derivadas de diferentes extensiones de Geometría Computacional y Topología Diferencial. Las motivaciones generales para la CDT derivadas de las áreas de ingeniería son comunes a otras áreas de la dinámica computacional en Física o Biología. Afectan al modelado de formas o medios en evolución, el reconocimiento de formas y la personalización adaptable a las necesidades del usuario en aplicaciones multimedia, entre otras áreas.

Una diferencia importante con respecto al enfoque basado en PL-aproximaciones a objetos o aplicaciones radica en el carácter suave de las transformaciones o deformaciones a realizar. Contrariamente a lo que podría parecer a primera vista, este carácter amplía el marco teórico, pues la categoría diferenciable es “densa” (con respecto a la topología compacta-abierta) en la categoría C^r para $1 \leq r < \infty$. Por ello, la categoría diferenciable se adapta mejor que la lineal a trozos a problemas típicos casi universales como los relacionados con optimización y control donde la PL-categoría es bastante menos útil.

El estudio relativo de objetos se lleva a cabo mediante *morfismos* (dados por aplicaciones definidas localmente en el complementario de cerrados) entre estructuras superpuestas como fibrados o, con más generalidad fibraciones. Están dados como pares de aplicaciones entre los espacios base y total de las estructuras que hacen conmutativo el diagrama subyacente. Los morfismos permiten describir información cambiante sobre la evolución de objetos cambiantes o procesos que tienen lugar en el espacio-tiempo correspondiente a la descripción de fenómenos observados.

Por ello y de cara a las aplicaciones, el principal problema a resolver se refiere a la *estimación y caracterización de estructuras*. Para ello, utilizamos funciones (campos escalares) o campos más generales de tipo vectorial o covectorial. Los campos tensoriales están dados como productos formales de campos escalares, vectoriales y co-vectoriales; permiten describir las formas y sus deformaciones, así como “atributos” relativos a la evaluación de funcionales sobre regiones (no necesariamente variedades) del soporte. Las deformaciones

asociadas a cualquier tipo de campos definidos sobre objetos (espacios y aplicaciones) proporcionan pequeños entornos para modelos tentativos de C^r -estructuras y morfismos entre C^r -estructuras.

A primera vista y desde el punto de vista de la ingeniería, las observaciones anteriores parecen demasiado abstractas para ser útiles. Pero esta primera sensación es errónea. Las estrategias de retroalimentación entre modelos y observaciones que aparecen en ingeniería pueden describirse en términos de deformaciones que actúan sobre los espacios de origen y destino para las aplicaciones correspondientes a sus respectivas relaciones de equivalencia. Esta acción se puede entender de forma desacoplada (acción producto o izquierda-derecha) o acoplada (acción de contacto sobre el grafo de la aplicación).

Interesa no sólo detectar características topológicas que sean invariantes por deformaciones, sino identificar la variación en los caracteres geométricos (grado y clase en el caso más sencillo de curvas, p.e.). La persistencia de invariantes topológicos y su relación con caracteres proyectivos para variedades M inmersas en el espacio proyectivo tiene una gran cantidad de aplicaciones. En particular, permiten representar fenómenos dinámicos asociados a la propagación de frentes de ondas como olas, burbujas o formación de arrugas (en la tela o la cara, por ejemplo), o incluso para esbozar un primer acercamiento a muchos fenómenos complejos relacionados con la evolución de las nubes, por ejemplo, desempeñan un papel importante para comparar formas “cercanas”. Esta comparación (sólo resuelta en el caso 1D) tiene gran interés para el reconocimiento semiautomático con respecto en términos de sistemas expertos.

2.2.1. Auto-intersecciones.

Las auto-intersecciones se estudian habitualmente en el contexto geométrico como propiedades que pueden presentar las variedades. Sin embargo, su estudio también es útil en el contexto de campos donde se considera la intersección del soporte de una función con una deformación genérica suya; un ejemplo típico en Topología Diferencial está dado por la autointersección de la sección nula de un fibrado, p.e.

La no-trivialidad de estas construcciones se ilustra con las 2-variedades compactas y *no orientables* sin borde que no admiten embebimiento en \mathbb{R}^3 ; por ello, todos sus modelos se representan con auto-intersecciones en el espacio ordinario. Por el contrario, todas las 2-variedades compactas orientables admiten embebimientos, pero sus modelos pueden aparecer con auto-intersecciones inesperadas aparentes (debidas a oclusiones, p.e.).

Los modelos suaves no presentan auto-intersecciones; por ello, en dicho marco interesa eliminarlas. Sin embargo, los PL-modelos o los modelos algebraicos de objetos presentan habitualmente auto-intersecciones; por ello, es necesario incorporarlas, pues proporcionan relaciones entre las Geometrías Intrínseca y Extrínseca del objeto a modelar.

El caso suave.

Para fijar ideas, consideramos inicialmente variedades M y las aplicaciones $f : M \rightarrow \mathbb{R}^3$ son diferenciables o suaves. A continuación se repasan algunos conceptos básicos.

Definición 2.11 *Sea M una 2-variedad compacta sin borde. Definimos una **carta** como un par (U, ϕ) donde $U \subset M$ es abierto y $\phi : U \rightarrow \mathbb{R}^2$ es un homeomorfismo sobre su imagen. Decimos que dos cartas (U, ϕ) y (V, ψ) son **compatibles** si $U \cap V = \emptyset$ o bien la aplicación*

$$\phi \circ \psi^{-1} : \psi(U \cap V) \rightarrow \phi(U \cap V)$$

se extiende a una función diferenciable de \mathbb{R}^2 a \mathbb{R}^2 .

Definición 2.12 *Una función $f : M \rightarrow \mathbb{R}$ es **diferenciable** si para cada carta (U, ϕ) la composición $f \circ \phi^{-1}$ es diferenciable. Una aplicación $f : M \rightarrow \mathbb{R}^3$ es diferenciable si cada una de las funciones $f_i = \pi_i \circ f$, $\forall i \in \{1, 2, 3\}$, también lo son (π_i denota la aplicación proyección sobre la i -ésima componente).*

Con una carta tenemos una parametrización local de M con dos variables x_1 y x_2 . Junto con la aplicación $f : M \rightarrow \mathbb{R}^3$ obtenemos la *matriz Jacobiana* de f :

$$J_f = \begin{pmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \\ \partial f_3/\partial x_1 & \partial f_3/\partial x_2 \end{pmatrix}$$

La descripción anterior motiva la siguiente definición.

Definición 2.13 La aplicación f es una *inmersión* si su diferencial tiene rango máximo en todos sus puntos; en coordenadas locales, la diferencial está dada localmente por la matriz jacobiana J_f . Decimos que f es un *embebimiento* si f es una inmersión inyectiva y propia sobre su imagen.

Un embebimiento es necesariamente una inmersión pero no viceversa.

Los embebimientos no tienen auto-intersecciones, y las inmersiones solo tienen los dos primeros tipos; es decir, no tienen puntos de ramificación.

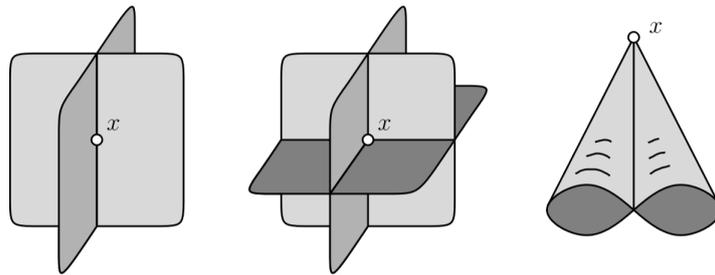


Figura 2.14: Punto doble, punto triple y punto de ramificación

Los puntos dobles y triples han aparecido previamente en relación con juntas de interés para el Reconocimiento de objetos y de escenas; la más novedosa a efectos diferenciales es la correspondiente al punto de ramificación que puede aparecer de formas distintas. Un ejemplo muy conocido es el *paraguas de Whitney* cuya construcción se ilustra en la siguiente figura.

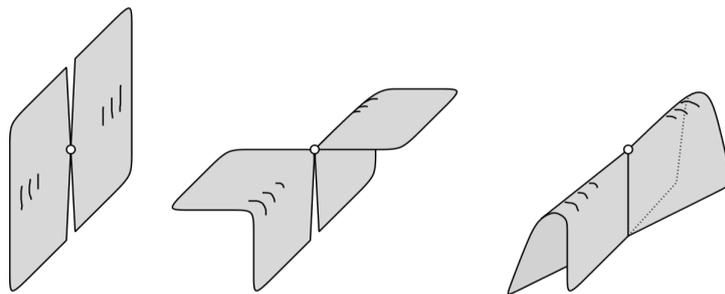


Figura 2.15: Se corta un disco por la mitad menos el punto central y después se pega

Una aplicación a complejos cuboidales.

Frecuentemente, interesa comparar PL- y PS-aproximaciones a objetos o aplicaciones. En nuestro caso, las OK-aproximaciones de tipo cuboidal son las más cercanas al enfoque basado en cartas utilizado en Geometría y Topología Diferencial.

La clasificación genérica de auto-intersecciones es similar en el caso PL donde M viene dada como un complejo cuboidal, K . En la parte de **Estructuras y deformaciones PL-superpuestas** se ha descrito la construcción de un complejo; sin embargo, en el reconocimiento de objetos pueden aparecer fenómenos de auto-intersección que es necesario incorporar en las fases previas de modelado.

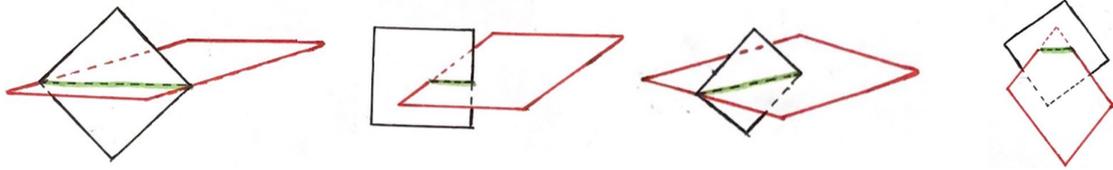


Figura 2.16: Tipos de cruces de dos 2-cubos

Consideremos un complejo K inmerso en el espacio ambiente asignando cada vértice del cuadrado a un punto en \mathbb{R}^3 . Las aristas y cuadrados correspondientes al borde forman parte de la envolvente convexa de las imágenes de sus vértices. Esta asignación es un embebimiento si dos cuadrados cualesquiera son o bien disjuntos, o bien comparten un vértice, una arista o bien comparten una cara (frontal o trasera). Cualquier otro tipo de intersecciones es inapropiada y se conoce como *cruzamiento*. Es conveniente suponer que los puntos están en posición general; en nuestro caso, no hay tres que sean colineales y no hay cuatro que sean coplanares. Bajo esta hipótesis, sólo hay tres tipos de cruces posibles entre dos cubos que son los que se muestran en la figura precedente.

La extensión de este enfoque al caso dual (funcionales sobre complejos cuboidales) está justificada por el Teorema de Transversalidad de Thom en espacio de jets. Cabe suponer que el desarrollo de un enfoque “estratificado” para las cuestiones de clasificación utilizará este resultado, pero aún es necesario desarrollar aspectos básicos de dicha “estratificación” en el marco del Reconocimiento. Por ello, no profundizamos en esta cuestión, centrando la atención en el estudio de representaciones simbólicas para los complejos graduados introducidos.

2.2.2. Persistencia.

La idea básica de esta subsección consiste en reducir la información mediante la supresión o minimización de los datos “innecesarios” ó redundantes en tareas de reconocimiento. Para ello, debemos resolver problemas que afectan al muestreo (selección de los datos más significativos), eliminación de información redundante ó fuera de umbral (incluyendo *outliers*) y asignación de formas según jerarquías.

No existe una estrategia única para alcanzar estos objetivos y su implementación computacional puede presentar una gran dificultad; en particular, la diferencia entre lo necesario e innecesario no está bien definida. En cualquier caso, se adopta un enfoque multiescala en el que se ignora lo que está fuera del rango de dicha escala (por ser “más pequeño” o “más grande”). En otras palabras, seleccionamos un “tamaño significativo” (dado por una métrica) y una “unidad” de comparación para objetos o procesos.

La elección de los elementos de referencia tampoco es unívoca y puede estar vinculada a aspectos cualitativos (variedades o aplicaciones deformables), estadísticos (distribuciones permitidas) ó métricos (diferentes tipos de métrica sobre tipos particulares de objetos o de aplicaciones⁷). Para abordar estas cuestiones se introduce la noción de “persistencia”.

El concepto de Persistencia.

La *homología persistente* se puede utilizar para medir la “escala” de objetos o la “resolución” de aplicaciones entre estructuras asociadas a una característica topológica. Hay dos “ingredientes”: Uno geométrico o con más precisión topológico (asociado a una función definida sobre una variedad o, con más generalidad, un espacio topológico X) y otro analítico (que convierte la función en mediciones). Las mediciones solo tienen sentido si la función lo tiene.

Sea X un espacio topológico conexo y $f : X \rightarrow \mathbb{R}$ una función continua. El flujo acumulado asociado a los conjuntos de nivel $f^{-1}(r)$ de f (introducidos en **Homotopía en términos de valores críticos**) se interpreta en términos de un grupo uniparamétrico de subespacios anidados, $X_a \subseteq X_b$ ($a \leq b$). Es conveniente escribir sobre esta familia como si se tratara de un conjunto de subniveles que evoluciona a medida que aumenta el umbral.

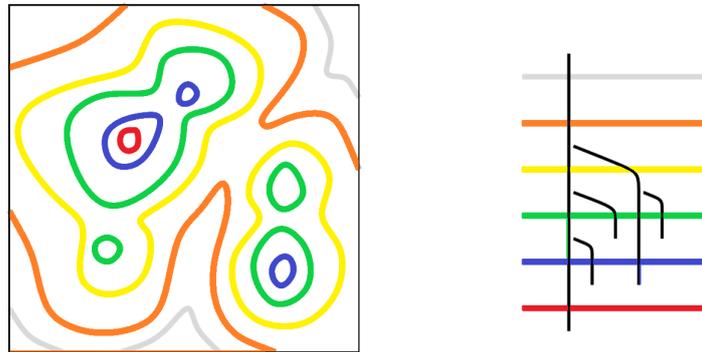
Visualizamos la evolución de los subniveles dibujando cada componente de X_a como un punto. El resultado es un gráfico 1-dimensional, G_f . Si se interpreta f como si fuera una “función de altura”, dibujamos el gráfico de abajo hacia arriba. Como las componentes nunca

⁷En general, ninguno de ellos es un espacio metrizable

se deforman, los arcos del gráfico se pueden fusionar, pero nunca se dividen. Al final, para un umbral lo suficientemente grande a , tenemos una sola componente. Por tanto, G_f es un árbol y lo llamamos *árbol de fusión de la función f* .

A continuación, descomponemos el árbol de fusión en sendas disjuntas que aumentan monótonamente con f . Para obtener el trazado, lo dibujamos de abajo hacia arriba, simultáneamente, manteniendo sus puntos finales superiores a la misma altura, a . El trazado se extiende hasta completar los subniveles no vacíos cuando M es compacta; sin embargo, cuando se fusionan, terminamos el que comenzó más tarde. Si interpretamos la diferencia entre dos valores de función como si fuera la “edad”, damos prioridad a la ruta más antigua.

Ejemplo: En la figura de la izquierda se tiene una función en la unidad cuadrada visualizada que se obtiene dibujando 6 conjuntos de nivel con diferentes colores que indican valores más grandes; mientras que, en la figura de la derecha tenemos la descomposición de la ruta del árbol de combinación de la función.



Este ejemplo se puede entender de otra manera distinta, dando la vuelta a la imagen de la derecha e interpretándola como un mapa topográfico donde el inicio de cada uno de los caminos se correspondería con los picos de las montañas de una cordillera.

Para gestionarla información seguimos la siguiente regla: “El más antiguo de los dos caminos de fusión continúa y el camino más joven termina”, esta “regla” se conoce como *regla del anciano*.

Sean $a \leq b$ dos umbrales y denotamos por $\beta(a, b)$ el número de componentes en X_b que tienen a intersecciones no vacías con X_a . En términos del árbol de fusión, este es el número de subárboles con puntos más arriba en el valor de b que alcanzan hasta el nivel a o por debajo. Cada uno de estos subárboles tiene un camino único, el más largo, que abarca todo el intervalo $[a, b]$. De esto se sigue que $\beta(a, b)$ es también el número de trayectorias en la descomposición del trayecto de G_f que abarca $[a, b]$.

Nótese que cualquier descomposición del camino que no se genere usando la *regla del anciano* no tiene esta propiedad. En particular, si f es una función de Morse, entonces la *regla del anciano* genera una descomposición de trayecto única, que es la única para la cual el número de trayectorias que abarcan $[a, b]$ es igual a $\beta(a, b)$ para todos los valores del intervalo.

Este proceso es similar al presentado en la Teoría de Morse, donde la función f es la función altura y los subniveles son los diferentes $M_a := \{p \in M | f(p) \leq a\}$, donde ahora M sería el espacio con el que trabajamos y a es el umbral.

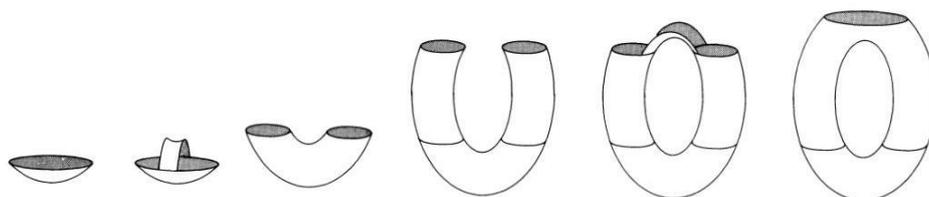


Figura 2.17: Diferentes subniveles del Toro.

Filtraciones y Homología persistente.

Ahora estamos interesados en la evolución topológica de la cadena de complejos. Para ello, utilizamos la noción de *filtración*: Como cada nivel de filtración tiene su propia homología interesar ver como podemos relacionar unas con otras. Para ello, se introduce una secuencia correspondiente a los grupos de homología que presentamos a continuación. Para cada $i \leq j$, tenemos que la aplicación *inclusión* $K^i \hookrightarrow K^j$ induce un homomorfismo

$$f_n^{(i,j)} : H_n(K^i) \rightarrow H_n(K^j).$$

Por tanto, la filtración de un complejo permite obtener la siguiente cadena de grupos de homología que muestra una “evolución” en términos de las n -cadenas de K que nacen y mueren:

$$0 = H_n(K^0) \rightarrow H_n(K^1) \rightarrow \dots \rightarrow H_n(K^n) = H_n(K).$$

A medida que nos desplazamos de K^i a K^{i+1} , existe la posibilidad de obtener nuevas clases de homología y de perder algunas cuando se vuelven triviales o si se fusionan entre sí. Esta idea es similar a la de los ciclos evanescentes (Lefschetz) que aparecen en la Topología de las Variedades Complejas. La idea básica consiste en agrupar las clases que nacen “en” o “antes de” un umbral dado y las que mueren después de otro umbral en los grupos de homología⁸. De este modo, se obtiene un primer modelo topológico, cuya representación simbólica permite gestionar eventos vinculados a la (des)aparición de objetos a lo largo de procesos.

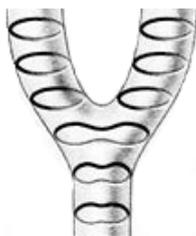


Figura 2.18: Nacimiento de nuevos ciclos.

Definición 2.14 Los n -ésimos grupos de homología persistente son las imágenes de los homomorfismos inducidos por la aplicación inclusión, es decir,

$$H_n^{(i,j)} = \text{Im } f_n^{(i,j)}, \quad \forall 0 \leq i \leq j \leq n.$$

De forma similar a como describimos la homología reducida en el primer capítulo, definimos los *grupos de homología persistentes reducidos*⁹. Básicamente, los grupos de homología persistente son las clases de homología de K^i que todavía viven en K^j , es decir,

$$H_n^{(i,j)} = \frac{Z_n(K^i)}{(B_n(K^j) \cap Z_n(K^i))}.$$

Podemos especificar más en relación con las clases contadas por los grupos de homología persistente. Por ello, debemos especificar los procesos de “muerte de cadenas” y “nacimiento de cadenas”.

Definición 2.15 Sea $\gamma \in H_n(K^i)$ una clase.

1. γ nace en K^i si $\gamma \notin H_n^{(i-1,i)}$.
2. Si γ nace en K^i , entonces decimos que *muere entrando en K^j* si se fusiona/une con una clase más antigua/anterior a medida que vamos de K^{j-1} a K^j (es decir, $f_n^{(i,j-1)}(\gamma) \notin H_n^{(i-1,j-1)}$ pero $f_n^{(i,j)}(\gamma) \in H_n^{(i-1,j)}$).

⁸Los superíndices de los complejos de una filtración también suelen llamarse *tiempos*, ya que puede pensarse que la filtración muestra cómo nace y se desarrolla el objeto hasta convertirse en su “total” K .

⁹Nótese que $H_n^{(i,i)} = H_n^i = H_n(K^i)$ y $H_n^{(i,j)} \subseteq H_n(K^j)$.

El criterio que se acaba de mostrar es una variante de la *regla del anciano*: Cuando dos clases se unen, la clase que *persiste* es la más antigua.

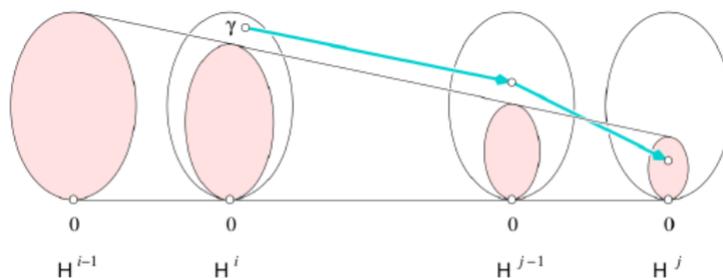


Figura 2.19: La clase γ nace en K^i y muere entrando en K^j .

Definición 2.16 Con la notación anterior, decimos que una clase γ que nace en K^i y muere en K^j tiene un *índice de persistencia*

$$Pers(\gamma) = j - i.$$

Si γ nunca muere decimos que $Pers(\gamma) = \infty$.

Los correspondientes n -ésimos números de Betti persistentes son los rangos de los correspondientes grupos de homología (se definen igual que en el resto de homologías), es decir,

$$\beta_n^{(i,j)} := \text{rg}(H_n^{(i,j)}).$$

Diagramas de persistencia.

Los *diagramas de persistencia* y los *códigos de barras* son las dos formas más habituales de presentar el resultado de la homología persistente sobre una filtración. En este trabajo tratamos el caso de los diagramas de persistencia que permiten codificar la diferencia entre las homologías de la filtración de un complejo. De este modo, es posible visualizar toda la información de estos grupos.

A las clases de nivel n^{10} que nacen en i y mueren en j las llamamos la *generación de i a j* . Con esto damos paso a la siguiente definición.

Definición 2.17 Llamamos *multiplicidad de la generación de i a j* al número de clases en la generación de i a j de nivel n , $\mu_n^{(i,j)}$.

La multiplicidad de una generación está relacionada con los números de Betti persistentes. Lo cual nos da un criterio para interpretar la evolución de los ciclos de cada nivel. Más formalmente:

$$\mu_n^{(i,j)} = (\beta_n^{(i,j-1)} - \beta_n^{(i,j)}) - (\beta_n^{(i-1,j-1)} - \beta_n^{(i-1,j)}).$$

De los diagramas de persistencia debemos ser capaces de extraer la siguiente información:

- La persistencia de las distintas clases.
- Las multiplicidades de las generaciones.
- Los números de Betti persistentes.

Para ello, necesitamos poder obtener información que de alguna manera sea invariante, es decir, que dependa de lo menos posible de la filtración.

La construcción de estos diagramas es sencilla y requiere seguir una serie de pasos:

1. Fijar un nivel $0 \leq p \leq n$.
2. Calcular qué generaciones de i a j no son vacías.

¹⁰Hace referencia al subíndice del grupo de homología.

3. Para cada una de ellas calcular su multiplicidad de $\mu_n^{(i,j)}$.
4. Marcar los puntos $(i, j) \in \mathbb{R} \times (\mathbb{R} \cup \{\infty\})$ y ponerle un $\mu_n^{(i,j)}$ para indicar su multiplicidad.

El objeto obtenido se le conoce como *Diagrama de persistencia de nivel p* y se le denota por $Diag_p(\mathfrak{S})$ donde \mathfrak{S} es la filtración asociada al complejo K .

Teorema 2.2 (Lema fundamental de la Homología persistente)

Sea $\emptyset = K^0 \subseteq K^1 \subseteq \dots \subseteq K^n = K$. Para cada par de índices $0 \leq p < q \leq n$ y cada nivel $0 \leq m \leq n$ del grupo de homología se tiene que el m -ésimo número de Betti persistente de $H_m^{(p,q)}$ es

$$\beta_m^{(p,q)} = \sum_{i \leq p} \sum_{j > q} \mu_m^{(i,j)}$$

Esta es una de las propiedades más importantes que nos dice que los diagramas de persistencia codifican toda la información sobre los grupos de homología persistente.

Persistencia extendida.

En este apartado se discute una extensión de la persistencia que está motivada por un enfoque para encajar formas entre sí, lo cual permite la reconstrucción del objeto. Una ilustración del problema de encajar las formas está dada por la resolución de un rompecabezas.

En términos de la Teoría de Morse, sea M un embebimiento de una 2-variedad diferenciable en \mathbb{R}^3 . Dada una dirección $\underline{u} \subset \mathbb{S}^2$, la *función altura en la dirección \underline{u}* , $f : M \rightarrow \mathbb{R}$, está definida por $f(\underline{x}) = \langle \underline{x}, \underline{u} \rangle$. Anteriormente tomábamos \underline{u} como la dirección vertical y se utilizaba la altura como la distancia que hay desde el corte de nivel por un plano hasta la base como se muestra en la siguiente figura.

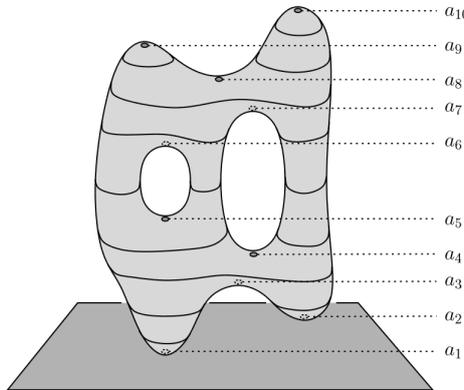


Figura 2.20: Puntos críticos de la función altura de la variedad.

Al igual que en la sección 1.3 de la memoria, consideramos $M_a = f^{-1}((-\infty, a])$ donde $a \in \mathbb{R}$ es el umbral del conjunto de subnivel. Estos conjuntos están anidados y son los que definen la persistencia del objeto a través de la secuencia de los grupos de homología que hemos visto anteriormente. En general, en las variedades diferenciables los valores críticos homológicos de una función de altura son los valores de altura de los puntos críticos aislados. Si, además, la dirección es genérica, entonces sólo hay tres tipos diferentes:

- *Mínimos* que inician las componentes.
- *Puntos de silla* que fusionan componentes.
- *Máximos* que rellenan/finalizan la componente.

Suponiendo que los puntos críticos se encuentran a diferentes alturas, los puntos de los diagramas de persistencia de f corresponden a pares de puntos críticos. La elevación en los puntos x e y de un par de este tipo se traduce como $|f(x) - f(y)|$. Suponiendo que x es un punto crítico para ambas direcciones opuestas, tenemos que asegurarnos de que

el emparejamiento es el mismo en ambas direcciones; de lo contrario tenemos asignaciones contradictorias de elevación. También necesitamos que todos los puntos críticos estén emparejados, pues de lo contrario se obtienen “áreas vacías”, es decir, áreas en las que la elevación permanece indefinida. Esta última es la razón por la que extendemos la persistencia y la primera es una restricción que debemos observar en esta extensión.

Definición 2.18 Sean $a_1 < a_2 < \dots < a_m$ los valores críticos de la función altura f . Consideramos valores intermedios

$$b_0 < a_1 < b_1 < a_2 < \dots < a_m < b_m,$$

donde los subniveles $M_{b_i} = f^{-1}((-\infty, b_i])$ son 2-variedades con borde. De forma simétrica, consideramos las 2-variedades complementarias a los subniveles con el mismo borde, $M^{b_i} = f^{-1}([b_i, +\infty))$, que llamamos *superniveles*. Finalmente, junto con la aplicación de inclusión, usamos ambos para construir una secuencia de grupos de homología “ascendentes” y otra secuencia de grupos de homología relativa “descendentes”:

$$0 = H_n(M_{b_0}) \rightarrow \dots \rightarrow H_n(M_{b_m});$$

$$H_n(M, M^{b_m}) \rightarrow \dots \rightarrow H_n(M, M^{b_0}) = 0,$$

A esta construcción se la conoce como *filtración extendida*.

El recorrido del grupo trivial al grupo trivial se interpreta diciendo que “todo lo que nace eventualmente muere”. Como consecuencia, todos los nacimientos se emparejan con las muertes correspondientes, según se desee.

Rastrear lo que nace y muere en los grupos de homología relativa es menos intuitivo que para los grupos de homología absoluta que van hacia arriba. Sin embargo, podemos traducir los eventos entre la homología absoluta de M^{b_i} y la homología relativa del par (M, M^{b_i}) . Bajando, el umbral disminuye, por lo que el conjunto de supernivel crece.

Diremos que una clase de homología en el conjunto de superniveles es *esencial* si vive todo el camino hasta el umbral b_0 . Aunque para evitar anomalías, debemos seguir unas reglas:

1. Una clase de $H_n(M^{b_i})$ muere al mismo tiempo que una clase de $H_{n+1}(M, M^{b_i})$ muere.
2. Una clase no esencial de $H_n(M^{b_i})$ nace al mismo tiempo que una clase de $H_{n+1}(M, M^{b_i})$ nace.
3. Una clase esencial de $H_n(M^{b_i})$ nace al mismo tiempo que una clase de $H_n(M, M^{b_i})$ muere.

Las relaciones precedentes se demuestran estudiando el núcleo y el conúcleo de las aplicaciones entre los grupos de homología de M^{b_i} y M^{11} .

2.2.3. Estabilidad.

La persistencia es un concepto de teoría de la medida construido sobre estructuras algebraicas. Su propiedad más importante es la estabilidad con respecto a perturbaciones en los datos. En otras palabras, pequeños cambios en los datos implican pequeños cambios en la persistencia medida. En esta subsección estudiamos cómo el cambio continuo de los datos afecta a la persistencia.

Para ello, adoptamos un enfoque basado en deformaciones genéricas que pueden afectar a cada uno de los tipos de campos. Sería interesante disponer de representaciones simbólicas asociadas a los tipos que pueden aparecer en baja codimensión para facilitar el reconocimiento semi-automático, pero el estado actual de conocimientos presenta aún serias limitaciones en este terreno.

¹¹Los diagramas/subdiagramas de persistencia extendida/relativa junto con el teorema de la dualidad persistente proporcionan una herramienta muy importante en este estudio de casos relevantes para el problema del Reconocimiento.

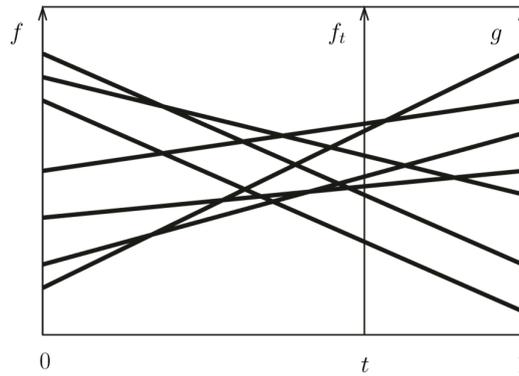
Familias uniparamétricas de funciones.

Sea $f : K \rightarrow \mathbb{R}$ y $g : K \rightarrow \mathbb{R}$ dos funciones monótonas¹² sobre complejos. Definimos la homotopía por línea-recta $F : K \times [0, 1] \rightarrow \mathbb{R}$ mediante

$$F(\sigma, t) = f_t(\sigma) = (1 - t)f(\sigma) + tg(\sigma),$$

que interpola f con g . Como f_t también es monótona, podemos encontrar una ordenación compatible de los cubos singulares, es decir, un orden total que amplía las órdenes parciales definidas por f_t . Con este orden, calculamos los diagramas de persistencia de f_t .

Sin embargo, si ya tenemos el diagrama para f , entonces podemos considerar modificaciones para obtener el diagrama de f_t . Esto resulta más eficaz que volver a recalcular los diagramas, siempre y cuando los dos no sean demasiado diferentes. Para describir esto, trazamos los valores de la función con el tiempo; esta construcción proporciona una línea recta para cada cubo singular).



Es conveniente suponer que tanto f como g son inyectivas porque esto implica que f_t lo es excepto en ciertos momentos t cuando dos o más líneas se cruzan. Para simplificar esta situación, supondremos que no hay dos pares diferentes de líneas cruzadas en el mismo momento. Equivalentemente, como máximo dos cubos singulares tienen el mismo valor. Haciendo un barrido de izquierda a derecha en la dirección de incremento de t , vamos pasando por los diferentes puntos dobles mediante la transposición de dos cubos singulares. Esto motiva a estudiar el impacto de una transposición sobre la Persistencia.

Familias uniparamétricas de campos.

La trayectoria $\gamma_i(t)$ llevada a cabo por un agente a_i se puede interpretar como la curva integral de un campo vectorial X_0 . Una perturbación sobre el campo vectorial X_0 en la dirección de otro campo vectorial X_j (genéricamente transversal) se puede describir en términos de una deformación uniparamétrica $(1 - \lambda)X_0 + \lambda X_1$ del campo inicial X_0 . Por ello, inicialmente podríamos esperar que la curva integral genérica $\gamma_0(t)$ solución de X_0 se deformara en la curva integral genérica $\gamma_1(t)$ solución de X_1 sobre un trozo de superficie dado localmente por el grafo Γ_{01} asociado al camino producto $\Gamma_0 \times \gamma_1$.

Interpretemos la trayectoria $\gamma_0(t)$ en términos del aprendizaje de una tarea llevada a cabo por un agente inicial a_0 . Una perturbación de grado bajo ≤ 2 a lo largo de un camino γ_1 con los mismos extremos y genéricamente transversal, permite ajustar la realización de dicha tarea de acuerdo con un funcional de energía asociado a la vibración local de una lámina ideal definida asociada al producto $\gamma_0 \times \gamma_1$. Esta idea tiene interés para el modelado del aprendizaje individual en relación con la realimentación interna en el Ciclo Percepción-Acción (PAC: Perception-Action Cycle) que se aborda en el último capítulo.

Asimismo, la interacción entre dos agentes a_0 y a_1 se puede interpretar en los puntos de "coincidencia" (máxima proximidad entre trayectorias) en términos de una secuencia de acoplamientos (para distancia mínima y orientación óptima) vs desacoplamientos de los campos vectoriales asociados. Esta interpretación se puede modelar en términos de propiedades de los correspondientes campos asociados y de funciones de energía asociadas tanto a cada campo por separado como a su transferencia en el momento del acoplamiento. Esta idea tan simple tiene interés para el modelado del aprendizaje colaborativo por reforzamiento, p.e.

¹²Significa que ofrece un orden parcial, es decir, si tenemos una secuencia de cubos singulares T_1, \dots, T_m entonces $f(T_i) < f(T_j)$ (para $i < j$) o se da el caso de que T_i es una cara de T_j .

Familias uniparamétricas de formas.

Frecuentemente, el espacio de parámetros asociado a un modelo tiene una dimensión muy elevada o la mayor parte de los datos que lo determinan no son observables en tiempo real¹³. Sin embargo, es posible observar algún atributo o propiedad significativa que evaluamos de forma numérica. La variación de este valor numérico en el espacio-tiempo se representa mediante una forma lineal $H : V \rightarrow \mathbb{R}$ definida sobre un espacio vectorial V a valores en \mathbb{R} .

La versión diferencial de esta construcción sobre una variedad M se representa mediante una 1-forma ω cuyo soporte es un hiperplano H_p (lugar de anulación de la forma lineal H comentada más arriba) dependiente de $p \in M$. Formalmente, la 1-forma ω_U es una sección local del fibrado cotangente Ω_M^1 ; por ello, es la dual de un campo vectorial X (sección local del fibrado tangente τ_M) sobre M .

De la misma forma que se ha mostrado en el apartado anterior, podemos considerar una perturbación genérica de la 1-forma inicial ω_0 en términos de otra 1-forma ω_1 que sea genéricamente transversal.

Para controlar de manera más eficiente la deformación podemos suponer que la intersección de los dominios de cada forma es un subespacio de codimensión 1 para el soporte de cada forma en cada punto.

Familias uniparamétricas de tensores.

Un tensor $t^{r,s}$ de tipo (r,s) es un producto formal de r campos vectoriales y s formas diferenciales definidos en cada punto $p \in M$. Se pueden visualizar como una colección de r líneas (tangentes a r trayectorias en $p \in U \subset M$) y una colección de s hiperplanos (formas lineales definidas sobre el espacio tangente $T_p(M)$) que varían con el punto base $p \in M$. Desde el punto de vista local están dados por la restricción a un abierto $U \subset M$ del producto de una distribución \mathcal{D} generada por r campos vectoriales y un sistema \mathcal{S} generado por s formas diferenciales.

Existe un número creciente de aplicaciones en Ingeniería donde se utilizan tensores para representar “cantidades mixtas” de tipo multilineal. Su estimación se lleva a cabo mediante procedimientos de “votación tensorial” (tensor voting). En estos procedimientos, se presentan varios candidatos entre posibles tensores en relación con la resolución de un problema y se elige el que tiene “mayor número de votos” desde el punto de vista de compatibilidad con un mayor número de muestras para su determinación unívoca.

Un *ejemplo típico* es la estimación de la matriz fundamental \mathbf{F} correspondiente a la restricción estructural ${}^T \mathbf{p}_i \mathbf{F} \mathbf{p}'_i = 0$ para cada par de candidatos a puntos homólogos $(\mathbf{p}_i, \mathbf{p}'_i)$ en un par de vistas, es decir, verificando $\pi(\mathbf{P}_i) = \mathbf{p}_i$ y $\pi'(\mathbf{P}_i) = \mathbf{p}'_i$. Para determinar \mathbf{F} de forma unívoca se utiliza un algoritmo lineal de 8 puntos debido a su rapidez en la convergencia. Por ello, la estimación tipo RanSaC (Random Sampling Consensus) de \mathbf{F} utiliza 8-tuplas de candidatos a homólogos en “posición general”. Para una estimación de \mathbf{F} se evalúa el resultado para 8-tuplas de candidatos a homólogos; se elige la primera estimación de \mathbf{F} que presenta resultados satisfactorios para un 60% de las 8-tuplas candidatas. A pesar del carácter aparentemente enrevesado del procedimiento es el más estable y robusto para la estimación de la matriz fundamental.

En general, como $\det(\mathbf{F}) = 0$, uno de los autovalores de \mathbf{F} es nulo, lo cual permite llevar a cabo procedimientos de “redondeo” en las proximidades del autovalor nulo para \mathbf{F} . Sin embargo, el procedimiento RanSaC se vuelve inestable cuando la matriz \mathbf{F} tiene rango 1 en lugar del rango esperado 2. Esto ocurre p.e. si los 8 puntos de una de las 8-tuplas están contenidos en un plano de la escena (parte trasera de un camión, p.e. en una escena de tráfico). Una estrategia para resolver este problema consiste en generar de forma automática una perturbación genérica $(1 - \lambda)t_0^{r,s} + \lambda t_1^{r,s}$ sobre el tensor inicial $t_0^{r,s}$ que permita recuperar las condiciones de genericidad sobre el tensor asociado a la Reconstrucción 3D.

En este caso, no nos preocupamos de mantener las condiciones de integrabilidad para el producto tensorial de la distribución \mathcal{D} de r campos vectoriales X_i y del sistema \mathcal{S} de s formas diferenciales ω_j . Ello se debe a que una perturbación genérica de una distribución ó de un sistema integrable en general no es integrable. En términos estadísticos, la probabilidad

¹³Una gran cantidad de modelos estadísticos utilizados para estimación son no-paramétricos, situación en la que no podemos aplicar directamente los métodos basados en campos vectoriales (incluso aunque el modelo sea “regular”), pero sí las formas diferenciales.

de que la perturbación de una distribución o un sistema integrable siga siendo integrable es nula. Por ello, mantener la condición de integrabilidad en modelos experimentales es poco realista. En otras palabras, hay que poner énfasis en la estabilidad (y, por consiguiente en soluciones aproximadas) y no en la integrabilidad (soluciones exactas).

Además, hay un gran número de distribuciones o sistemas que funcionan con una elevada estabilidad, incluso aunque no conozcamos las soluciones de forma exacta. Un ejemplo típico está dado por el problema de tres cuerpos (Sol, Tierra, Luna, p.e.) que no es integrable y, sin embargo, es estable. A medida que aumenta el número de cuerpos (agentes móviles) verificar condiciones de integrabilidad se vuelve un problema prácticamente imposible de resolver. En consecuencia, la *cuestión crucial* no es calcular la solución de forma exacta, sino *garantizar la estabilidad* para distribuciones o sistemas suficientemente generales.

Teoremas de Estabilidad.

La Persistencia nos sirve como un dispositivo de medición y los diagramas de Persistencia son los que nos permiten leer estas medidas y establecer ciertas similitudes.

Recordemos que un diagrama de Persistencia es un conjunto de múltiples puntos en el plano \mathbb{R}^2 . Bajo las suposiciones que hacemos en este trabajo, el diagrama presenta varios puntos por encima de la diagonal. Para este multi-conjunto finito añadimos infinitos puntos en la diagonal, cada uno con una multiplicidad infinita. Estos puntos no son esenciales para el diagrama, pero su presencia simplifica las próximas definiciones y resultados.

Definición 2.19 Sean \mathcal{X} y \mathcal{Y} dos diagramas de Persistencia. Consideramos la biyección $\eta : \mathcal{X} \rightarrow \mathcal{Y}$ y registramos el supremo de las distancias entre los puntos correspondientes a cada uno. Midiendo la distancia entre puntos $\underline{x} = (x_1, x_2)$ e $\underline{y} = (y_1, y_2)$ con la norma infinito $\|\underline{x} - \underline{y}\|_\infty = \max\{|x_1 - y_1|, |x_2 - y_2|\}$ y tomando el ínfimo de todas las biyecciones η , obtenemos la *distancia de cuello-de-botella*¹⁴ entre los diagramas:

$$W_\infty(\mathcal{X}, \mathcal{Y}) = \inf_{\eta: \mathcal{X} \rightarrow \mathcal{Y}} \sup_{\underline{x} \in \mathcal{X}} \|\underline{x} - \eta(\underline{x})\|_\infty.$$

Como se ilustra en la siguiente figura, podemos dibujar cuadrados de longitud lateral dos veces mayor que la distancia de cuello-de-botella centrada en los puntos de \mathcal{X} de manera que cada cuadrado contenga el punto correspondiente de \mathcal{Y} .

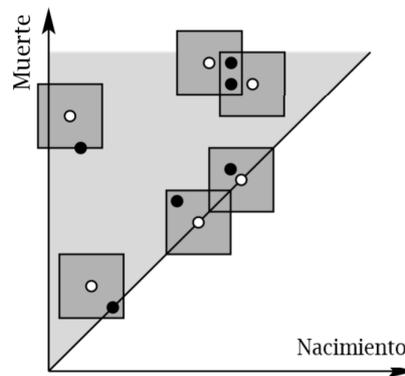


Figura 2.21: La superposición de dos diagramas de persistencia consiste en los puntos blancos y negros. Sólo los puntos en la diagonal corresponden a puntos fuera de la diagonal en el otro diagrama. La distancia de cuello-de-botella es la mitad de la longitud lateral de los cuadrados que ilustra la aplicación biyectiva.

Proposición 2.2 Si \mathcal{X} , \mathcal{Y} y \mathcal{Z} son diagramas de Persistencia, se verifica que

- $\mathcal{X} = \mathcal{Y} \Leftrightarrow W_\infty(\mathcal{X}, \mathcal{Y}) = 0$.
- $W_\infty(\mathcal{X}, \mathcal{Y}) = W_\infty(\mathcal{Y}, \mathcal{X})$.
- $W_\infty(\mathcal{X}, \mathcal{Z}) \leq W_\infty(\mathcal{X}, \mathcal{Y}) + W_\infty(\mathcal{Y}, \mathcal{Z})$.

¹⁴Este tipo de distancia es una variante de la distancia de Hausdorff.

Corolario 2.1 *La distancia de cuello-de-botella W_∞ es una métrica.*

Teorema 2.3 (Teorema de Estabilidad para Filtraciones)

Sea K un complejo y $f, g : K \rightarrow \mathbb{R}$ dos funciones monótonas. Si \mathcal{X} e \mathcal{Y} son los diagramas de Persistencia de f y de g , respectivamente, se tiene que para diferentes niveles p de los diagramas

$$W_\infty(\mathcal{X}, \mathcal{Y}) \leq \|f - g\|_\infty.$$

La idea de este resultado consiste en considerar las familias paramétricas de la sección anterior. En concreto, cada una de las funciones f_t proporciona un diagrama de Persistencia. Se debe considerar la variable del tiempo t como una tercera variable que nos sirve para estudiar la transición de los diferentes puntos de un diagrama de un tiempo a otro en $\mathbb{R}^2 \times [0, 1]$.

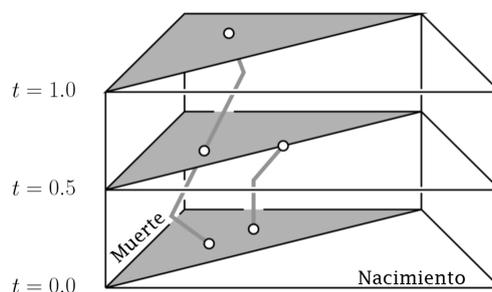


Figura 2.22: La familia uniparamétrica de los diagramas de Persistencia de la línea recta homotópica entre $f_0 = f$ y $f_1 = g$. Un trozo marca una “raíz” que abarca todo el intervalo, mientras que el otro desaparece en la diagonal en la mitad de la homotopía.

Para aplicar el teorema de Estabilidad es conveniente extenderlo a funciones más generales. Sabemos que toda función continua en un espacio topológico cúbico puede ser aproximada por PL-funciones, para cada una de las cuales existe una función monótona que genera los mismos diagramas de Persistencia.

Definición 2.20 *Decimos que $a \in \mathbb{R}$ es un **valor crítico homológico** si existe algún $\epsilon > 0$ para el cual $f_n^{(a-\epsilon, a+\epsilon)}$ es un isomorfismo para cada dimensión n .*

Definición 2.21 *Decimos que f es una **función moderada** (o **de crecimiento moderado**) si f sólo tiene un número finito de valores críticos homológicos y todos los grupos de homología de todos los conjuntos de subniveles tienen rango finito.*

La principal motivación para esta definición es la relativa facilidad con la que podemos definir los diagramas de Persistencia. Con estas definiciones podemos enunciar el siguiente Teorema de Estabilidad.

Teorema 2.4 (Teorema de Estabilidad para funciones moderadas)

Sea K un complejo y $f, g : K \rightarrow \mathbb{R}$ dos funciones de “suavizado” (corrección de irregularidades). Si \mathcal{X} e \mathcal{Y} son los diagramas de Persistencia de f y de g , respectivamente, se tiene que para diferentes niveles p de los diagramas

$$W_\infty(\mathcal{X}, \mathcal{Y}) \leq \|f - g\|_\infty.$$

La idea de la demostración se ilustra en la siguiente figura.

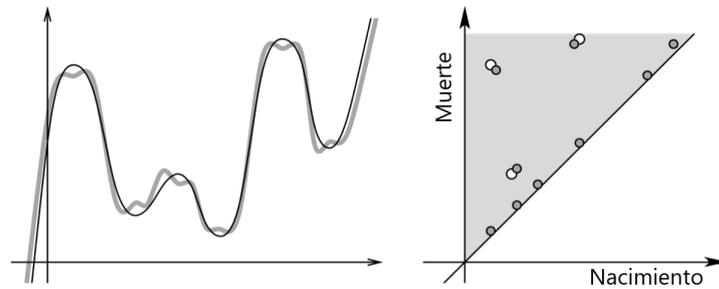


Figura 2.23: Izquierda: Dos funciones con pequeña distancia respecto a la norma infinito. Derecha: Los dos diagramas de persistencia correspondientes con pequeña distancia de cuello-de-botella.

2.3. Sistemas Expertos y fibraciones.

De una forma muy general, un *Sistema Experto* es una versión computacional de un conjunto de reglas lógicas. Existen diferentes tipos de lógica; una primera clasificación distingue entre lógica de clases ($A = A$), lógica proposicional (Si A , entonces B) y lógica descriptiva (Si A entonces B_i , con cierta probabilidad p_i para $1 \leq i \leq r$), según orden creciente de dificultad y, por consiguiente, de implementación computacional.

Los Sistemas Expertos avanzados pueden utilizar varios tipos de lógica de forma simultánea (procedimiento distribuido) o de forma consecutiva (procedimiento secuencial). Para facilitar la toma de decisiones se utilizan árboles de decisión. Un árbol multidimensional se puede interpretar como una representación simbólica de una *fibración con singularidades* que representan las bifurcaciones del árbol. Un ejemplo básico está dado por grafos de Reeb para fibraciones de curvas integrales asociadas al campo gradiente de una función de Morse.

La *toma de una decisión* se toma en función de “pesos” relativos localizados en los elementos (aristas, caras, etc) del grafo dirigido que están situados por encima que confluyen en cada vértice. De forma complementaria, se toman en función de la “esperanza” (integral de una distribución de probabilidad) asociada al subárbol situado por debajo de cada vértice en el grafo dirigido. Los esquemas de tipo bayesiano proporcionan una realimentación entre ambos sistemas de pesos variables.

La toma automática de decisiones requiere *diferentes fases* que afectan a

1. El procesamiento (selección, extracción, mejora) de la información digital (estática o dinámica):
2. El análisis local (agrupamiento de primitivas básicas, p.e.) y la evaluación (cuantitativa o cualitativa) de la información asociada al procesamiento;
3. la interpretación global en función del contexto o de la tarea a realizar;
4. la simulación de los efectos producidos por la decisión seleccionada (sobre una representación simplificada)

El *objetivo fundamental* de esta sección es proporcionar una introducción a estas fases que pueden combinar multitud de procesos diferentes (la ordenación no es unívoca). Para facilitar una mejor comprensión de los modelos subyacentes se ilustran con las correspondientes fases para imágenes digitales como primer paso para la comprensión del contenido en secuencias de vídeo. Para ello, es necesario disponer de un modelo matemático de una imagen digital como soporte y del reconocimiento de su contenido como proceso; la primera cuestión es conocida desde hace más de cincuenta años, mientras que la segunda es aún objeto de investigación.

2.3.1. Un ejemplo ilustrativo: Imágenes Digitales.

Hay muchos tipos de imágenes digitales. En esta sección se supone que una imagen es un mapa de bits (bitmap), es decir, una aplicación discreta definida sobre un rectángulo cuyos puntos son píxeles sobre los cuales se tiene definida una función (escalar o vectorial) discreta de intensidad. Por ejemplo, para una imagen de 8 bits en escala de grises, la función de intensidad I_g en la escala de grises puede tomar $2^8 = 256$ valores que se representan como

los puntos de coordenadas enteras en el intervalo $[0, 255]$. Para una imagen en color RGB en 8 bits se tienen tres canales de intensidad que toman valores en el intervalo $[0, 255]$ cada uno, lo cual da lugar 256^3 colores p.e. (los utilizados en aparatos digitales de TV, p.e.).

La extracción de información contenida en imágenes o vídeos digitales se lleva a cabo mediante una estrategia basada en “hechos detectables” (enfoque *bottom-up*) o bien basada en “modelos” (enfoque *top-down*). Habitualmente, se utiliza una realimentación entre ambos. El reconocimiento automático presenta una complejidad bastante mayor que requiere Sistemas Expertos (ver más abajo).

A continuación se ilustran las fases descritas en la introducción para imágenes digitales¹⁵.

Procesamiento.

Salvo casos triviales de imágenes sintéticas generadas por el ordenador, la distribución de la intensidad es habitualmente muy irregular. Es necesario realizar procesos de suavizado y procesos de realzado en la imagen digital. Los primeros refuerzan las restricciones de continuidad para I_g y, por consiguiente, facilitan el agrupamiento local de píxeles con propiedades similares contenidos en la imagen. Los segundos, refuerzan las discontinuidades de I_g (tras minimizar el ruido) y, por consiguiente, permiten extraer las primitivas básicas lineales (mini-segmentos).

Análisis.

El objetivo del Análisis de imágenes digitales es proporcionar una colección de primitivas topológicas básicas (regiones r_α) y geométricas básicas (bordes o siluetas $s_\alpha = \partial r_\alpha$ de regiones). Los elementos a detectar y enlazar en los bordes b_α son puntos (junturas) y minisegmentos agrupados según condiciones de colinealidad para los PL-modelos subyacentes.

Para ello se utilizan estrategias de agrupamiento y de seguimiento a lo largo de poligonales simples (topológicamente equivalentes a una curva de Jordan, es decir, sin auto-intersecciones). Las características de la forma resultantes se describen en términos de momentos (eventualmente centrales) de orden bajo; por consiguiente, los modelos algebraicos usan curvas de grado bajo (habitualmente ≤ 4).

Interpretación.

La interpretación requiere agrupar regiones adyacentes r_α con características “similares” relativas a “alguna propiedad” de la imagen o de la escena. Esta propiedad se expresa inicialmente en términos de una función continua definida (profundidad, altura, p.e.) ó alguna característica cinemática (energía, p.e.), tanto en su versión determinista como probabilista (usando pdf: probability density function).

El agrupamiento de un número finito de regiones $r_{\alpha,i}$ proporciona objetos planares b_α con características comunes respecto a la función elegida. Dependiendo de la función se pueden obtener diferentes resultados. La *interpretación* requiere distinguir entre los objetos b_α de interés situados en el fondo *BG* (Background) con respecto a los objetos situados en el primer plano *FG* (Foreground). Los objetos contenidos en *BG* proporcionan el contexto.

En esta fase es necesaria la utilización de Sistemas Expertos a bajo nivel para discriminar objetos usando diferentes tipos de restricciones. Existe un gran número de restricciones que se pueden etiquetar como lógicas, topológicas, geométricas, ó analíticas para el caso estático. Todas ellas tienen su correspondiente versión estadística o cinemática o ambas a la vez. Los sistemas Expertos se construyen sobre modelos de razonamiento que facilitan la automatización en procesos de razonamiento a través del aprendizaje (Machine Learning).

Los Sistemas Expertos utilizan diferentes tipos de restricciones (asociadas al alineamiento ó ajuste con respecto a curvas, p.e.), la mejora (suprimiendo outliers, minimizando el ruido, p.e.) y compleción de imágenes (corrigiendo discontinuidades mediante enlazado automático, p.e.). Como resultado se obtienen “primitivas” radiométricas (relativas a regiones con color casi-homogéneo, es decir, módulo un umbral de variabilidad) y geométricas (relativas a bordes de dichas regiones) que sean significativos y computacionalmente manejables en un modelo analítico de la imagen.

¹⁵Una versión detallada con referencias adicionales se puede encontrar en los apuntes de Javier Finat correspondientes al módulo 1 (Procesamiento y Análisis de Imágenes) de CEViC (Curso de Especialista en Visión por Computador)

Simulación.

Para identificar si la información resultante de las fases anteriores es consistente o no con modelos de objetos o de escenas, o bien con características de tareas (multicaminos, acciones, interacciones), es necesario estimar las transformaciones ó las deformaciones que afectan a los objetos b_α , las escenas S ó su evolución conjunta a lo largo del tiempo que denotamos mediante $b_\alpha(t)$ y $S(t)$, respectivamente.

Las tareas o las (inter)acciones se modelan en términos de campos (co)vectoriales o, con más generalidad, tensoriales. La simulación de la propagación se lleva a cabo en términos de derivadas de Lie de los diferentes tipos de campos (escalares, vectoriales, tensoriales). Debido a su mayor complejidad, en este TFG no se aborda esta cuestión.

2.3.2. Una introducción a las Redes Neuronales.

Las redes neuronales son uno de los paradigmas de programación más interesantes jamás inventados. En el enfoque convencional de la programación, es necesario fijar objetivos interpretables por el ordenador, descomponiendo cada problema en muchas tareas pequeñas y definidas con precisión que el equipo puede realizar fácilmente. Por el contrario, en una red neuronal no se llega a un nivel de especificidad tan elevado; en particular, no se especifica cómo resolver el problema. En lugar de ello, se introducen procedimientos de aprendizaje a partir de datos observacionales, que es necesario interpretar para obtener una solución al problema en cuestión.

El objetivo de la red neuronal es resolver los problemas de forma similar a cómo opera el cerebro humano; no se pretende imitar la morfología del cerebro ni su estructura en cinco capas, aunque inicialmente las redes neuronales artificiales (ANN: Artificial Neural Networks) tuvieran asimismo cinco capas (entrada y salida más tres capas intermedias). Para ello, se utilizan modelos simplificados abstractos en las que las ANN tienen un número relativamente bajo de “células” idénticas entre sí, pero con pesos diferentes. Por ello, las ANN son más abstractas que las biológicas donde es posible encontrar más de diez tipos de células diferentes. Las ANN actuales pueden contener desde unos miles a unos pocos millones de unidades neuronales.

Una red neuronal es un modelo computacional basado en el comportamiento del cerebro humano. Consiste en un conjunto finito de neuronas artificiales. El modelo inicial de una neurona artificial está dado por un perceptrón. Las células de capas consecutivas se conectan entre sí mediante “segmentos” (aristas de un grafo) que se activan o no dependiendo del nivel de intensidad de la señal; este comportamiento es análogo al de las neuronas biológicas conectadas entre sí en función de características de las señales recibidas; dicha señal es modificada y transmitida a las neuronas adyacentes. La modificación depende del peso w_{ij} de la j -ésima neurona en la capa i -ésima, así como de los pesos en las neuronas adyacentes. Según el esquema tradicional, la información en la capa de entrada (asociada a sensores) se transmite a las capas sucesivas atravesando la red neuronal a lo largo de caminos que conectan neuronas en diferentes capas, hasta suministrar valores de salida que deben ser comparados con los esperados.

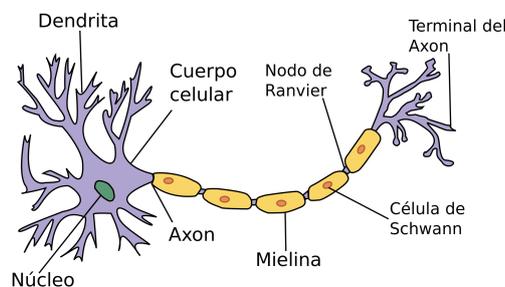


Figura 2.24: Estructura neuronal

Perceptrones.

El *Perceptrón* es un tipo de red neuronal artificial básica que utiliza un modelo de discriminación lineal, es decir, basado en la evaluación con respecto a hiperplanos: se asigna el

valor uno o cero, dependiendo del signo que toma $\underline{w} \cdot \underline{x} - u$, donde u es un umbral y $\underline{w} \cdot \underline{x}$ el producto escalar del vector de pesos \underline{w} por el de valores actuales \underline{x} de las variables a evaluar. Los perceptrones fueron desarrollados en la década de 1950 y 1960 por Frank Rosenblatt, inspirado en los trabajos de Warren McCulloch y Walter Pitts. Hoy en día, es más común utilizar otros modelos de neuronas artificiales con una respuesta no necesariamente binaria (uno de los modelos más utilizados está dado por la *Neurona Sigmoidea*, de las que hablaremos más adelante).

Un Perceptrón tiene varias entradas binarias y produce una sola salida binaria.

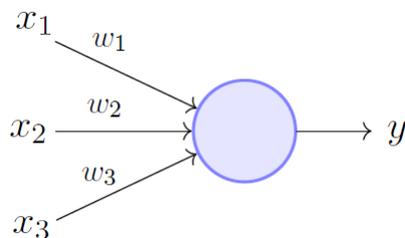
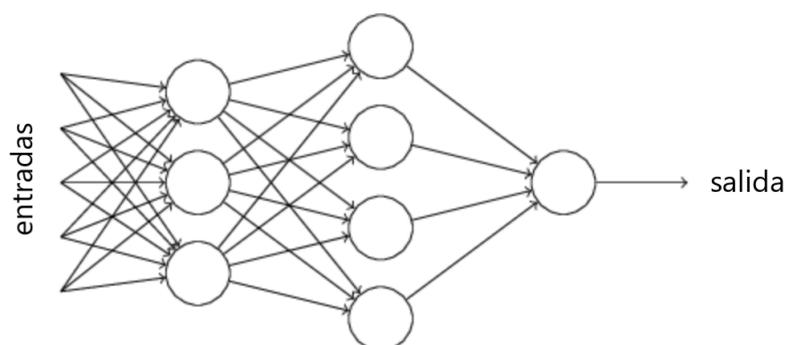


Figura 2.25: Representación Perceptrón

En el ejemplo mostrado, el Perceptrón tiene 3 entradas: x_1, x_2, x_3 . En general podría tener más o menos entradas. Rosenblatt propuso una regla simple para calcular la salida. Introdujo pesos w_1, w_2, w_3 expresando la importancia de las entradas correspondientes a la salida de los números reales. La salida de la neurona y , que puede ser 0 o 1 según que la suma ponderada $\sum_j w_j x_j$ (producto escalar) sea menor o mayor que cierto valor umbral $u = \delta$. Al igual que los pesos, el umbral δ es un número real que es un parámetro de la neurona. En una terminología más matemática, el modelo matemático básico de un Perceptrón es el siguiente:

$$y = \begin{cases} 0 & \text{Si } \sum_j w_j x_j \leq \delta, \\ 1 & \text{Si } \sum_j w_j x_j > \delta. \end{cases}$$

Obviamente el Perceptrón es un modelo muy simplificado para la toma de decisiones humanas, que afecta esencialmente a la lógica de clases. Sin embargo, una combinación distribuida de perceptrones puede sopesar diferentes tipos de evidencia para tomar decisiones. Esto hace parecer plausible que una red compleja de Perceptrones podría tomar decisiones bastante sutiles.



En esta red, la *primera capa de perceptrones* (representado por una columna), está tomando tres decisiones muy simples, junto con los pesos de las entradas. Cada uno de los perceptrones de la *segunda capa* está tomando decisiones en base a las salidas de la primera capa. Mediante este procedimiento recurrente, un Perceptrón en la segunda capa puede tomar decisiones en un nivel más abstracto y más complejo que los perceptrones situados en la primera capa. Decisiones aún más complejas se pueden hacer por el Perceptrón en la *tercera capa*¹⁶. De esta manera, una red de muchas capas de perceptrones puede participar en tomas de decisión más sofisticadas.

¹⁶Las salidas de un perceptrón son todas la misma, solo tienen una salida.

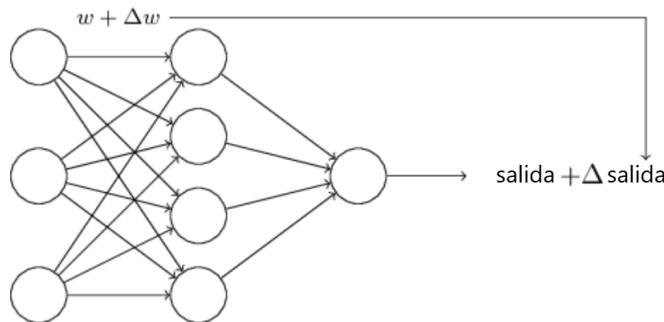
De una forma más sintética, el perceptrón es un modelo en el que las entradas y los pesos pueden ser interpretados como dos vectores cuyo producto escalar es $\underline{w} \cdot \underline{x} = \sum_j w_j x_j$, y el umbral se desplaza al otro lado de la desigualdad y se renombra como *Sesgo del Perceptrón*, $b \equiv -\delta$. El nuevo modelo se reescribe de la siguiente manera:

$$y = \begin{cases} 0 & \text{Si } \underline{w} \cdot \underline{x} + b \leq 0, \\ 1 & \text{Si } \underline{w} \cdot \underline{x} + b > 0. \end{cases}$$

Los Perceptrones se describen como un método que permite “pesar la evidencia” en relación con la toma de decisiones. Otra forma en que se pueden utilizar los perceptrones es para computar las funciones lógicas elementales que solemos pensar como computación subyacente (funciones como **AND**, **OR** y **XOR**).

Neuronas sigmoides.

Supongamos que tenemos una red de perceptrones que nos gustaría usar para aprender a resolver algún problema. Por ejemplo, las entradas a la red podrían ser los datos de píxel sin procesar de una imagen escaneada manuscrita de un dígito. nos gustaría que la red aprendiera *pesos* y *sesgos* para la salida, incluyendo que la salida de la red clasifique correctamente un dígito ó una letra (caracteres alfa-numéricos). Para ver cómo podría funcionar el aprendizaje, supongamos que realizamos un pequeño cambio en algún peso (o sesgo) en la red. Sería deseable que este pequeño cambio de peso cause sólo un pequeño cambio correspondiente en la salida de la red. De este modo, el aprendizaje es no sólo posible, sino también robusto con respecto a “pequeñas perturbaciones”. Esquemáticamente, esto es lo que queremos:



Si fuera cierto que un pequeño cambio en un peso (o sesgo) sólo causa un pequeño cambio en la salida, entonces podríamos usar este hecho para modificar los pesos y sesgos para conseguir que nuestra red se comporte más de la manera que queremos. Por ejemplo, supongamos que la red clasifica erróneamente una imagen como un “8” cuando debe ser un “9”. Podríamos averiguar cómo hacer un pequeño cambio en los pesos y sesgos para que la red se acerque un poco a clasificar la imagen como un “9”. Y luego repetimos esto, cambiando los pesos y los sesgos una y otra vez para producir una mejor y mejor salida. La red estaría aprendiendo.

El problema es que esto no es lo que sucede cuando nuestra red contiene perceptrones. De hecho, un pequeño cambio en las ponderaciones o el sesgo de cualquier Perceptrón en la red a veces puede hacer que la salida de ese Perceptrón se voltee por completo, digamos desde 0 hasta 1. Esa “vuelta” puede hacer que el comportamiento del resto de la red cambie por completo de una manera muy complicada. Así, mientras el “9” podría clasificarse ahora correctamente, el comportamiento de la red en todas las demás imágenes es probable que haya cambiado por completo en alguna manera difícil de controlar. Eso dificulta ver cómo modificar gradualmente los pesos y sesgos para que la red se acerque al comportamiento deseado.

Podemos superar este problema introduciendo un nuevo tipo de neurona artificial llamada *Neurona sigmoidea*. Las neuronas sigmoides son similares a los perceptrones, pero modificadas para que pequeños cambios en sus pesos y sesgos causen sólo un pequeño cambio en su producción. Ese es el hecho crucial que permitirá que una red de “neuronas sigmoides” para aprender.

Para describir las *neuronas sigmoides* usamos la misma representación que el perceptrón en la figura anterior. La diferencia es que las entradas pueden ser también valores entre 0 y 1,

es decir, $x_j \in [0, 1]$. Otra diferencia es que la salida no es 0 o 1 sino $\sigma(\underline{w} \cdot \underline{x} + b)$ donde σ se lo conoce por *Función sigmoide*¹⁷ y está definida de la siguiente manera:

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

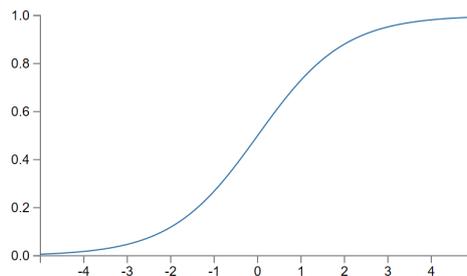
De forma más explícita, la salida de una neurona sigmoide es

$$\frac{1}{1 + e^{-\underline{w} \cdot \underline{x} - b}}.$$

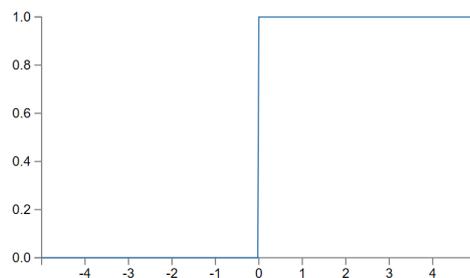
A primera vista, las neuronas sigmoides parecen muy diferentes a los perceptrones. La forma algebraica de la función sigmoide puede parecer opaca y prohibir si no estás ya familiarizado con ella. De hecho, hay muchas semejanzas entre perceptrones y neuronas sigmoides; en particular, la forma algebraica de la función sigmoide resulta ser más un detalle técnico que una verdadera barrera a la comprensión.

Para entender la semejanza con el modelo del Perceptrón, supongamos que $z = \underline{w} \cdot \underline{x}$ es un número positivo grande. Luego $e^{-z} \approx 0$ y por tanto $\sigma(z) \approx 1$. En otras palabras, cuando $z = \underline{w} \cdot \underline{x}$ es grande y positivo, la salida de la neurona sigmoide es aproximadamente 1, tal como lo habría sido para un Perceptrón. Por otra parte, supongamos que $z = \underline{w} \cdot \underline{x}$ es muy negativo. Por tanto $e^{-z} \rightarrow \infty$, luego $\sigma(z) \approx 0$. Así que cuando $z = \underline{w} \cdot \underline{x}$ es muy negativo, el comportamiento de una neurona sigmoidea también se aproxima estrechamente a un Perceptrón. Es solo cuando el valor de z es de tamaño “neutral” que hay mucha desviación del modelo del Perceptrón.

Lo realmente importante de la función sigmoide σ es la forma de la función cuando se traza:



Esta forma es una versión suavizada de una función de salto/paso:



Si σ de hecho había sido una función de salto, entonces la neurona sigmoide sería un Perceptrón, ya que la salida sería 1 o 0 dependiendo de si z fuera positiva o negativa. Mediante el uso de σ se obtiene un Perceptrón “suavizado”. De hecho, es la suavidad de la función σ donde reside su importancia. La suavidad de σ significa que pequeños cambios en los pesos, Δw_j , y en el sesgo, Δb , producirá pequeños cambios en la salida (output) de la neurona, Δsalida . De hecho el cálculo nos dice que Δsalida está bien aproximada por

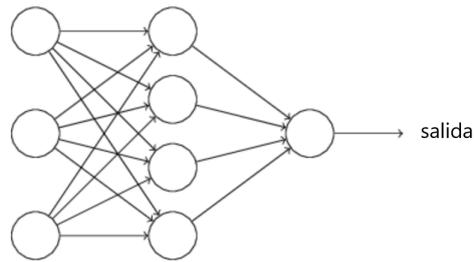
$$\Delta \text{salida} \approx \sum_j \frac{\partial \text{salida}}{\partial w_j} \Delta w_j + \frac{\partial \text{salida}}{\partial b} \Delta b,$$

donde Δsalida es una función lineal de los cambios Δw_j y Δb . Esta linealidad hace que sea fácil elegir pequeños cambios en los pesos y sesgos para lograr cualquier cambio pequeño deseado en la salida. Así, aunque las neuronas sigmoides tienen en buena parte un comportamiento cualitativo similar al de los perceptrones, su carácter suave hace mucho más fácil averiguar cómo cambiar los pesos y sesgos para modificar la salida de una forma gradual.

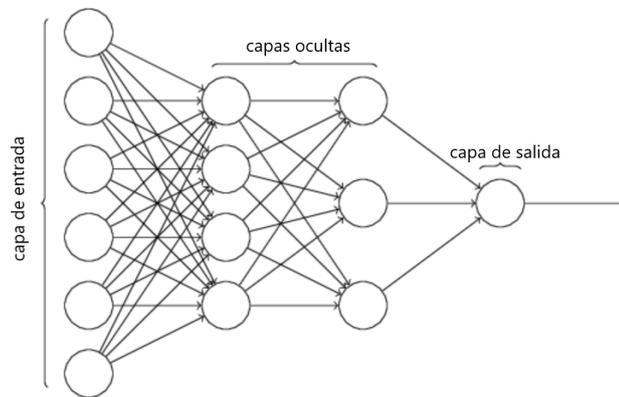
¹⁷A veces a esta función se llama *función logística*, y esta nueva clase de neuronas llamadas *neuronas logísticas*.

Arquitectura de la Redes Neuronales.

Supongamos que tenemos la siguiente red:



Como se mencionó anteriormente, la capa más a la izquierda en esta red se llama la *capa de entrada*, y las neuronas dentro de la capa se denominan *neuronas de entrada*. La capa más a la derecha o de salida contiene las neuronas de salida, o, como en este caso, una sola neurona de salida. La capa media se llama *capa oculta*, ya que las neuronas de esta capa no son ni entradas ni salidas. El término “oculto” realmente significa nada más que “no es una entrada o una salida”. La red anterior tiene una sola capa oculta, pero algunas redes tienen varias capas ocultas. Por ejemplo, la siguiente red de cuatro niveles tiene dos capas ocultas:



Por razones históricas y a pesar de alguna confusión asociada, las redes de múltiples capas a veces se denominan perceptrones multicapa o MLPs (*multilayer perceptrones*), aunque puedan estar formadas por neuronas sigmoides, no perceptrones.

Mientras que el diseño de las capas de entrada y salida de una red neuronal es a menudo sencillo, se requiere una habilidad especial para el diseño de las capas ocultas. En particular, no es posible resumir el proceso de diseño de las capas ocultas con unas pocas reglas simples. Para avanzar en ese terreno, los investigadores de las redes neuronales han desarrollado muchas heurísticas de diseño para las capas ocultas, que ayudan a los desarrolladores a obtener el comportamiento que quieren de sus redes. Por ejemplo, esta heurística se puede utilizar para ayudar a determinar cómo negociar el número de capas ocultas en el tiempo necesario para entrenar la red.

Hasta ahora, hemos estado discutiendo redes neuronales donde la salida de una capa se usa como entrada para la siguiente capa. Estas redes se denominan *redes neuronales de alimentación directa*. Esto significa que no hay bucles en la red (la información siempre se alimenta hacia adelante, nunca se vuelve a alimentar). Si tuviéramos bucles, terminaríamos con situaciones en las que la entrada a la σ función dependía de la salida.

Sin embargo, hay otros modelos de redes neuronales artificiales en los que los bucles de retroalimentación son posibles. Estos modelos se denominan *redes neuronales recurrentes*. La idea en estos modelos es tener neuronas que “disparan” (se activan) durante un tiempo limitado, antes de volverse quiescente ó temporalmente inactivo. Ese disparo puede estimular otras neuronas, que pueden disparar un poco más tarde, también por una duración limitada. Eso provoca que aún más neuronas se disparen, y así con el tiempo se obtiene una cascada de neuronas “disparando”. Los bucles no causan problemas en un modelo de este tipo, ya que la salida de una neurona sólo afecta a su entrada en algún momento posterior, no instantáneamente.

Las redes neuronales recurrentes han sido menos influyentes que las redes de avance, en parte porque los algoritmos de aprendizaje para redes recurrentes son (al menos hasta la fecha) menos poderosos. Pero las redes recurrentes siguen siendo extremadamente interesantes. Están mucho más cerca en espíritu de cómo funcionan nuestros cerebros que las redes de feedforward. Y es posible que las redes recurrentes puedan resolver problemas importantes que sólo pueden resolverse con gran dificultad por las redes de alimentación directa.

Aprendizaje con descenso degradado.

Denotaremos por x a una entrada de entrenamiento y por $y = y(x)$ la correspondiente salida. Lo que nos gustaría es un algoritmo que nos permite encontrar pesos y sesgos para que la salida de la red se aproxima a $y(x)$ para todas las entradas de entrenamiento x . Para cuantificar si nos acercamos ó no a este objetivo definimos una *función de coste*¹⁸:

$$C(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2.$$

Aquí, w denota la colección de todos los pesos en la red, b todos los sesgos, n es el número total de entradas de entrenamiento, a es el vector de salidas (depende de w , x y b) de la red cuando x es la entrada, y la suma es sobre todas las entradas de entrenamiento, x . Llamaremos a C la función de coste cuadrático; también se conoce a veces como *error cuadrático medio* o simplemente MSE (*Mean Squared Error*).

Inspeccionando la forma de la función de coste cuadrático, vemos que $C(w, b)$ no es negativo, ya que cada término en la suma no es negativo. Además, el coste $C(w, b)$ se vuelve pequeño, es decir, $C(w, b) \approx 0$, precisamente cuando $y(x)$ es aproximadamente igual a la salida, a , para todas las entradas de entrenamiento x . Por tanto un algoritmo será “eficiente” si puede encontrar pesos y sesgos para que $C(w, b) \approx 0$. El caso contrario significaría que $y(x)$ no está cerca de la salida. El objetivo está en minimizar el costo, es decir, encontrar un conjunto de pesos y sesgos que hacen que el coste sea lo más pequeño posible. Esto se consigue usando un algoritmo conocido como *descenso degradado*.

Supongamos que queremos minimizar una función $C(\underline{v})$, donde $\underline{v} = v_1, v_2, \dots$ es un vector real¹⁹. Pensemos por ejemplo que $\underline{v} = (v_1, v_2) \in \mathbb{R}^2$.

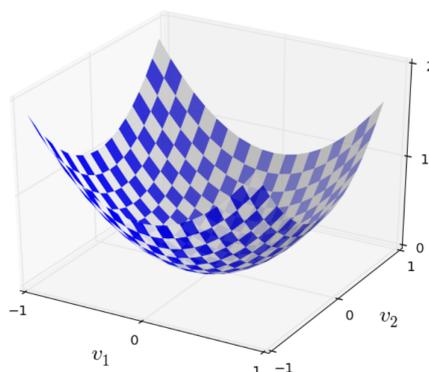


Figura 2.26: Ejemplo de grafo de una función $C(\underline{v})$.

Lo que queremos es encontrar donde alcanza su mínimo global. Podríamos aplicar el cálculo de las derivadas de la función para encontrar los diferentes sitios donde se puede encontrar el extremo, pero esto en las redes neuronales es muy costoso pues las dimensiones son muy grandes. Pero existe una analogía que sugiere un algoritmo que funciona bastante bien. La idea es la siguiente: Imaginemos que el grafo de la función es como un “valle” y colocamos una pelota en una de sus laderas, entonces la pelota descenderá por la pendiente y quedará atraída por el punto más bajo del valle. Por tanto, comenzamos eligiendo un punto de partida al azar (donde colocamos esa pelota imaginaria) y luego simulamos la caída de ese punto por la pendiente hasta el punto más bajo. A través del cálculo de las derivadas de $C(\underline{v})$ podemos conocer la forma local del grafo, y por tanto saber también como cae nuestra pelota.

¹⁸A veces se la conoce como *pérdida* que está relacionado con la diferencia con respecto a la *función objetivo*

¹⁹Nótese que hemos reemplazado w y b por \underline{v} .

De una forma más precisa, cuando movemos el punto de partida en una pequeña cantidad Δv_1 en la dirección de v_1 y una pequeña parte Δv_2 en la dirección de v_2 . El cálculo nos dice que la función C cambia de la siguiente manera:

$$\Delta C \approx \frac{\partial C}{\partial v_1} \Delta v_1 + \frac{\partial C}{\partial v_2} \Delta v_2.$$

Veamos como encontrar una forma de elegir Δv_1 y Δv_2 con el fin de hacer ΔC negativo (así la pelota quedará en el fondo del valle). Para ello denotamos $\Delta \underline{v} = (\Delta v_1, \Delta v_2)^T$, donde T es la operación “trasposición”. También denotamos por $\nabla C = (\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2})^T$ al vector gradiente de C . Con esta notación tenemos que

$$\Delta C \approx \nabla C \cdot \Delta \underline{v}.$$

Supongamos que elegimos $\Delta \underline{v} = -\eta \nabla C$, donde η es un pequeño parámetro positivo conocido como *tasa de aprendizaje*. Entonces la ecuación anterior nos dice que

$$\Delta C \approx -\eta \nabla C \cdot \nabla C = -\eta \|\nabla C\|^2.$$

Como $\|\nabla C\|^2 \geq 0$, esto garantiza que $\Delta C \leq 0$, es decir, C siempre disminuirá, nunca aumentará si cambiamos \underline{v} según la expresión anterior. Esta es la propiedad que buscábamos. Tomaremos la ecuación $\Delta \underline{v} = -\eta \nabla C$ para calcular un valor para $\Delta \underline{v}$, luego moveremos la posición del punto de partida por esa cantidad:

$$\underline{v} \rightarrow \underline{v}' = \underline{v} - \eta \nabla C.$$

A continuación, usaremos de nuevo esta regla de actualización para realizar otro movimiento. Si seguimos haciendo esto, una y otra vez, vamos a seguir disminuyendo C hasta llegar al mínimo global.

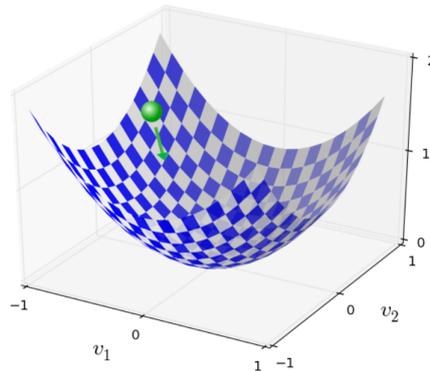


Figura 2.27: Representación gráfica del algoritmo.

Todo esto que acabamos de explicar funciona igual de bien para $\underline{v} = (v_1, \dots, v_n) \in \mathbb{R}^n$.

Para hacer que el descenso de degradado funcione correctamente, tenemos que elegir la tasa de aprendizaje η lo suficientemente pequeña como para que la ecuación

$$\Delta C \approx \nabla C \cdot \Delta \underline{v}.$$

sea una buena aproximación. Si no lo hacemos, podríamos terminar con $\Delta C > 0$, que obviamente no es lo que queremos. Al mismo tiempo, no queremos que η sea demasiado pequeño, ya que eso hará que los cambios $\Delta \underline{v}$ sean pequeños, y por lo tanto el algoritmo de descenso degradado funcionará lentamente.

Algoritmo Backpropagation.

Anteriormente hemos visto cómo las redes neuronales pueden aprender sus sesgos y pesos usando el algoritmo de descenso degradado. Sin embargo no discutimos cómo calcular el gradiente de la función de costo. En esta parte explicaremos un algoritmo rápido para computar tales gradientes, un algoritmo conocido como *backpropagation*. Este algoritmo nos da información detallada sobre cómo cambiar las ponderaciones y los sesgos que implican

cambios en el comportamiento general de la red. Backpropagation se introdujo originalmente en la década de 1970, pero su importancia no fue completamente apreciada hasta 1986 por David Rumelhart, Geoffrey Hinton y Ronald Williams. Hoy en día backpropagation es el algoritmo usado habitualmente para el aprendizaje en las redes neuronales.

Vamos a escribir el algoritmo a continuación:

1. **Entrada (x):** Se establece la activación correspondiente a^1 para la capa de entrada.
2. **Feedforward:** Para cada $l = 2, 3, \dots, L$ computar $z^l = w^l a^{l-1} + b^l$ y $a^l = \sigma(z^l)$.
3. **Error de salida δ^L :** Computar el vector $\delta^L = \nabla_a C \odot \sigma'(z^L)$.
4. **Retroalimentar el error:** Para cada $l = L-1, L-2, \dots, 2$ computar $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$.
5. **Salida:** El gradiente de la función coste está dado por

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$$

y

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l.$$

Examinando el algoritmo se puede ver por qué se llama backpropagation: Calculamos los vectores de error δ^l hacia atrás, comenzando por la capa final. Puede parecer peculiar que estemos atravesando la red hacia atrás. Pero si pensamos en la prueba de retropropagación, el movimiento hacia atrás es una consecuencia del hecho de que el coste es una función de las salidas de la red. Para entender cómo el coste varía con pesos y sesgos anteriores, necesitamos aplicar repetidamente la regla de cadena, trabajando hacia atrás a través de las capas para obtener expresiones utilizables. El algoritmo de retro-propagación se basa en operaciones algebraicas lineales comunes (como la adición de vectores, multiplicación de un vector por una matriz, ...). Pero una de las operaciones es poco frecuente y es conocido como *Producto de Hadamard* o *Producto de Schur* que denotamos mediante \odot , aunque en realidad corresponde a \otimes en el lenguaje algebraico del Álgebra Tensorial.

Se define de la siguiente manera: Sean las matrices

$$A_{m,n} \equiv (a_{ij}); B_{m,n} \equiv (b_{ij}); P_{m,n} \equiv (p_{ij}).$$

Se dice que la matriz P es *el producto de Hadamard, Schur ó Kronecker de las matrices A y B* (diferentes formas de llamar al producto tensorial), y se denota $A \odot B$ ó $A \otimes B$, si

$$P_{m,n} = A_{m,n} \odot B_{m,n} \Rightarrow p_{ij} = a_{ij} b_{ij} \quad \forall i = 1, 2, \dots, m; \quad \forall j = 1, 2, \dots, n.$$

2.3.3. Algunas extensiones de las ANN.

Para remediar los problemas mencionados en relación con el diseño inicial de las ANN, a lo largo de los años ochenta se introducen estructuras adicionales que aceleran de forma sustancial procesos de aprendizaje y entrenamiento. Los desarrollos más relevantes permiten describir:

- Cambios cualitativos aleatorios o “mutaciones” (GAs: Algoritmos Genéticos);
- Procesos evolutivos controlados de acuerdo con funciones de coste (EP: Evolutionary Programming); o, bien
- Distribuciones de probabilidad convergentes sobre representaciones planares obtenidas como “proyección no-lineal” (submersiones) de colecciones de multivectores localmente parametrizables (SOM: Self-Organizing Maps).

El enfoque basado en análisis de datos procedentes de múltiples fuentes utiliza estrategias supervisadas o, alternativamente, no-supervisadas. En los dos apartados siguientes se presentan una breve descripción de ambas estrategias para el Reconocimiento Estadístico de Patrones. Los títulos de los diferentes items son los que aparecen en la sección 1.3 de [12], aunque los comentarios son del autor.

Clasificación supervisada.

A continuación se presentan los métodos más relevantes para clasificación *supervisada* junto con una *interpretación geométrica*:

- Análisis de los *K-Vecinos más próximos* etiquetado como KNN (K-Nearest Neighborhoods) basado en una discretización del espacio de vectores de referencia m_i y la selección de aquellos que se presentan con una mayor frecuencia. Utiliza una realimentación entre métodos Bayesianos y métricas sobre el espacio de muestras. En la práctica, soporta métodos de optimización que combinan máxima verosimilitud para valores esperados (MLE: Maximum Likelihood Expectation) y sumas ponderadas de distancias L^1 (para suprimir outliers) y L^2 (para acelerar la convergencia). El enfoque probabilista de la Geometría Riemanniana proporciona el contexto geométrico más próximo a este enfoque.
- *Clasificación por ángulos* correspondientes a las “tendencias” o direcciones predominantes de acuerdo con análisis de vectores propios asociado a SVD (Singular Value Decomposition). Especialmente indicado en presencia de varias métricas locales con diferentes comportamientos. En la práctica, compara los cosenos de las direcciones seleccionando aquellos que maximizan la proyección (minimizando el ángulo) en espacios que representan posibles deformaciones espaciales o evolución temporal asociada a tareas (ajuste de distribuciones irregulares a superficies, p.e.).
- *Funciones de importancia* que utilizan información significativa ó conocimiento a priori para orientar la búsqueda en torno al soporte de dichas funciones. Los procedimientos de muestreo ImpSaC (Importante Sampling Consensus) son una variante de RanSaC (Random Sampling Consensus) que incluye dicha información y que se puede entender como una convergencia hacia las “variedades” representadas por el lugar de ceros de dichas funciones. Dicha convergencia se evalúa en términos de las normales a las variedades según métodos de descenso más rápido tipo gradiente.
- *Análisis Discriminante* basado en la introducción de funciones que “separan clases” de acuerdo con criterios métricos (minimización de distancias para múltiples centros ó sitios), algebraicos (basados en la detección del lugar discriminante asociado a raíces múltiples de polinomios o, con más generalidad, el lugar singular de aplicaciones) o probabilísticos (según criterios de tipo Bayesiano y sus extensiones, p.e.).
- *Clasificación paramétrica* donde los parámetros se detectan en términos de una mayor “concentración” de datos en torno a valores “medios locales” de acuerdo con distribuciones con “buenas propiedades” matemáticas (Poisson en el caso discreto, normales en el continuo). Estos métodos se extiende de forma inmediata a un análisis basado en matrices de covarianza que se interpretan geoméricamente en términos de distribuciones de curvatura; de una forma intuitiva, los autovalores de la matriz de covarianza son estimadores de los autovalores de la matriz de curvatura.

Clasificación no-supervisada.

A continuación se presentan los métodos más relevantes para clasificación *no-supervisada* junto con una *interpretación simbólica*:

- *Agrupamiento simple* basado en la verificación exacta o bien aproximada de un predicado, una función o bien un funcional de tipo determinista o probabilista. Este método se corresponde con la utilización de diferentes tipos de lógica (clases, proposicional, descriptiva) y es compatible con cualquier tipo de representación geométrica o topológica (cuando se “relaja” la exactitud por una “banda de fluctuación”).
- *Agrupamiento jerárquico* basado en sistemas de reglas (taxonomías). Las clases resultantes expresan una jerarquía que se puede visualizar en términos de algún tipo de estratificación para variedades. Las clases se representan como nodos de grafos que interesa convertir en árboles mediante supresión de aristas (conservando la conexividad del grafo). Los algoritmos tipo MST (Minimum Spanning Tree) proporcionan una herramienta de uso común en el tratamiento computacional de la información.

- *Métodos de descomposición* asociados a la localización espacio-temporal de clases y la introducción de una métrica para representar la “atracción” con respecto a cada valor típico promedio. Los diagramas de Voronoi proporcionan el prototipo 2-dimensional que descompone el espacio muestral con los “sitios” como atractores, las aristas como puntos de silla y los vértices como repulsores.
- *Aplicación de Sammon* para representar las relaciones geométricas entre las muestras mediante un grafo 2D que permite ajustar de manera simultánea las tasas de variación de primer y segundo orden; las primeras están representadas por derivadas direccionales (a lo largo de direcciones con variación maximal), mientras que las segundas proporcionan información sobre la variación en la “curvatura” para acelerar la convergencia. La ratio entre ambas tasas de variación permite formalizar la variabilidad espacio-temporal en los procedimientos de clasificación.

Métodos basados en subespacios.

La topología de multi-caminos es una extensión natural en Topología Algebraica para diferentes caminos o tareas que pueden tener de forma simultánea sobre espacios topológicos. En Geometría y Topología Diferencial el estudio de distribuciones de campos o de sistemas de formas diferenciales es la extensión natural del estudio de campos o de 1-formas aisladas. La extensión es no-trivial pues, un campo o una 1-forma siempre es integrable (al menos localmente), mientras que una distribución de campos o un sistema de formas diferenciales rara vez es integrable. Debe verificar el Teorema de Frobenius que define un cerrado de elevada codimensión en los espacios correspondientes.

De forma análoga, en Sistemas Expertos interesa detectar subespacios vectoriales que permitan identificar el comportamiento agrupado de “cantidades” (geométricas o analíticas) en el espacio-tiempo muestral. Para ello, existen diferentes estrategias donde las clases se pueden representar como subespacios variables con sus correspondientes herramientas de estimación basadas en Análisis de Componentes Principales (PCA: Principal Component Analysis) como variante probabilista del método SVD (Singular Value Decomposition) ó bien en propiedades de “proyectores” sobre subespacios. Ambos métodos son muy antiguos y se utilizaban ya en los años treinta.

La novedad de cara a cuestiones de optimización sobre ANN consiste en interpretar los subespacios k -dimensionales L^k como elementos de una Grassmanniana $Gras(k, N)$ donde $N \gg 0$ es la dimensión del espacio muestral o bien como el conúcleo de la proyección sobre el subespacio de “mayor significación” (por su capacidad discriminatoria). Este enfoque permite incorporar de forma natural las condiciones de equilibrio (como ceros de sistemas dinámicos lineales sobre la Grassmanniana) y la evolución estable incluyendo convergencia hacia puntos críticos o ciclos límite en términos de las ecuaciones diferenciales tipo Ricatti sobre espacios de matrices como ecuaciones estructurales sobre las Grassmannianas.

Redes Neuronales para Reconocimiento.

El *Reconocimiento en Visión Computacional* utiliza diferentes tipos de Sistemas Expertos para extraer la información. Según la aproximación bottom-up, el reconocimiento de objetos móviles utiliza habitualmente descriptores, detectores y clasificadores como elementos de un modelo estructural. La inmensa mayoría de las aproximaciones a este tópico utiliza una aproximación muy casuística a cada uno de los 3 objetos.

La introducción de un marco general en el contexto de fibrados vectoriales (ver la segunda sección del capítulo siguiente) para descriptores, detectores y clasificadores, proporciona un nexo común que permite transvasar información entre diferentes aproximaciones.

El enfoque tradicional de las ANN presenta múltiples problemas relacionados con el elevado número de muestras necesario para entrenar una ANN (a veces de varios miles), el elevado coste del entrenamiento (tanto humano como computacional), la lentitud en la convergencia (el aprendizaje se realiza off-line y de forma supervisada), la falta de robustez en los modelos (cualquier pequeña perturbación en los datos o en las características del modelo convierte en inútil los procesos anteriores).

Para remediar estos problemas, desde finales de los años ochenta se han desarrollado diferentes extensiones de las Redes Neuronales Artificiales (ANN). En la reformulación presentada en este trabajo, todos ellos se reinterpretan en términos de estructuras superpuestas a mallas

cuboidales. Localmente se pueden interpretar como una versión discreta local de fibrados vectoriales. Los nodos representan “neuronas” que pueden activarse o no en función de la intensidad de las señales recibidas y transformadas por capas anteriores.

Las funciones de activación pueden presentar diferentes patrones de activación-inhibición; es frecuente utilizar patrones de tipo sigmoideal o más recientemente ondículas (wavelets). En el diseño original la inicialización se lleva a cabo de forma aleatoria; en fases más avanzadas, se puede introducir “funciones de importancia” de acuerdo con las tareas a aprender y, posteriormente, realizar. Algunos de los mecanismos de aprendizaje más frecuentes siguen patrones de aprendizaje por reforzamiento, pudiendo incluir o no precisión métrica.

En el contexto clásico, si $S(x) := (1 + e^{-x})^{-1}$ es la función sigmoideal, el modelo más simple para la realimentación entre capas (layer) sucesivas $\ell - 1$ y ℓ (actual) está dado por

$$o_i^\ell := S\left(\sum_k w_{ki}^{\ell-1} o_k^{\ell-1}\right) + b_i^\ell$$

donde o_i^ℓ denota el output de la i -ésima célula en la capa ℓ , $w_{ki}^{\ell-1}$ representa el peso en las células de la capa precedente (máscara del núcleo) y b_i^ℓ el sesgo (bias). En particular, como la función de salida es una sigmoide asociada a los valores de un hiperplano afín variable (sometido a incertidumbre), debemos desarrollar una aproximación que garantice la “regularidad” de la proyección entre capas sucesivas. En el contexto de la Topología Diferencial, las immersiones y las submersiones son los ejemplos naturales para garantizar dicha regularidad. Curiosamente, sólo las submersiones han sido utilizadas hasta ahora en los SOM de Kohonen que se abordan en la subsección siguiente.

2.3.4. Aplicaciones Auto-Organizantes de Kohonen.

En esta subsección se reformulan en lenguaje geométrico algunos de los elementos básicos de las Aplicaciones Auto-Organizantes (Self-Organizing Maps) de Kohonen [12].

Aunque el enfoque basado en SOM sea el menos desarrollado en la literatura (debido a una mayor complejidad aparente), esta propuesta es la más consistente y robusta desde el punto de vista estructural. Asimismo, presenta aplicaciones espectaculares a lo largo de los noventa como los sistemas más avanzados de reconocimiento de voz y habla.

Un reto importante es la extensión de dicho enfoque al Reconocimiento de contenidos sobre señales 2D (la señal acústica es 1D) y más allá correspondientes a video convencional para señales $2D + 1d$, representaciones volumétricas para señales 3D, vídeo 3D para señales $3D + 1d$. El reconocimiento automático de estas extensiones es un área incipiente de investigación.

Actualmente, se están desarrollando los primeros modelos para reconocimiento automático de contenidos en señales $2D + 1d$ (video digital) basadas en seguimiento simultáneo de múltiples objetos utilizando RNN (Recurrent Neural Networks), que son una extensión natural de las CNN (Convolutional Neural Networks) basadas en productos tensoriales convencionales. Debido a su mayor complejidad, en este TFG no se abordan las RNN salvo en aspectos teóricos que se describen en términos de Flujos Tensoriales (ver última sección del capítulo 3). Por ello, centramos la atención en señales estáticas y sus posibles deformaciones (enfoque casi-estático) donde disponemos de modelos robustos ya probados.

El desarrollo de modelos para reconocimiento de contenido en señales 2D (sobre imágenes digitales) se lleva a cabo inicialmente mediante entrenamiento supervisado de un número “suficiente” de muestras etiquetadas²⁰.

El sistema experto debe ser capaz de reconocer (mediante barrido) la zona etiquetada y extraer las características a detectar de esa zona etiquetada. Una vez validada la primera fase del aprendizaje, se introducen imágenes no-etiquetadas, algunas de las cuales no contienen los elementos buscados y se evalúan tanto los falsos positivos (detección incorrecta de hechos buscados que no están presentes) como los falsos negativos (ausencia de detección en presencia de hechos).

Un modelo básico.

Inicialmente y según T.Kohonen (capítulo 3 de [12]) una aplicación auto-organizante es una “proyección no-lineal” de un array de “vectores de referencia” $\underline{m}_i = [\mu_{i1}, \dots, \mu_{in}] \in \mathbb{R}^n$

²⁰Actualmente, es posible gestionar poblaciones de varios millones de datos con entrenamiento basado en decenas de miles de muestras etiquetadas

(representando pesos escalares variables) sobre elementos de una red discreta 2D correspondientes al soporte de una red neuronal planar que representamos mediante un retículo regular Λ .

Para que este enfoque sea compatible con aprendizaje bajo condiciones de información incompleta o de incertidumbre, se utiliza un vector \underline{x} de datos aleatorios con función de densidad de probabilidad (pdf) dada por $f(\underline{x})$. La proyección no-lineal debe ser compatible con la noción de “topológicamente próxima” con respecto a una distancia topológica. En términos matemáticos, la aplicación proyección debe ser “propia”.

Realimentación y control.

El carácter auto-organizante procede de minimizar *cualquier* tipo de distancia²¹ con respecto al resultado esperado (salida de la red). Este enfoque es compatible con la utilización de diferentes tipos de distancias que induce “cambios” en el vector de pesos usando un efecto de suavizado (formalmente, difeomorfismos) o de relajación local (PL-aproximación a un difeomorfismo). La actualización de la información para mejorar el aprendizaje (de una forma o de un proceso) se lleva a cabo inicialmente en términos de

$$\underline{m}_i(t+1) = \underline{m}_i(t) + \underline{h}_{ci}[\underline{x}(t) - \underline{m}_i(t)]$$

donde r representa una discretización del tiempo y \underline{h}_{ci} es una “función de entorno” para generar un suavizado sobre el retículo.

Alternativamente, el aprendizaje se puede definir en términos de un producto escalar maximizando la proyección con respecto a las direcciones “óptimas” para la estimación de una forma o bien la ejecución de una tarea (descrita en términos de proximidad para las funciones coeficientes de campos).

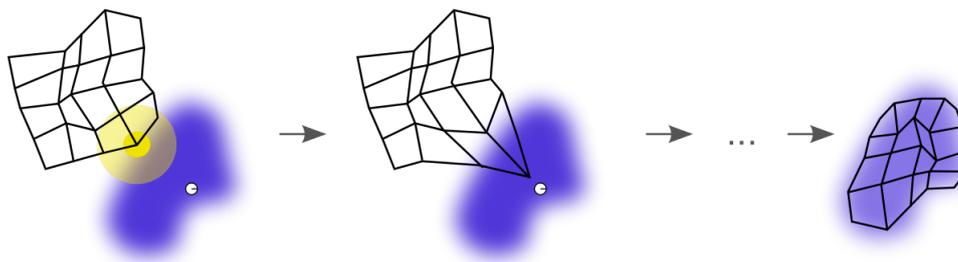


Figura 2.28: Una ilustración de la formación de un mapa de auto-organización. La gota azul es la distribución de los datos del entrenamiento, y el pequeño disco blanco es el dato actual de formación de esa distribución. Al principio los nodos (izquierda) del SOM se colocan arbitrariamente en el espacio de datos. Se selecciona el nodo (resaltado en amarillo) que es la más cercana de los datos de referencia de formación. Se mueve hacia el dato de la formación, como (en menor medida) son sus vecinos en la red. Después de varias iteraciones la red tiende a aproximar la distribución de datos (derecha).

Una reformulación en el marco de la Topología Diferencial.

En el contexto de la Topología Diferencial, el espacio de “estados” asociado a los “vectores de referencia” se reemplaza por una variedad M cuyas coordenadas locales (x_1, \dots, x_m) están dadas por variables aleatorias X_1, \dots, X_m con función de distribución de probabilidad $p(\underline{X})d\underline{X}$, siendo $d\underline{X}$ la medida de Lebesgue sobre \mathbb{R}^m . De la misma forma que en Geometría Diferencial de Variedades, suponemos que las variables aleatorias son funcionalmente independientes en cada abierto U de trivialización; el carácter independiente de dichas variables permite escribir $p(\underline{X})$ como $p(X_1) \dots p(X_m)$.

la “proyección no-lineal” de los datos se reemplaza por una *submersión sobre el retículo* Λ en \mathbb{R}^2 . Los avances recientes en arquitectura distribuida para ANN permiten gestionar no sólo retículos planares sino volumétricos Λ . Para simplificar supondremos que todos los retículos son de tipo cuboidal, pudiendo presentar diferentes tipos de caminos asociados a diferentes

²¹Esta descripción sugiere que el espacio soporte es un espacio de Banach (donde las métricas son equivalentes), hipótesis que no aparece de forma explícita en los trabajos de T.Kohonen

“hilos” (multithread architecture) cuya versión topológica ha sido descrita en el **Capítulo 1**.

La actualización de la información descrita más arriba corresponde a una versión afín de la derivada direccional con respecto a la dirección asociada a h_{ci} . Obviamente, esta versión se puede reinterpretar geoméricamente en términos de una “conexión afín” entendida como derivada covariante de un campo vectorial ó con más generalidad de un tensor (ver sección siguiente). Esta reformulación geométrica permite abordar situaciones mucho más generales como las correspondientes a derivadas direccionales de tensores de uso común en Geometría Diferencial de Variedades.

Asimismo, la “función de entorno” se puede reinterpretar como una función “radial” correspondiente a un entorno tubular (retracción del fibrado normal punteado) encargado del ajuste a la forma deseada o al proceso de aprendizaje de una tarea (entendida como un campo). La “función de entorno” permite evaluar la “bondad del ajuste”, pudiendo interpretarse la proyección con la función de entorno como una “submersión controlada” en el sentido de R.Thom [26].

Con estas adaptaciones terminológicas, reformulamos la noción de aplicación auto-organizante en los términos siguientes:

Definición 2.22 Una *aplicación auto-organizante* es una submersión controlada $M \rightarrow \Lambda^d$ del espacio de estados M de un sistema sobre un retículo d -dimensional Λ con $d = 2$ ó $d = 3$ cuya función de control es genéricamente regular (rango máximo).

Extensión al PAC.

De una forma abstracta, el Ciclo Percepción Acción (PAC) se puede formalizar como una colección de fibraciones sucesivas soportadas por capas sucesivas de una ANN. Los SOM son submersiones controladas entre capas sucesivas. Por ello, desde el punto de vista matemático, proporcionan los elementos básicos para controlar el rendimiento de los Sistemas Expertos basados en funciones.

El reto consiste en extender esta idea tan simple a campos (co)vectoriales o, con más generalidad, tensoriales que se puedan definir sobre el soporte dado por la red neuronal subyacente, entendido como una colección de fibraciones sucesivas. Para poder abordar este problema, es necesaria una reformulación de aspectos funcionales del cerebro en términos tensoriales.

Capítulo 3

Un enfoque computacional en Reconocimiento

El *objetivo del reconocimiento automático* (*machine recognition*) es la detección, análisis y clasificación de regularidades relacionadas con *objetos y relaciones* entre dichos objetos. Tanto unos como otros, pueden ser locales o globales; se pueden describir de múltiples formas (lógicas, funcionales, estadísticas, icónicas, p.e.). Por ello, utilizan métodos y herramientas procedentes de todas las áreas de Matemáticas.

El tratamiento computacional de la información utiliza asimismo diferentes *estructuras de datos* (arrays, listas, tablas, p.e.) sobre los cuales es necesario diseñar e implementar *algoritmos* (inserción y borrado, p.e.) y *procedimientos* (clasificación, p.e.). La *gestión simbólica* se lleva a cabo mediante diferentes tipos de grafos (binarios, k-dimensionales, árboles más generales, p.e.) que facilitan los procedimientos de búsqueda, así como conexiones estructurales con métodos matemáticos para la gestión de objetos y de escenas.

Para articular ambos enfoques (matemático vs computacional) es necesario utilizar herramientas de Inteligencia Artificial (AI). En una primera aproximación, los *Sistemas Expertos* son conjuntos de reglas lógicas que facilitan la capacidad de deducción automática a partir de datos o de predicados sobre dichos datos.

La *evaluación de objetos* se lleva a cabo inicialmente en términos de *operadores* lógicos, funcionales ó estadísticos (para incluir condiciones de incertidumbre); cada operador proporciona una partición en semiespacios y se les etiqueta como *detectores*.

- Los *operadores lógicos* utilizan diferentes tipos de lógica tales como la lógica de clases (resumible en la aparentemente trivial $A = A$, p.e.), proposicional (si $A \Rightarrow B$, p.e.) ó descriptiva (Si A , entonces B con una probabilidad del 60 % ó C con una probabilidad del 40 %, p.e.). Los más difíciles de implementar son los correspondientes a modelos de lógica descriptiva.
- Los *operadores funcionales* más simples devuelven un valor numérico $f(x) \leq 0$ vs $f(x) \geq 0$ con respecto al valor esperado $f(x) = 0$ donde f es una función (a valores en $\mathbb{Z}_2, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$, p.e.) no siempre definida en todo el dominio¹. Para acelerar la convergencia hacia los valores esperados habitualmente utilizan criterios de optimización basados típicamente en distancias definidas sobre el espacio de funciones.
- Los *operadores estadísticos* más simples son funciones de densidad de probabilidad; los criterios de decisión más sencillos son de tipo bayesiano que proporcionan una realimentación entre lo que ya sabemos (probabilidades a priori) y la información que vamos aprendiendo (probabilidades a posteriori). La utilización de modelos de inferencia más avanzados sobre espacios no paramétricos permite extender los operadores precedentes a situaciones más generales usando lógica difusa (fuzzy) o relaciones de semejanza o proximidad que no pueden ser especificados en un contexto geométrico o funcional. Para acelerar la convergencia hacia patrones esperados se utilizan diferentes criterios de optimización que extienden la Estimación basada en Máxima Verosimilitud (MLE).

¹La indeterminación se puede representar en términos de funciones racionales

La utilización de un número finito k de detectores como los descritos en los items precedentes da lugar a *descriptores*, es decir, si pensamos en un detector como si fuera una función f_i , un descriptor sería una aplicación con k componentes f_i . La variabilidad de los detectores dependiendo del punto base da lugar a *campos*; la variación de la información proporcionada por dichos detectores da lugar a *flujos*. De acuerdo con el enfoque presentado en el **Capítulo 2**, hay campos escalares (representados localmente por funciones), vectoriales (conjuntos finitos de campos escalares), covectoriales (duales de los vectoriales) o tensoriales (productos formales de todos los anteriores), según un orden creciente de complejidad. La *clasificación* se lleva a cabo en términos de isomorfismos entre las estructuras superpuestas asociadas a los diferentes tipos de campos.

Idealmente el soporte para todos los tipos de campos (escalares, co-vectoriales, tensoriales) que acabamos de describir es una variedad diferenciable M . Las estructuras superpuestas más simples están dadas por fibrados vectoriales y, en casos “más estructurados” por fibrados principales; en general, sólo se tiene una estructura como fibración localmente trivial. En la práctica, los procedimientos de “suavizado” (minimización del ruido, generación automática de PL-estructuras superpuestas, alisamiento a trozos para el soporte y las funciones, etc) sólo permiten contar con una PS-estructura para el modelo M que, en el mejor de los casos, es una variedad.

Algunos problemas difíciles que habitualmente se ignoran consiste en que

- Los *espacios* de “formas” (asociadas a las apariencias) *no son metrizables*. Para intentar evitar este problema, habitualmente, sólo se utiliza una realimentación entre diferentes métricas locales; un ejemplo típico es la realimentación entre las distancias L^1 (para suprimir outliers, p.e.) y L^2 (para ajustar de forma más precisa los resultados).
- La información disponible es incompleta. En este caso, el recurso a procesos estocásticos es inconsistente, pues por un lado se ignora habitualmente el modelo diferencial estructural a perturbar y, por otro, se ignoran la “adyacencia” entre aspectos básicos (funcionales) de los datos con los que se trabaja. En nuestra aproximación la información disponible se trata en términos de subespacios k -dimensionales L en un espacio ambiente N -dimensional (la dimensión puede ser infinita), es decir, en términos de Grassmannianas, $Grass(k, N)$, sobre los cuales se diseñan diferentes estrategias diferenciales (eventualmente estocásticas) para identificar la convergencia hacia los valores de equilibrio de sistemas dinámicos (representados globalmente por fibrados vectoriales o por fibraciones más generales).

Los argumentos presentados más arriba sugieren un enfoque topológico que sea compatible con variabilidad en la “forma” (representada por deformaciones de variedades), las propiedades de los funcionales definidos sobre objetos (deformaciones de campos definidos sobre variedades) y las relaciones entre objetos (representadas por morfismos entre estructuras superpuestas). Lamentablemente, en la literatura no existe un enfoque unificado de estas características. Por ello, en este TFG se han utilizado materiales no publicados procedentes del módulo 4 (Reconocimiento) del Curso de Especialista en Visión por Computador (CEViC) redactados por Javier Finat.

3.1. Métodos topológicos en Reconocimiento.

El Reconocimiento Automático de objetos en Visión Computacional debe ser compatible con “pequeñas” deformaciones de los objetos y de los funcionales (detectores y, con más generalidad, descriptores). Asimismo, con diferentes tipos de ruido (sólo consideramos ruido blanco, es decir, perturbaciones aleatorias de una señal) e incertidumbre. Por ello, el enfoque que desarrollamos en esta sección pone énfasis en los aspectos estructurales de los modelos que afectan tanto al soporte (una variedad en el caso más simple), como a los diferentes tipos de campos (escalares, vectoriales, tensoriales) definidos sobre el soporte.

Para los objetos, las deformaciones afectan a la “forma” o apariencia del objeto y pueden ser aparentes (asociadas a diferentes localizaciones relativas de la cámara con respecto al objeto) o reales (debido a propiedades de elasticidad o plasticidad de los objetos considerados); por ello, se dedica una primera subsección a la noción de forma. Para los funcionales generalizados asociados a flujos de campos las deformaciones más simples se describen en términos de

familias uniparamétricas de funciones, campos vectoriales o formas diferenciales, relegando a un segundo plano el enfoque basado en tensores debido a su mayor complejidad formal.

Las *deformaciones aparentes* más simples proceden de la interpretación ideal de una cámara como un modelo de proyección central cuya imagen es un modelo de perspectiva. En este caso, deformaciones aparentes se estiman en términos de transformaciones afines que se conectan entre sí usando transformaciones proyectivas. El problema radica en las herramientas a utilizar para estimar dichas transformaciones. La utilización de grupos de transformaciones permite dotar de estructura localmente homogénea a la información estructural que se va generando sobre el agrupamiento de datos de cara a construir un modelo.

Para formalizar la estrategia de estimación, se adopta un esquema de complejidad creciente basado en “momentos” centrales de orden superior para los objetos. Los momentos permiten evaluar la concentración o dispersión de los datos en torno a “valores centrales” cuyo soporte (inicialmente desconocido) puede tener dimensión cero (puntos aislados) ó uno (trozos de líneas o de curvas); ambos juegan el papel de “organizadores” (en el marco estadísticos) o de “atractores” (en el marco diferencial) para la información que debe ser ordenada para su ajuste con respecto a un patrón (clase de equivalencia con respecto a una relación).

Desde un punto de vista más general, cuando el desplazamiento tiene lugar sobre una PS-variedad M , el “transporte” de una cantidad multilineal (representada por un tensor) a lo largo de un camino γ_i (sobre la escena o sobre el espacio de representación) se representa mediante una conexión afín que no requiere ningún tipo de métrica.

3.1.1. Sobre el concepto de “forma”.

La noción de “forma” es elusiva y presenta diferentes descripciones en la literatura dependiendo del contexto topológico, geométrico o estadístico, entre otros. El contexto más general es el topológico, pero las clases de equivalencia (módulo homeomorfismo) son demasiado generales para ser útiles; no obstante, existen ya aproximaciones computacionales a la construcción de homeomorfismos locales en términos de la integración de campos vectoriales.

Enfoque lógico.

Las características topológicas más simples son de tipo conjuntista y afectan a propiedades de igualdad, inclusión, complementariedad y las operaciones booleanas básicas (unión e intersección, y sus composiciones). Todas ellas son fácilmente implementables en cualquier tipo de lógica, incluyendo la lógica difusa (fuzzy). Como sus versiones en lógica básica (tablas de verdad) y lógica proposicional de primer orden son bien conocidas, nos limitamos a comentar el enfoque basado en lógica difusa.

Para ello, denotemos mediante $\mathcal{F}(X)$ el conjunto de todos los subconjuntos difusos de X , es decir, $m \in \mathcal{F}(X)$ si y sólo si, $m : X \rightarrow [0, 1]$ (interpretable como una función de probabilidad). Si m_A denota la función asociada a un subconjunto A tenemos las siguientes descripciones ²:

- *Igualdad:* $A = B \Leftrightarrow m_A = m_B$.
- *Inclusión* $A \subseteq B \Leftrightarrow m_A \leq m_B$.
- *Complementariedad:* $m_{\overline{A}}(x) = 1 - m_A(x)$.
- *Intersección:* $m_{(A \cap B)}(x) = \min[m_A(x), m_B(x)]$.
- *Unión:* $m_{(A \cup B)}(x) = \max[m_A(x), m_B(x)]$.

Enfoque topológico.

Las operaciones básicas conjuntistas descritas más arriba permiten caracterizar propiedades topológicas de objetos. De cara a su implementación computacional es importante evitar el uso de cuantificadores (existenciales ó universales), salvo que se disponga de criterios para su verificación en un número finito de pasos. El carácter discreto del soporte digital para las

²Aparentemente, la primera descripción se debe a Zadeh (1965).

imágenes y el diseño de procedimientos rapidísimos de barrido para toda la imagen permite evitar ambos problemas³.

A mayores, frecuentemente es necesario contar con operaciones lógicas sobre regiones contenidas en cada imagen digital. En este caso, se utilizan operadores morfológicos básicos (suma o diferencia de Minkowski) para las operaciones booleanas más simples (dilatación o erosión); su composición (cierre vs apertura) permite obtener una primera simplificación de regiones contenidas en imágenes que proporciona las regiones r_α a analizar como candidatas a formar parte de objetos $b_\alpha^{beta} = \cup_{\alpha \in A} r_\alpha$.

Sobre esta representación se pueden aplicar los filtros usuales de procesamiento y análisis de imagen dados localmente por convoluciones (productos tensoriales “pesados”) de funciones discretas definidas sobre el soporte de la imagen. Estos filtros permiten extraer tanto las regiones radiométricamente homogéneas (nivel de intensidad “similar” o distribución de color similar para el caso de texturas) como sus bordes a los que se etiqueta como “siluetas” $s_\alpha := \partial r_\alpha$ (borde de r_α).

Aproximación estadística.

El enfoque estadístico más simple utiliza “momentos de orden bajo” para los estimadores de la forma procedente del enfoque topológico. Estas “cantidades” son el resultado de aplicar herramientas estadísticas básicas (asociadas a histogramas, p.e.) para evaluar la concentración de datos en torno a valores centrales (medias, modas, medianas), direcciones lineales (eventualmente orientadas) o figuras geométricas muy sencillas (correspondientes a curvas de grado bajo, habitualmente $d \leq 4$).

Para una función continua $f(x, y)$ definida sobre el plano (correspondiente a la función de intensidad en la escala de grises I_g , p.e.) el momento de orden $(p + q)$ se define como

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

para $p, q = 0, 1, 2, \dots$. En particular, para el caso discreto si f es la función de intensidad $I(x, y)$ en la escala de grises, los momentos M_{ij}

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$

El resultado crucial que muestra el interés de los momentos está reflejado en [28].

Proposición 3.1 *Si $f(x, y)$ es una función continua a trozos y con soporte compacto, entonces existen los momentos M_{pq} de cualquier orden, y la colección de momentos M_{pq} está unívocamente determinada por $f(x, y)$. Recíprocamente, la colección M_{pq} determina de forma unívoca $f(x, y)$.*

Detectando la forma geométrica.

En la práctica para las regiones r_α contenidas en una imagen sólo se consideran unos pocos momentos de orden bajo. Frecuentemente, se toman “momentos centrales” correspondientes a una “traslación” de los momentos descritos en el apartado anterior a un centroide dado por los momentos de orden cero $(\bar{x}, \bar{y}) = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}})$ de cada una de las regiones casi-homogéneas (viabilidad por debajo de un umbral) segmentadas en la imagen.

Los momentos proporcionan valores numéricos asociados a las formas. Por ello, se pueden reinterpretar como tipos avanzados de detectores de la forma. Además de los momentos de orden cero descritos más arriba, los momentos de orden 1 (interpretables en términos de rectas de regresión) se pueden interpretar como detectores de orden 1.

Un ejemplo un poco más sofisticado está dado por los momentos de orden 2 que permiten calcular la elipse que mejor se ajusta a la distribución de los datos (inicialmente dispersos) asociados a la región. Una primera aproximación región se puede calcular en términos de una envolvente convexa aproximada (calculable en tiempo real) a la que se ajusta la elipse

³Cuando la imagen es demasiado grande (imágenes aéreas, p.e.) o cuando la tasa de captura es superior a la de procesamiento (en fenómenos que ocurren a gran velocidad, p.e.) se utiliza muestreo sobre la imagen o sobre el número de imágenes.

de inercia basada en la POSE (Position, Orientation, Size, Eccentricity) asociada a los momentos de orden dos.

Este ejemplo se utiliza para representaciones intermedias de objetos articulados complejos como el cuerpo humano donde la cabeza, el tronco y las componentes de cada extremidad se representan mediante elipses conectadas entre sí. El árbol poligonal obtenido conectando los semiejes mayores de cada elipse proporciona una representación esquelética que se utiliza para la simulación de modelos biomecánicos para la dinámica del cuerpo humano.

Desde un punto de vista formal, es importante destacar que la selección del “orden bajo” $p+q$ de los momentos seleccionados M_{pq} corresponde a identificar los valores extremos sobre la truncación de orden $p+q$ (de la parte simétrica) del álgebra tensorial graduada en x, y . Esta observación tan simple es clave para la extensión al reconocimiento de tareas representadas por aplicaciones entre espacios de baja dimensión. Debido a su mayor complejidad, no desarrollamos este enfoque.

3.1.2. Descriptores usados en Visión Computacional.

En la literatura sobre el tema no se hace una distinción clara entre detectores y descriptores. Esta falta de distinción da lugar a una confusión entre procedimientos de tipo cualitativo (detección) (y cuantitativo (descripción) que es contraria a la metodología usada frecuentemente en AI en relación con modelos no paramétricos vs paramétricos.

Para fijar ideas, detectar la localización relativa (derecha, izquierda, arriba, abajo) de un punto con respecto a una curva en el plano resulta relativamente fácil (salvo para casos patológicos como la curva de Peano, p.e.). Basta evaluar el signo de una forma (lineal o no) en el punto ⁴. Sin embargo, detectar la localización (posición y orientación) exacta tiene un coste computacional mucho mayor.

Según nuestro enfoque los descriptores son colecciones de campos definidos en cada punto que son invariantes con respecto a un grupo G de transformaciones:

- Los *campos escalares* están dados por funciones que proporcionan información puntual sobre “puntos salientes” (junturas múltiples, máximos de intensidad) para imágenes estáticas; son las más utilizados actualmente.
- Los *campos vectoriales* permiten una variabilidad en los coeficientes para cada una de las direcciones de variación (representadas por operadores de la forma $\partial/\partial x_i$); permiten incorporar datos asociados a la cinemática correspondiente al movimiento relativo de una cámara en una escena.
- Los *campos tensoriales* son producto de campos escalares, vectoriales y co-vectoriales (duales de los vectoriales). Permiten evaluar predicados, direcciones de movimiento y evaluar características complicadas como las asociadas a una textura (modelable como una 2-forma) o gestionar propiedades radiométricas complicadas como la reflectancia de una superficie (modelable localmente como una 2-forma diferenciable).

Para fijar ideas, nos centramos en descriptores asociados a los descriptores correspondientes a los campos escalares más usados en Visión Computacional desde comienzos del siglo XXI son SIFT, SURF, GLOH y HOG. A continuación se describe brevemente cada uno de ellos.

SIFT.

La transformación de la característica invariante por cambio de escala **SIFT** (*Scale-Invariant Feature Transform*), es un algoritmo de detección de características en visión artificial para detectar y describir las características locales en las imágenes que son invariantes por transformaciones de semejanza, es decir, rotaciones, traslaciones y cambios de escala.

Este descriptor de imagen (desarrollado por David Lowe, 1999-2004) permite emparejar “hechos geométricos salientes” (junturas múltiples, p.e.) y reconocer configuraciones de puntos invariantes por dichas transformaciones. El descriptor **SIFT** es robusto en relación con las transformaciones de perspectiva (transformaciones afines que dejan invariante una recta del infinito) y pequeñas variaciones de iluminación. Experimentalmente, el descriptor **SIFT** ha

⁴Por ello, al nivel más tosco las formas (incluidas las formas diferenciales) se pueden considerar como detectores.

demostrado ser muy útil en la práctica para el reconocimiento pues permite emparejar datos del modelo del objeto y datos contenidos en la imagen en condiciones reales.

En su formulación original, el descriptor **SIFT** incluye un método para la detección de puntos de interés de una imagen en escala de grises. La estrategia básica consiste en agrupar (usando histogramas) los datos relativos a las direcciones del campo vectorial gradiente ∇I_g de imagen que representan las normales locales a la curva. Para reforzar el emparejamiento, es conveniente añadir restricciones locales en el proceso de emparejamiento asociadas a la selección de una pequeña “ventana de búsqueda” (método “windowing”).

Posteriormente, el descriptor **SIFT** se ha aplicado a nubes de puntos “semidensas” obteniendo resultados estables desde el punto de vista estadístico que incluyen aplicaciones a tareas como categorización, clasificación de textura, alineación de la imagen y biometría. Un inconveniente de este método radica en la elevada dimensión del espacio de parámetros asociado a **SIFT**; esta elevada dimensión procede de la necesidad de comprobar la invariancia con respecto a una discretización de las transformaciones de semejanza utilizadas.

SURF.

El descriptor **SURF** (*Speed-Up Robust Features*) fue desarrollado por Hebert Bay como un detector robusto y un descriptor asociado a configuraciones de puntos de interés. El descriptor **SURF** guarda cierta similitud con la filosofía del descriptor **SIFT**, si bien presenta notables diferencias relacionadas con:

- Velocidad de cálculo considerablemente superior sin ocasionar pérdida del rendimiento.
- Mayor robustez ante posibles transformaciones de la imagen.

Estas mejoras se consiguen mediante la reducción de la dimensionalidad y complejidad en el cálculo de los vectores de características de los puntos de interés obtenidos que persisten en imágenes sucesivas, siempre y cuando sigan siendo “elementos característicos” (junturas múltiples vs máximos de intensidad, p.e.).

Las diferencias más originales respecto del descriptor **SIFT** son las siguientes:

- La normalización o longitud de los vectores de características de los puntos de interés es considerablemente menor; de forma más específica, se trata de vectores con una dimensionalidad igual a 64, lo que supone una reducción de la mitad de la longitud del descriptor **SIFT**.
- El descriptor **SURF** utiliza siempre la misma imagen, la original.
- Utiliza el determinante de la matriz Hessiana para calcular tanto la posición como la escala de los puntos de interés.

HOG.

El **HOG** (*Histogram of Oriented Gradients*) es un descriptor que se utiliza para detectar objetos en el procesamiento de imágenes y visión por computador. La técnica del descriptor **HOG** permite detectar y describir localmente la orientación del campo gradiente (normal a la silueta) o del campo asociado al movimiento. Esta detección y descripción se realiza sobre una ventana de detección de imágenes o región de interés (**ROI**).

La implementación del algoritmo del descriptor es la siguiente:

1. Se divide la imagen en pequeñas regiones conectadas llamadas células y para cada celda se calcula un histograma del gradiente para identificar las direcciones u orientaciones de borde para los píxeles dentro de la célula.
2. Se discretiza cada célula en compartimentos angulares (discretización del ángulo polar θ) según la orientación local.
3. Los píxeles de cada célula contribuyen al gradiente (mediante un sistema de pesos) de la coordenada angular correspondiente.
4. Los grupos de células adyacentes se consideran como regiones espaciales llamadas bloques. La agrupación de células en un bloque es la base para la agrupación y normalización de histogramas.

5. El grupo normalizado del histograma representa el histograma de bloque. El conjunto de estadísticos asociados a estos histogramas de bloques representa el descriptor.

A la vista de la diversidad de datos o dispersión que pueden presentar los histogramas, en la práctica, se agrupan los datos en torno a valores centrales (media, moda, mediana) más frecuentes (algoritmos de K-medias) y se realiza algún tipo de agrupamiento (suavizado vs escalonado) en torno a dichos valores.

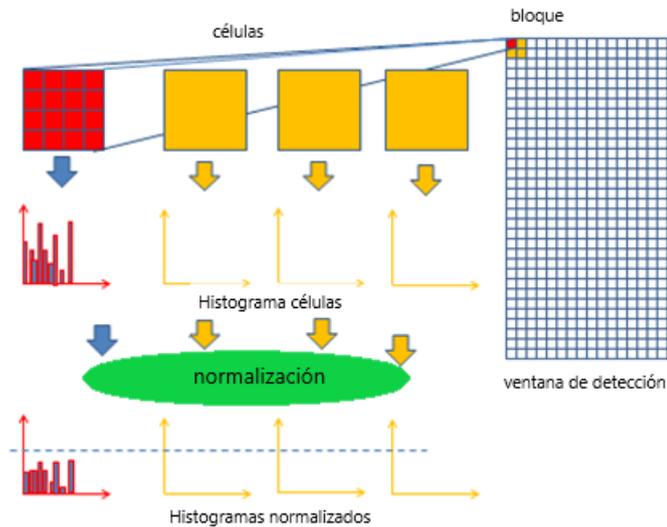


Figura 3.1: Esquema de la implementación del algoritmo.

GLOH.

El descriptor local de **GLOH** (*Gradient Location and Orientation histogram*) está diseñado para aumentar la robustez del conocido descriptor **SIFT**, que atenúa tanto la apariencia local como la información de posición. Similar al descriptor **SIFT** o al descriptor **HOG** incorporando la ventaja de localización de los elementos más significativos dentro del campo gradiente (habitualmente normalizado). También se basa en evaluar histogramas locales bien normalizados de las orientaciones de gradiente de imagen en una rejilla densa.

3.1.3. Una interpretación en secuencias de video.

En secuencias de vídeo la percepción básica corresponde a regiones cambiantes con una tasa de variación diferente dependiendo de la profundidad, la orientación relativa y la posibilidad de movimiento propio para elementos detectados en la secuencia de vídeo. En ausencia de una información bien definida sobre dichas regiones, es conveniente introducir una jerarquía con un árbol de decisión que facilite la aplicabilidad de diferentes criterios de búsqueda. Esta jerarquía debe incorporar las diferentes posibilidades para las tasas de variación de datos móviles.

Una primera distinción afecta al carácter estático o móvil de la cámara de vídeo con la que se captura la información.

Segmentación de video.

En una secuencia de vídeo, el fondo *BG* (Background) se caracteriza como el conjunto de regiones sin movimiento para una cámara fija, o bien dotadas de movimiento relativo inercial en relación con el movimiento propio de la cámara (egomotion). El complementario del fondo es el primer plano *FG* (Foreground).

La segmentación de vídeo consiste en la descomposición de cada frame de la secuencia en unión de regiones contenidas en el fondo *BG* (Background) y en el primer plano *FG*.

Dichas regiones se caracterizan en términos del histograma orientado asociado a la coordenada polar correspondiente a los vectores móviles detectados que representan pequeños

desplazamientos evaluados entre frames consecutivos. Habitualmente, se toman vectores unitarios, para minimizar las distorsiones debidas a diferente profundidad en la escena; en una primera aproximación, la profundidad es inversa de la disparidad, es decir, de la diferencia en la localización de datos homólogos en imágenes alineadas.

Sustracción del fondo.

Para una cámara estática y suponiendo que la intensidad de la luz no varía entre dos frames consecutivos, la sustracción del fondo consiste en calcular la diferencia $I_t(x, y) - I_{t-1}(x, y)$ entre imágenes correspondientes a dos instantes consecutivos. La región nula (BG) corresponde al fondo estático, mientras que el soporte de la región no-nula (FG) corresponde a los objetos móviles.

Como la información correspondiente a los objetos resultantes de esta operación de sustracción pueden estar corrompidos por el ruido o presentar irregularidades en el borde de las regiones debido a un efecto de emborronamiento (blurr), es conveniente aplicar operadores morfológicos tales como suma o diferencia de Minkowski, o sus versiones iteradas (cierre y apertura). De este modo, se obtiene una primera aproximación al desplazamiento de una silueta móvil $s_\alpha(t) = \partial b_\alpha(t)$ para cada objeto móvil $b_\alpha(t)$ presente en la escena dotado de movimiento propio⁵.

La extracción y el seguimiento de ‘puntos salientes’ (esquinas en los PL-modelos subyacentes) contenidos en la silueta $s_\alpha(t)$ proporcionar una estimación tosca de las características cinemáticas para cada componente del objeto $b_\alpha(t)$ eventualmente articulado (como ocurre con las extremidades de una persona, p.e.). La estimación de las velocidades (resp. aceleraciones) en cada punto de control se expresan en términos de las tasas de variación de primer orden (resp. de segundo orden) para cada uno de los puntos significativos en dos (resp. tres) frames consecutivos.

De este modo se obtiene una descripción tosca de las características cinemáticas asociadas al movimiento de objetos para una cámara móvil. Otros procedimientos más avanzados que permiten relacionar el movimiento de varios puntos pertenecientes al mismo objeto o bien a objetos relacionados utilizan versiones avanzadas (IEKF) de Filtros de Kalman⁶.

Segmentación de vídeo para cámara móvil.

Recordemos que la *segmentación de imagen* es una descomposición de cada imagen en unión de regiones con características radiométricas similares (módulo un umbral de tolerancia); en otras palabras, la segmentación de imagen produce como salida un mapa de regiones r_α . Una secuencia de vídeo se puede considerar inicialmente como una colección finita de frames o imágenes consecutivas que representamos mediante I_t .

Por ello, la estrategia descrita para segmentación de imagen se puede aplicar a cada una de las imágenes de una secuencia de vídeo, dando lugar a un mapa de regiones móviles $r_\alpha(t)$. Este proceso recibe el nombre de *segmentación espacial* de una secuencia de vídeo.

La *segmentación temporal de vídeo* consiste en una descomposición a lo largo de la línea de tiempo en colecciones de imágenes que presentan ‘baja variación’ en el histogram de frecuencias $\mathcal{H}(I_t)$ de las imágenes I_t . La descripción de ‘baja variación’ depende del contexto. En particular, la (des)aparición de regiones móviles proporciona modelos a bajo nivel de los ‘eventos’ detectables que separan minisecuencias con un contenido similar.

A las imágenes que separan dos minisecuencias de video con contenidos similares se les llama disparos (shots) y corresponden a los elementos críticos de una aplicación del espacio de histogramas de frecuencias sobre el espacio de parámetros cuyas componentes son los estadísticos del histograma. Por ello, admiten un tratamiento en términos de una extensión de la teoría de Morse al análisis del lugar discriminante de una aplicación.

La *segmentación espacio-temporal de vídeo* consiste en la descomposición de una secuencia de vídeo en una colección de regiones móviles significativas $r_\alpha(t)$ cuyo ‘enlazado’ proporciona una colección finita de superficies espacio-temporales, identificando los posibles eventos asociados a (des)aparición de regiones móviles (shots). La utilización de esta segmentación es

⁵Este procedimiento no permite detectar objetos eventualmente móviles que están estáticos en instantes consecutivos.

⁶Estas cuestiones se tratan con más detalle en el módulo B_{33} (Análisis del Movimiento) del Curso de Especialista en Visión por Computador (CEViC) de Javier Finat.

clave para cuestiones de etiquetado automático, así como para la compresión y codificación de vídeo (y los procesos inversos).

Desde el punto de vista del reconocimiento, este último tipo de segmentación es la más interesante, pues permite centrar la atención en dar respuesta a los eventos significativos que puedan aparecer en relación con la gestión de la información. Para ello, se requieren elementos más avanzados de Inteligencia Artificial que permitan relacionar aspectos locales y globales; en nuestro enfoque dichas relaciones se describen en términos de fibrados (ver sección 2 de este capítulo). La parte de Inteligencia Artificial relevante para este problema se presenta a partir de la sección 3 de este capítulo.

Actualizando la información.

Para una cámara móvil el problema es más complicado de resolver, pues todas regiones contenidas en la secuencia de vídeo tienen un movimiento aparente, incluso aunque toda la escena sea estática y lo único que se mueva sea la cámara. Además, las características del movimiento no están distribuidas de manera homogénea en las regiones. De una forma genérica, los objetos situados en la parte periférica de las imágenes $I_t(x, y)$ de la secuencia de vídeo tienen una tasa de variación mucho más elevada que los objetos situados en la parte central de la imagen.

Así, p.e. en una escena de conducción en autovía el punto de fuga para un modelo de perspectiva frontal está aparentemente inmóvil, mientras que los objetos situados a derecha o izquierda desaparecen muy rápidamente. Análogamente, cuando se gira en una calle hacia la derecha, la tasa de variación de los objetos situados a la derecha es mucho menor que la de los situados a la izquierda.

Por ello, es necesario descontar el efecto de la profundidad o de la orientación relativa con respecto al movimiento de la cámara. La estrategia de segmentación debe tener en cuenta la distribución asociada a la cinemática relativa asociada a las diferentes regiones móviles que aparecen en la segmentación de vídeo. Para fijar ideas, empezamos con el caso más simple.

Supongamos una cámara en movimiento rectilíneo a lo largo de una escena urbana simplificada (sin texturas complicadas debidas a vegetación, p.e.). La extracción de elementos lineales y el alineamiento de mini-segmentos en líneas de perspectiva permite generar de forma automática un mapa de perspectiva. Además de los elementos estructurales 0-dimensionales de la imagen (puntos de fuga) se tiene un número reducido de juntas múltiples (puntos triples, típicamente) que son fáciles de extraer usando filtros específicos (filtro de Harris, p.e.).

La estimación tosca del movimiento, permite poner en correspondencia elementos comunes contenidos en una vista frontal. Para alinear dos imágenes consecutivas (muestreadas dentro de la secuencia de vídeo) basta aplicar una transformación de escala vinculada al movimiento relativo que permite alinear las dos imágenes y calcular el desplazamiento relativo de los elementos homólogos.

A pesar de su simplicidad, el ejemplo anterior muestra la estrategia a seguir que se basa en los items siguientes:

1. Realizar una segmentación de la secuencia con dos hilos (threads) en paralelo: temporal para un hilo, espacial para la otra.
2. Fusionar las dos segmentaciones en una segmentación espacio-temporal.
3. Estimar las características del movimiento en términos de una transformación rígida (composición de traslaciones y rotaciones planares) en la escena.
4. Descontar el efecto del movimiento efectuado entre dos instantes consecutivos $t - 1$ y t para facilitar el alineamiento entre imágenes.
5. Calcular las características cinemáticas del movimiento para cada una de las regiones segmentadas $r_\alpha(t)$ en términos de diferencias finitas para la localización de "puntos salientes" (juntas triples y máximos de intensidad, típicamente).
6. Propagar las características cinemáticas estimadas sobre una representación simplificada de la escena.

7. Predecir posibles eventos que puedan alterar la realización del movimiento planeado.
8. Evaluar posibles incidencias y resolverlas mediante árboles de decisión.

Este esquema razonable no contiene grandes novedades con respecto a los modelos utilizados de forma experimental para conducción en autovías monitorizadas y con tráfico restringido a finales de los noventa. Sin embargo, lo que se pretende ahora es bastante más ambicioso, pues se desea aplicarlo a tráfico real en escenarios urbanos donde la mayoría de los vehículos no tienen dispositivos artificiales inteligentes. Por ello, es conveniente incorporar efectos adicionales inesperados, incluyendo el carácter imprevisible (incluyendo comportamientos estúpidos o irracionales) de conductores humanos.

Este es un objetivo a medio plazo; por ello, la sección siguiente está dedicada a incorporar modelos “globales” que puedan ser compartidos por diferentes agentes, lo cual requiere herramientas de pegado tanto para el soporte (una variedad en el caso más favorable, p.e.) como para las estructuras superpuestas (fibrados vectoriales en el caso más favorable). Como es lógico, el tratamiento de diferentes tipos de acciones se lleva a cabo en términos de campos sobre variedades. Por ello, el marco para la unificación está dado por el complejos graduados “fluctuantes” definidos sobre un Álgebra Tensorial.

3.1.4. Modelo tensorial para aspectos funcionales del ciclo PAC.

En la subsección anterior se ha mostrado como la información móvil detectada correspondiente a los objetos del primer plano se puede describir en términos de campos vectoriales (representados por trayectorias para los puntos de control) o bien en términos de formas diferenciales (correspondientes a la evaluación de propiedades que representamos mediante hiperplanos cambiantes como soporte para las formas).

El producto formal de s campos vectoriales y de r formas diferenciales es un tensor de tipo (r, s) . Por ello, la estimación simultánea de trayectorias y atributos está dada geoméricamente por un tensor. El uso de tensores es cada vez más común en Visión Computacional. Dos ejemplos típicos están dados por el tensor de estructura y el tensor de estructura generalizado a los que dedicamos los dos primeros apartados de esta subsección. Un tensor de uso común en Reconstrucción 3D es el tensor multilineal como extensión de las matrices fundamental o esencial (para dos vistas) o el tensor trilineal (para tres vistas).

El último apartado está dedicado al enfoque de Pelionisz [30] en relación con el modelado funcional del cerebro humano el cuál comentamos muy brevemente.

Tensor de estructura.

En Matemáticas y sus aplicaciones a Dinámica de Fluidos, Robótica, Visión Computacional o Informática Gráfica (los 4 módulos etiquetados como B_i en los apuntes de Javier Finat), el tensor de estructura está representado localmente por las matrices correspondientes a los momentos de orden 2 (descritos en la sección 1 de este capítulo).

En este trabajo hablamos sobre el tensor de estructura en el caso 3D.

Definición 3.1 Para una función I de tres variables $p = (x, y, z)$, el *Tensor de estructura* es

$$S_w(p) = \int w(r)S_0(p-r)dr,$$

donde

$$S_0(p) = \begin{bmatrix} (I_x(p))^2 & I_x(p)I_y(p) & I_x(p)I_z(p) \\ (I_x(p))^2 & I_x(p)I_y(p) & I_x(p)I_z(p) \\ (I_x(p))^2 & I_x(p)I_y(p) & I_x(p)I_z(p) \end{bmatrix},$$

donde I_x, I_y, I_z son las derivadas parciales de I y w es algo fijo conocido como función ventana.

La versión discreta de este tensor es

$$S_w[p] = \sum_r w[r]S_0[p-r],$$

donde

$$S_0[p] = \begin{bmatrix} (I_x[p])^2 & I_x[p]I_y[p] & I_x[p]I_z[p] \\ (I_x[p])^2 & I_x[p]I_y[p] & I_x[p]I_z[p] \\ (I_x[p])^2 & I_x[p]I_y[p] & I_x[p]I_z[p] \end{bmatrix}$$

Cuando se evalúan formas geométricas complicadas, pueden aparecer varias direcciones predominantes, que pueden evolucionar a lo largo del tiempo, pudiendo ser compatibles con fenómenos de bifurcación o de agrupamiento (como ocurre con fenómenos típicos de la Dinámica de Fluidos, incluyendo escenas de tráfico).

Este tensor permiten representar las direcciones predominantes (semieje mayor) del elipsoide de inercia asociado a una función o, con más generalidad, a una aplicación.

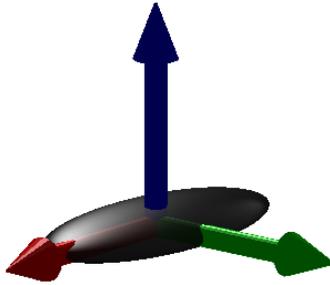


Figura 3.2: Representación elipsoidal 3D del Tensor de estructura.

Tensor de estructura generalizado.

Las posibles deformaciones de los fenómenos observados en las áreas citadas más arriba pueden afectar a los objetos o bien a su comportamiento en relación con otros objetos (modelos de interacción). Afortunadamente, la matriz jacobiana para la aplicación que representa dichas acciones proporciona una solución general al problema.

Además, la matriz jacobiana J describe la diferencial del cambio de carta, por lo que facilita un soporte para extender la versión local (relativa a un abierto U de trivialización de una estructura superpuesta) al caso “global” utilizando el solapamiento sobre dos abiertos. La estratificación por el rango de la matriz jacobiana induce una descomposición celular (extendiendo la de Morse asociada al campo gradiente) en la que los valores críticos se reemplazan por el análisis del lugar discriminantes.

Asimismo, la utilización de la matriz jacobiana proporciona una representación natural de procesos de realimentación asociados a la diferencial de la aplicación proyección $\pi : E \rightarrow B$ para fibraciones topológicas como las que se presentan en Robótica en relación con la aplicación de transferencia $\pi : \mathcal{C} \rightarrow \mathcal{W}$ (donde \mathcal{C} es el espacio de configuraciones y \mathcal{W} es el espacio de trabajo) o en el diseño de Sistemas Expertos para el ciclo Percepción Acción, cuya componente básica es la fibrición $\mathcal{P} \rightarrow \mathcal{A}$ (donde \mathcal{P} es el espacio de percepción y \mathcal{A} es el espacio de acción). En ambos casos, la realimentación está dada por el producto de matrices JJ^T ó bien $J^T \cdot J$ extendiendo la aproximación clásica basada en ∇f^2 .

La extensión de este enfoque a aspectos dinámicos requiere una formulación intrínseca de la diferencial de orden 2 que excede el marco de este TFG.

Modelado funcional del cerebro según Pellionisz.

Pellionisz afirmaba que las señales del cerebro llevan “información biológica”, pero que la definición matemática de este término es inexistente y que además el núcleo de la función cerebral reside en su representación por un modelo (1983).

Por ello, Pellionisz desarrolla una teoría cerebral representacional conceptual y formalmente homogénea que se basa en la filosofía y la estructura algebraica que considera más adecuada basada en Álgebra Tensorial.

El concepto específico y el formalismo en la teoría de los campos tensoriales es que la función cerebral se implementa mediante transformaciones de redes neuronales que representan objetos físicos por vectores generales multidimensionales duales, sensoriales y de tipo motor (matemáticamente, estos son tensores covariantes y contravariantes). La teoría sobre

tensores se acerca al estructural cerebro-mente o a la entidad funcional desde el punto de vista de la geometría funcional multidimensional, utilizándolo para construir una teoría de representación geométrica.

Matemáticamente hablando, la teoría sobre los tensores se basa en el hecho de que la estructura de la geometría física de los organismos determina aquellos sistemas de coordenadas naturales que son intrínsecos a la expresión de su función. En [30] se muestra un esquema con el ejemplo de los movimientos del cuello a través de contracciones musculares que demuestra la posibilidad de aplicación de esta teoría de tensores sobre las funciones del sistema nervioso central ó CNS (*Central Neuronal System*).

Durante mucho tiempo ha sido costumbre representar matemáticamente las redes neuronales masivamente paralelas del CNS por matrices que se convierten, desde el enfoque de Pellionisz, en implementaciones concretas específicas de los operadores de tensores específicos. La solución propuesta por Pellionisz (1984) frente a las principales dificultades que plantea una coordinación/transformación general del sistema (por ejemplo, el sistema motor del cuello) utiliza la diferencia entre las representaciones covariantes y contravariantes del movimiento deseado, ambas determinadas por la geometría muscular. La representación covariante se puede establecer de forma única proyectando el vector de movimiento sobre cada uno de los ejes musculares. El problema es encontrar su inversa única, la representación contravariante. En un sistema sobredeterminado el problema no es que esto no exista, sino que hay un número infinito de inversas. Pellionisz propone que el CNS elige una solución única, la inversa de Moore-Penrose⁷.

En el capítulo 4 de [30] se muestran ejemplos sobre modelos de redes tensoriales como el sistema motor del cuello o el sistema visual de los ojos.

En la actualidad, existe un interés cada vez mayor en la computación por las redes neuronales. Por lo tanto, puede plantearse la cuestión de cómo el enfoque tensorial se relaciona con esta tendencia en desarrollo. En primer lugar, mientras que otros enfoques tienen por objeto interpretar la función de las redes neuronales imaginarias que carecen de una estructura específica (caracterizada por un conjunto de interconexiones “todo a todo”), el enfoque tensor se ocupa de las existentes, no redes neuronales arbitrarias. Además, este enfoque proporciona medios formales para manejar tanto su estructura como su función, en términos de transformación de expresiones vectoriales generales. Tal vez la diferencia más importante es, sin embargo, que el formalismo tensor define la naturaleza matemática intrínseca del cálculo, afirmando que los cálculos realizados por las redes son transformaciones de vectores generalizados que se expresan en coordenadas intrínsecas. Así, en el caso del cerebelo, por ejemplo, es posible indicar la función general de circuitos cerebelares específicos (por ejemplo, en diferentes especies).

3.2. Una reformulación de Sistemas Expertos.

En este apartado se comentan brevemente los tres tipos de Sistemas Expertos de carácter general que se pueden superponer a las ANN. Estos tres tipos están dados por Algoritmos Genéticos (GAs), Programación Evolutiva (EP) y Mapas Auto-Organizados (SOM) de Kohonen. El más potente y universal corresponde a los SOM, además de ser el más próximo a la formulación en términos de fibrados vectoriales. Por ello, este apartado está centrado sobre todo en mostrar algunas aplicaciones básicas de los SOM al problema de Reconocimiento en imágenes digitales.

Desarrollos tecnológicos recientes (ANN con decenas o cientos de capas, paralelización de datos y procesos, diferentes tipos de neuronas en la red, realización de procesos en algunas capas sin esperar a la salida final, etc) proporcionan un soporte para modelos de reconocimiento más avanzados que mejoran en un factor superior a cien el rendimiento de los métodos de finales del S.XX. Deep Learning proporciona el marco apropiado para todos estos desarrollos.

Lamentablemente, apenas existen manuales que incorporen una introducción a este tema. Asimismo, existe una elevada casuística y una ausencia de modelos estructurales. La propuesta que se presenta a continuación proporciona algunos elementos que facilitan el desarrollo

⁷En álgebra lineal, una pseudoinversa A^+ de una matriz A es una generalización de la matriz inversa. El tipo más conocido de pseudo inversa es la *inversa de Moore-Penrose*.

de dichos modelos estructurales que combinan aspectos robustos y adaptativos de forma simultánea. La propuesta se basa en una “traslación” de conceptos básicos de la Teoría de Fibrados que, a pesar de su carácter elemental, no está aún desarrollada en la literatura.

La aproximación que se propone en este trabajo consiste en realizar un tratamiento de estos tres tipo de elementos básicos en la que

- Los *descriptores* correspondan a (transformaciones entre) sistemas coordenados locales para un espacio base B de un fibrado.
- Los *detectores* a la evaluación numérica de datos (contenidos en descriptores) como si fueran las componentes de campos representados localmente por vectores variables.
- Los *clasificadores* correspondan a transformaciones sobre las fibras que “empaquetan” los detectores y que se reinterpretan como (clases de) funciones de transición de fibrados vectoriales asociados.

En las subsecciones siguientes se formalizan los descriptores, detectores y clasificadores en el contexto de Fibrados Vectoriales. Es importante destacar que todos ellos se pueden definir tanto sobre el espacio base como sobre el espacio total del fibrado. En el segundo caso y a la vista del carácter localmente trivial del fibrado, supondremos inicialmente que están dados por un par de elementos (detectores, descriptores, clasificadores) definidos sobre (un abierto de) la base y la fibra.

3.2.1. Detectores.

El objetivo de un detector es “descodificar” una señal de acuerdo con un patrón heredado o aprendido. El ajuste más sencillo con respecto al patrón corresponde a un procedimiento de regresión lineal con respecto a las “direcciones predominantes” (autovectores asociados a autovalores de módulo máximo de operadores lineales). El aprendizaje de dicho ajuste se puede llevar a cabo mediante modelos elásticos que facilitan el ajuste de parámetros. Por ello, el ajuste con respecto a una “forma” o una “tarea” se puede entender como un proceso definido por campos (escalares, vectoriales, tensoriales) que actúan sobre un soporte elástico.

En el caso más simple, los *detectores* están asociados a la evaluación numérica de datos correspondientes a funciones f_i (discretas, continuas, de probabilidad) definidas sobre un espacio topológico X de muestras. Es necesario minimizar la diferencia con respecto al valor esperado o, con más generalidad, con respecto al soporte de la distribución dada por los campos. Las funciones a evaluar se pueden interpretar como las *componentes de una aplicación F* (función vectorial) que no tienen por qué estar definidas globalmente (debido a falta de información, p.e.). La formulación probabilística permite incorporar la incertidumbre relacionados con los datos, su distribución o su evolución.

Por ello, suponemos inicialmente que cada *detector básico* está dado como una función real $f_i : X \rightarrow \mathbb{R}$, aunque su rango pueda ser \mathbb{Z}_2 (operadores lógicos), \mathbb{Z} (precisión entera), \mathbb{Q} (precisión racional) o cualquier dominio de \mathbb{R} . Por ello, en términos geométricos cada función es una sección del fibrado trivial ε_X^1 sobre el espacio muestral X . Una colección finita de detectores básicos es una aplicación; su variación espacio-temporal está dado por una distribución de campos vectoriales o co-vectoriales que se aborda en términos de descriptores (ver subsección siguiente).

En todos los casos es necesario proporcionar una estimación del error, así como términos que permitan corregir tanto los falsos positivos (detección de un objeto como correcto, cuando es falso) como los falsos negativos (detección de un objeto como falso, cuando es verdadero). Estas cuestiones afectan a aspectos más profundos de las redes neuronales artificiales (ANN) que no se consideran en esta memoria por razones de espacio.

Detectores discretos.

Para los elementos muestreados, los detectores discretos pueden ser de tipo lógico o bien tomar valores aislados en un rango finito de posibles valores. En el primer caso, se aplican métodos de la lógica de clases para la toma de decisiones en los correspondientes sistemas expertos. En el segundo caso, además de evaluar el valor numérico, se consideran los diferentes estadísticos básicos asociados al histograma de valores obtenidos.

Los algoritmos típicos para gestionar la información relativa a valores aislados extraen información para hechos muy relevantes (máximos de intensidad en una imagen, p.e.) o bien proporcionan una evaluación de la concentración en torno a los valores más significativos. Un ejemplo típico de los segundos está dado por los algoritmos de K-means, donde “means” no es necesariamente la media aritmética, sino que también puede referirse a otros valores “centrales” como la moda o la mediana. La aplicación del método de K-means proporciona criterios para facilitar un agrupamiento (clustering) en torno a los valores “medios”.

Detectores continuos.

En este caso, se supone que el soporte digital para llevar a cabo el análisis es continuo y se trata de evaluar propiedades locales (no aisladas). Esta evaluación se realiza para facilitar una propagación de los valores estimados a los elementos cercanos tales como píxeles en una imagen, vóxeles en una representación o trayectorias realizadas por elementos significativos aislados en una mini-secuencia de vídeo.

Por ello, los detectores continuos afectan sobre todo a propiedades locales de regiones representadas por funciones sobre abiertos. Desde el punto de vista topológico desarrollado en el primer capítulo son 0-cociclos. Ejemplos típicos están dados por la función de intensidad en la escala de grises I_g , la intensidad de cada uno de los canales utilizados para imágenes en color, la profundidad d o la altura h para representaciones tridimensionales, etc.

Funciones de probabilidad.

Las funciones de probabilidad se pueden describir de forma discreta o “continua”. Ejemplos típicos de las primeras se describen en el marco bayesiano como una realimentación entre las probabilidades a priori y a posteriori (Teorema de Bayes) asociadas a una distribución discreta de datos.

De forma complementaria, las funciones de densidad de probabilidad (*pdf*) juegan un papel fundamental para especificar las relaciones entre nubes de datos densas y las asociadas a diferentes tipos de muestreo que aparecen en relación con la toma de datos. Para organizar a bajo nivel la información relativa a dichas nubes es frecuente utilizar funciones de energía R o de entropía H (medida del desorden) para gestionar la información (Shannon) asociada a las configuración de datos.

Detectores mixtos y Optimización.

En la práctica, para analizar cualquier tipo de soporte digital (imágenes, secuencias de vídeo, representaciones volumétricas estáticas o en movimiento) se utilizan todos los tipos de detectores que se han comentado más arriba. Se puede trabajar con ellos por separado o bien de forma conjunta. La gestión separada se interpreta como una “aplicación” de detección (con componentes discretas, continuas, probabilistas) que proporciona los inputs “instantáneos” para los descriptores que se abordan en la subsección siguiente.

Cuando se combinan varios detectores en uno solo, es necesario especificar el “peso relativo” de cada componente, identificando la “importancia relativa” (coordenadas afines generalizadas) de cada componente en relación con una detección correcta. La selección de las “funciones de importancia” puede ser realizada por el diseñador del experimento (enfoque supervisado) o bien emerger a partir del análisis de datos (enfoque no-supervisado); el muestreo condicionado a funciones de importancia es un tópico que empieza a desarrollarse a finales de los noventa en diferentes subáreas de Visión Computacional. El problema de optimización multiobjetivo (asociada a las diferentes componentes) subyacente va más lejos del alcance de este trabajo.

3.2.2. Descriptores.

Los *descriptores* correspondan a algún tipo de campos (escalares, vectoriales, tensoriales) definidos localmente sobre un abierto U del espacio base B de una fibración $\xi = (E, \pi, B, F)$ localmente trivial, en la que F puede ser \mathbb{Z}_2 , \mathbb{Z} , \mathbb{Q} o cualquier dominio de \mathbb{R} . El carácter localmente trivial de la fibración ξ sobre U se expresa como $\pi^{-1}(U) \simeq U \times F$. Esta propiedad permite extender (usando secciones locales) los datos obtenidos sobre el abierto U del espacio base B (un espacio muestral en las aplicaciones prácticas) al espacio total E de ξ .

Los *campos escalares* sobre $U \subset B$ son esencialmente lo mismo que los detectores, pero su elevación a E (sólo definida localmente, es decir, sobre $\pi^{-1}(U)$) extienden la noción de detector. Esta extensión es el análogo a la consideración de la energía potencial de un sistema como función sobre un espacio de configuraciones de un sistema o, con más generalidad, sobre el espacio de las fases cuando se considera su elevación (via la sección nula) al espacio total del fibrado (co)tangente para dar la componente escalar del funcional de energía total de un sistema.

Campos vectoriales como descriptores.

Los *campos vectoriales* son la extensión natural de una colección finita de campos escalares en su forma funcional (es decir, sin evaluar aún en cada punto). Cada campo X_i se expresa localmente mediante un operador lineal (sobre el espacio de funciones) dado por $\sum_{j=1}^m f_{ij}(x_1, \dots, x_m) \partial/\partial x_j$ que a cada función $g : U \rightarrow \mathbb{R}$ le asocia $X(g)$. La condición de linealidad para cualquier campo X se expresa como

$$X(af + bg) = aX(f) + bX(g) \quad \forall a, b \in \mathbb{R}, \forall f, g \in C^r(U, \mathbb{R})$$

Cada campo es integrable de forma individual, aunque el conjunto finito de k campos (distribución vectorial) debe verificar condiciones adicionales para su integrabilidad (Teorema de Frobenius). La solución para cada campo X_i se puede interpretar como una “trayectoria” γ_i para $1 \leq j \leq k$. En la práctica, la estimación se lleva a cabo en términos de diferencias finitas que pueden afectar a coordenadas espaciales (para deformaciones, p.e.), temporales (para desplazamientos, p.e.) o espacio-temporales (para deformaciones que tienen lugar a lo largo del tiempo).

Los campos vectoriales permiten representar fenómenos complejos de acoplamiento que se expresan usando relaciones entre las funciones $f_{ij}(x_1, \dots, x_m)$ dependientes de varias coordenadas o parámetros para $j = 1, \dots, p$. En el caso diferenciable se expresan como $\sum_j f_j(x_1, \dots, x_n) \partial/\partial x_j$; en el caso discreto, hay que reemplazar las derivadas parciales por diferencias finitas. En las aplicaciones prácticas, las f_i son funciones de probabilidad y las distribuciones \mathcal{D} de campos son distribuciones de probabilidad (eventualmente generadas por campos de Markov para incorporar la incertidumbre). En este trabajo, para simplificar suponemos que los modelos estadísticos son “paramétricos”, es decir, disponemos de un sistema de variables o coordenadas locales (eventualmente variables aleatorias) con respecto a los cuales representamos los campos.

Como conclusión, el efecto conjunto de estos campos se puede interpretar como un “desplazamiento simultáneo” de “cantidades” o bien como una “deformación” que permite relacionar modelos teóricos (enfoque top-down) con los emergentes del pegado de observaciones locales (enfoque bottom-up). Por ello, su modelado es crucial para cuestiones de Reconocimiento tanto de objetos como de procesos, incluyendo tareas o acciones realizadas por dispositivos inteligentes (agentes).

Formas como descriptores.

En ocasiones, es difícil parametrizar un fenómeno incluso desde el punto de vista local y tan sólo podemos inferir propiedades a partir de la evaluación de ciertos funcionales inicialmente de tipo (multi)lineal. Los funcionales lineales más simples en un espacio cartesiano están dados por formas lineales $\sum_{i=1}^m a_i x_i$ cuya anulación representa un hiperplano H_i ; la no anulación permite representar “de qué lado está” el punto con respecto a la forma que se evalúa en dicho punto.

Esta representación con coeficientes constantes se extiende de forma natural al caso de coeficientes variables dadas por funciones que toman diferentes valores dependiendo del punto base. En este caso, en cada abierto coordenado U de una variedad M se tiene una expresión local $\sum_j f_{jk}(x_1, \dots, x_m) dx_j$ a la que se llama 1-forma diferencial ω_k , donde dx_j se interpreta como una forma lineal sobre el espacio tangente $T_p M$ a una variedad M en cada punto $p \in M$. En particular, $dx_j(\partial/\partial x_i) = \delta_{ij}$ es la delta de Kronecker. Como consecuencia, $\omega_k(X_j) = \sum_{k=1}^m f_{jk} g_{jk}$. La condición de linealidad para cualquier forma X se expresa como

$$\omega(aX_f + bX_g) = a\omega(X_f) + b\omega(X_g) \quad \forall a, b \in \mathbb{R}, \forall X_f, X_g \in \Gamma(U, TU),$$

donde $\Gamma(U, TU)$ denota el espacio de las secciones del fibrado tangente restringido a U (campos vectoriales locales). Un ejemplo típico está dado por la colección de valores numéricos

asociados a una representación tridimensional del color que varía con respecto a las condiciones de iluminación, por ejemplo. Otros ejemplos más sofisticados corresponden a la evaluación simultánea de los valores numéricos que presentan las distribuciones de velocidades para vehículos móviles en una escena de tráfico, p.e.

Formalmente, las 1-formas ω_j son las duales de los campos vectoriales. Por eso a las formas diferenciales también se les llama covectores. Aunque sean expresiones funcionales, en cada punto evalúan numéricamente el comportamiento de los campos vectoriales para identificar posible convergencia o no hacia la función o la distribución objetivo, de acuerdo con los principios básicos de optimización. En el caso diferenciable se expresan como $\sum_j f_j(x_1, \dots, x_n) dx_j$; en el caso discreto, hay que reemplazar las diferenciales por diferencias finitas.

Productos formales para el agrupamiento.

Los *campos tensoriales* están dados por productos formales de r campos vectoriales y de s formas diferenciales; en otras palabras, al mismo tiempo que se describe la evolución espacio-temporal de algunos datos, se evalúan numéricamente otros. Al objeto obtenido se le llama un campo tensorial de tipo (r, s) sobre el espacio muestral X y se le denota mediante $t^{s,r} \in T^{s,r}(X)$.

El conjunto de tensores de cualquier tipo tiene estructura de álgebra tensorial a la que se denota mediante $T^{*,*}(X) := \sum_{s,r} T^{s,r}(X)$. La evolución $t^{s,r}$ de un tensor en un espacio-tiempo se puede describir en términos de la derivada de Lie $\mathcal{L}_X t^{s,r}$ a lo largo de la curva integral del campo vectorial X . Una representación más intrínseca está dada por la noción de conexión que representa la “diferenciación covariante” definida sobre el álgebra tensorial $T^{*,*}(X)$. Además, se tienen operadores de contracción y de expansión $T^{*,*}(X)$ que permiten:

- Modificar la distribución de $(r + s)$ dentro del par (s, r) ; así, p.e. es posible pasar de una métrica dada por un tensor de tipo $(2, 0)$, a una forma bilineal $(1, 1)$, o a una métrica sobre el espacio dual dada por $(0, 2)$.
- Aumentar o disminuir el grado total $r + s$; un ejemplo clásico está dado por la transformación del tensor de Riemann de grado total 4 al tensor de Ricci de grado total 2, de uso común en Relatividad General o, más recientemente, en aplicaciones a Informática Gráfica en relación con la Geometría Conforme.

Algunos ejemplos en Ingeniería.

La utilización de tensores o, con más generalidad, campos tensoriales, permite formalizar una gran cantidad de propiedades observadas en relación con posibles eventos tanto estáticos o dinámicos. Los tensores son de uso común en Física e Ingeniería desde finales del s.XIX. Los tensores más frecuentes en Ingeniería aparecen en relación con propiedades de la Mecánica de Medios Continuos con la Dinámica de Fluidos como tópico central.

También en relación con la Teoría de la Elasticidad para objetos deformables (tensores de deformación, de restricciones, de contacto, viscosidad, etc). En este último contexto aparecen “singularidades” para los tensores correspondientes a fenómenos de dislocación de gran interés para aplicaciones relacionadas con la mecánica de materiales o, más recientemente, la Robótica.

En Visión Computacional aparecen tensores en relación con el procesamiento y análisis de imágenes biomédicas (para modelos no-lineales de propagación sobre medios anisótropos, p.e.). Es el elemento clave para la Reconstrucción 3D a partir de dos vistas (matrices fundamental y esencial), tres vistas (tensor trilineal) o múltiples vistas (tensor multilineal). Asimismo, el tensor de estructura asociado al gradiente de la intensidad en la escala de grises proporciona el soporte para un enfoque intrínseco vinculado a la estimación del movimiento.

Un reto actual es la estimación de las características cinemáticas para vehículos en movimiento, el seguimiento simultáneo y la coordinación supervisada en relación con la navegación automática de flotas de vehículos. Además de la distribución de velocidades o aceleraciones (cinemática propiamente dicha), se evalúan la variación de las características radiométricas de los vehículos o de la escena (incluyendo color, textura, iluminación, reflectancia, etc) que se representan con “formas” o aplicaciones lineales definidas sobre el espacio de la Radiometría.

3.2.3. Clasificadores.

El problema de la clasificación es crucial para la automatización en los procesos de Reconocimiento automático. Afecta al tipo de objetos a clasificar y la construcción explícita de las clases módulo algún tipo de transformaciones; esta cuestión se aborda en el primer apartado.

Clasificando objetos y transformaciones.

Los objetos pueden ser continuos (variedades topológicas, p.e.) o discretos (nubes de puntos, p.e.). Pueden tener un carácter denso o discreto. Se les puede asociar una C^r -estructura con los casos $r = 1$ (PL-estructuras) y $r = \infty$ (PS-estructuras) como casos extremos.

La Topología Algebraica y Diferencial proporciona herramientas para identificar invariantes y mostrar de forma explícita posibles deformaciones a visualizar en estructuras superpuestas (fibrados o fibraciones, p.e.). Para simplificar nos restringimos al caso de fibrados vectoriales.

Las transformaciones permiten agrupar el número de tipos que pueden aparecer en una lista más reducida. Dependiendo del carácter (discreto vs continuo) de los objetos se tienen diferentes tipos de transformaciones algebraicas vs topológicas que afectan a las configuraciones de objetos. En el caso clásico dichas transformaciones generan un grupo finito (permutaciones, reflexiones, p.e.), de dimensión finita (grupos clásicos de Lie, p.e.) o de dimensión infinita (subgrupos del grupo de los homeomorfismos).

De cara a las aplicaciones, el problema más complicado consiste en dar criterios efectivos para estimar las transformaciones. Una estrategia frecuente consiste en linealizar la acción del grupo, convirtiendo dicha acción algebraica en una infinitesimal (álgebra de Lie), para resolver el problema.

Clasificando estructuras superpuestas.

La caracterización de detectores y descriptores dada más arriba como campos escalares o (co)vectoriales en un fibrado, sugiere desarrollar el problema de clasificación en términos de clases de fibrados vectoriales ξ sobre una variedad M . Esta cuestión afecta al grupo de transformaciones que actúan sobre la fibra, al que se llama grupo estructural G del fibrado ξ .

En el marco diferenciable la clasificación de los fibrados vectoriales (módulo isomorfismo) sobre una variedad M es equivalente a la clasificación (módulo conjugación) de los sistemas de funciones de transición $g_{ij} : U_i \cap U_j \rightarrow G$ para cualquier par $i, j \in I$ correspondiente a dos abiertos U_i, U_j con intersección no vacía; inicialmente $G = GL(m, \mathbb{R})$, pero en realidad puede ser cualquier grupo (incluyendo grupos discretos o grupos infinito-dimensionales para deformaciones arbitrarias).

Por ello, las funciones de transición permiten relacionar mediante la acción de un elemento de un grupo G datos comunes de la misma observación desde dos localizaciones diferentes o bajo condiciones ambientales diferentes. Este enfoque es compatible con el enfoque probabilista (distribuciones de probabilidad sobre grupos).

Por ello, en la reformulación de Sistemas Expertos que llevamos a cabo en la sección siguiente, los *clasificadores* correspondan a las clases (módulo conjugación) de los grupos de transformaciones sobre las fibras generadas por los descriptores. En otras palabras, si partimos de la extensión de un sistema de descriptores s_i al espacio total E del fibrado (como sección local $s_i : U \rightarrow E$ del fibrado), la acción del grupo G sobre el subespacio $\langle s_1, \dots, s_k \rangle$ generado por los descriptores permite relacionar los resultados proporcionados por diferentes sistemas de detección y “empaquetarlos” en clases módulo la acción de conjugación de un grupo G ó al menos de un álgebra de Lie $\mathfrak{g} := T_e G$ para el caso vectorial ⁸.

Además y desde un punto de vista global, la clase algebraica resultante para las funciones de transición es equivalente a la clase (módulo isomorfismos) de la fibración, lo cual permite comparar diferentes estrategias de clasificación. Cuando los fibrados resultantes no son equivalentes, la comparación se lleva a cabo inicialmente en términos de homomorfismos entre fibrados (ver más arriba). A cada homomorfismo $\varphi : \xi \rightarrow \eta$ entre fibrados se le puede asociar el núcleo $Ker(\varphi)$ y el conúcleo $Coker(\varphi)$. Estos objetos “codifican” las relaciones

⁸La aplicación exponencial proporciona un difeomorfismo local $\mathfrak{g} \rightarrow G$

de “inclusión parcial” o las “diferencias” sobre un espacio de referencia común (“proyección no-lineal” en el sentido utilizado por Kohonen en SOM, p.e.)⁹.

Fibrados principales para Reconocimiento.

De la misma forma que ocurre en la Teoría Clásica de Fibrados, la estructura de la fibra como espacio vectorial no es imprescindible. En particular, la fibra puede ser un grupo en lugar de un espacio vectorial, lo cual da lugar a la noción de fibrado principal $\mathcal{P} = (P, \pi, B, G)$ sobre un espacio base B en el que se utilizan detectores, descriptores o clasificadores para resolver problemas de Reconocimiento (entrenamiento y aprendizaje, p.e.).

En este caso, se supone que la transformación entre los PL- ó PS-modelos asociados a cada par de vistas próximas está dada por un elemento de un grupo G correspondiente a un grupo clásico o bien a (un subgrupo del) grupo de los difeomorfismos del espacio ambiente.

Así p.e., las representaciones de una escena se definen como observables desde diferentes localizaciones de la cámara. En el caso de modelos de perspectiva para la escena, se tienen diferentes tipos de transformaciones lineales (euclídeas, afines, proyectivas) dependiendo de las restricciones sobre la cámara y su localización relativa, en función del marco estructural elegido para representar la escena. En este caso, la acción del grupo correspondiente sobre una representación permite dotar de estructura natural de fibrado principal, anticipando posibles movimientos de cámara en función de las condiciones iniciales para la trayectoria inicialmente prevista. En particular, las clases de equivalencia (modulo la acción de conjugación) que representan los clasificadores se construyen como funcionales elegidos por su capacidad de discriminación.

Reconocimiento en imágenes segmentadas.

La *segmentación de una imagen* es la descomposición en unión disjunta de objetos topológicos b_β definidos como unión de regiones $r_{\beta i}$ correspondientes a primitivas geométricas adyacentes. En presencia de movimiento real o aparente (asociado al movimiento real de la cámara), el problema es bastante más difícil. En este caso, se adopta una estrategia tosca-fina (refinamientos sucesivos) procediendo a una localización relativa dentro de imágenes muestreadas, acotando la región (mediante cuadrilátero, p.e.) y llevando a cabo un seguimiento de los cuadriláteros.

El seguimiento de un punto aislado está resuelto desde los años setenta en términos de variantes de filtros Kalman. Sin embargo, el seguimiento de configuraciones eventualmente deformables de configuraciones de puntos es un problema aún abierto. Una simplificación de este problema es el seguimiento de triángulos o cuadriláteros de aspecto cambiante.

Un modelo matemático pseudo-determinista está dado por el seguimiento de una distribución \mathcal{D} generada por tantos campos vectoriales (eventualmente de Markov) como vértices tenga el polígono asociado como envolvente. Si trabajamos en el espacio de triángulos o cuadriláteros “generalizados” (que admiten degeneraciones a polígonos con número menor de lados), el problema del seguimiento simultáneo se reduce a seguir un punto en el espacio generalizado correspondiente.

Esta descripción se extiende de forma natural a objetos deformables (diferentes tipos de agentes) o al caso de objetos móviles reemplazando las características geométricas iniciales por características topológicas o cinemáticas (incluyendo tasas de variación de primer y segundo orden para las coordenadas locales o las correspondientes funciones asociadas).

Las descomposiciones cuboidales descritas en el capítulo 1 de este trabajo proporcionan el marco natural para una gestión jerarquizada de la información global (resp. local) asociada a una escena estática. Análogamente, las descomposiciones simpliciales corresponden a diferentes agentes locales (personas, animales, etc) que operan en la escena. De este modo, se introduce un tratamiento por “capas” que facilita los procesos de reconocimiento tanto en el caso de imágenes estáticas como móviles.

⁹Una descripción más formal y rigurosa se presenta en el capítulo siguiente, incluyendo consideraciones sobre su naturaleza cambiante (no tienen por qué ser fibrados vectoriales, p.e.)

3.2.4. Una posible extensión a reconocimiento en vídeo.

El problema del Reconocimiento para objetos estáticos se ha comentado en la subsección anterior; empezamos recordando algunos hechos básicos en el primer apartado. Aunque aún quedan muchísimos problemas abiertos está razonablemente bien entendido. De una forma afecta en el fondo a un diseño e implementación eficientes de Sistemas Expertos. Actualmente, hay una casuística gigantesca, pero cabe esperar que la adopción de un lenguaje más potente como el presentado en las subsecciones anteriores para detectores, descriptores, clasificadores, permita avanzar en la reorganización de materiales.

La dificultad crece de forma exponencial para secuencias de vídeo. Además, el crecimiento del volumen de contenidos (en su mayor parte no etiquetados) para secuencias de vídeo en la red hace realmente difícil la resolución efectiva de los problemas de Reconocimiento. Basta pensar que en 2016 un humano para necesitaría 85 años de su vida para etiquetar los contenidos de las secuencias de vídeo que se suben un día a la red (sin contar los correspondientes a las redes sociales). Este simple hecho muestra el interés y la necesidad de avanzar en herramientas para la detección, descripción, etiquetado, extracción y clasificación de contenidos, en secuencias de vídeo digital.

Sobre el problema de Reconocimiento.

El problema más difícil en Visión Computacional es el desarrollo de estrategias lo más generales posibles al reconocimiento del contenido en imágenes digitales. Los Sistemas Expertos facilitan criterios para entrenamiento (supervisado o no), etiquetado semi-automático, indexación y clasificación final. Para ello, utilizan diferentes tipos de detectores, descriptores y clasificadores, con características “deterministas” o “probabilistas”. El intercambio de información entre ambos enfoques es fundamental para avanzar en modelos más estructurados.

Existen dificultades en todos los terrenos, pero uno de los mayores es la “ausencia” de una jerarquía lo más universal posible para los Sistemas Expertos a implementar. Las jerarquías deben afectar no sólo a los objetos, sino a las tareas realizadas por dichos objetos o los procesos en los que se ven involucrados. Estas cuestiones afectan sobre todo al *contexto*, es decir, a la Semántica del contenido digital de la secuencia de vídeo.

Una digresión sobre Sistemas Expertos.

Básicamente, un *Sistema Experto* es un conjunto de reglas lógicas; sin embargo, hay diferentes tipos de lógica:

- *Lógica de clases* correspondiente a la comparación (estricta o relajada) con un patrón bien conocido. Se resume en la sentencia aparentemente tautológica $A = A$, que no es tan trivial como parece, pues cuando hay pequeñas diferencias el sistema debe facilitar el modelo más cercano. Un ejemplo típico está dado por la compleción de una palabra en la pantalla de un dispositivo móvil, p.e.
- *Lógica proposicional* en términos del sistema de tablas de verdad para los razonamientos más simples en la lógica de primer y de segundo orden. Un ejemplo típico está dado por *Si A, entonces B* que expresamos simbólicamente como $A \Rightarrow B$, de uso común en la “demostración automática de teoremas”.
- *Lógica descriptiva* en la que se utilizan criterios de tipo probabilista (lógica difusa, p.e.) para la toma de decisiones. Un ejemplo típico está dado por *Si A entonces B con una probabilidad de un 40 % o bien B con una probabilidad de un 60 %*. La distribución de probabilidades puede verse alterada por la acumulación de información que se formaliza usando criterios de máxima verosimilitud para esquemas de razonamiento de tipo bayesiano.

En las aplicaciones prácticas y, por consiguiente, en su implementación computacional, se pueden presentar varios de estos tipos de lógica de forma simultánea. En este caso, es necesario, asignar “pesos diferentes” a cada componente que pueden verse modificados en función de eventos. El desarrollo de *Sistemas Expertos Mixtos* para la navegación automática de vehículos en escenas de tráfico urbano.

Segmentación de vídeo.

En una primera aproximación, un vídeo digital es una colección finita de imágenes digitales. Por ello, el procesamiento y análisis de una secuencia de vídeo se puede llevar a cabo inicialmente de forma secuencial; es decir, analizando cada frame por separado. En fases más avanzadas, aparecen fenómenos de entrelazamiento, persistencia de contenidos fácilmente detectables que se pueden “empaquetar” y fenómenos de propagación que simplifican el análisis del contenido.

En el caso estático (imágenes aisladas procedentes eventualmente de una secuencia de vídeo) es importante minimizar efectos de sobre-determinación, lo cual conlleva la resolución de problemas de optimización geométrica. Este análisis se extiende de forma natural a optimización cinemática para imágenes en movimiento utilizando el flujo de imagen asociado a las diferentes regiones móviles con movimiento inercial (característico de la cámara) o bien propio (de los objetos en la escena). Los casos más “simples” corresponden a cámara fija y objetos móviles (CFOM) o bien cámara móvil y objetos fijos (CMOF) en la escena.

Habitualmente, se supone que es imposible describir *todos* los elementos que pueden aparecer en imágenes digitales. Por ello, una estrategia básica consiste en tratar de detectar si existen objetos correspondientes a una lista creciente de patrones; actualmente, dependiendo del sistema, la lista es de varios cientos o unos pocos miles de objetos. Un objetivo a medio plazo consiste en desarrollar criterios de Optimización (Geométrica y Cinemática) asociados a restricciones básicas para los tipos de operadores más simples utilizados en Reconocimiento.

Un problema bastante más difícil es la extracción del contenido en mini-secuencias de vídeo. Esta cuestión afecta no sólo al tratamiento individualizado del contenido extraíble en cada frame, sino al seguimiento de algunos de ellos a lo largo de la secuencia, tratando de identificar los procesos que les afectan, las acciones que llevan a cabo y las interacciones con otros agentes que se puedan presentar. Para abordar estos problemas es imprescindible desarrollar técnicas bastante más avanzadas que se esbozan en las dos secciones siguientes de este capítulo.

3.3. Desde SOM a Deep Learning.

En esta sección se presentan algunas de las redes neuronales más relevantes para el problema del reconocimiento de la forma cuyos fundamentos teóricos se han presentado más arriba.

3.3.1. Redes neuronales convolucionales.

En 1998, Yann LeCun y sus colaboradores desarrollaron un reconocedor muy eficiente para dígitos manuscritos llamados LeNet. Es un modelo de capas organizado según un modelo de realimentación sobre una red con muchas capas ocultas. Cada capa contiene mapas de unidades replicadas, dispone de una puesta en común para las salidas de unidades replicadas, una amplia red que puede enfrentarse a varios “caracteres” de forma simultánea aunque se solapan y una forma inteligente de formación de un sistema completo, no sólo un reconocedor. Más tarde se formalizó bajo el nombre de *Redes Neuronales Convolucionales* (CNNs).

Las CNNs son muy diferentes de la mayoría de las demás redes. Siguen principios muy similares a los utilizados en procesamiento de imágenes digitales, pero puede también usarse para otros tipos de entrada como audio. Un caso típico de uso para un entrenamiento supervisado de CNNs consiste en introducir imágenes (inicialmente etiquetadas de forma manual) para que la red clasifique los datos.

Una estrategia básica por capa.

Las CNNs empiezan con un escaneo de entrada que no está diseñada para analizar todos los datos del entrenamiento a la vez. Por ejemplo, para introducir una imagen de 100×100 píxeles, no es necesaria una capa con 10.000 nodos. En lugar de ello, se crea una máscara para la capa de entrada de tamaño 10×10 que es alimentada por los primeros 10×10 píxeles de la imagen. A continuación, se traslada la máscara formada por los 10×10 píxeles siguientes según un procedimiento de barrido por bloques hasta completar la primera banda. Una vez terminada, se comienza por la siguiente.

De este modo, la “máscara pesada” (habitualmente cuadrada) proporciona el “núcleo” del operador que funciona como un mecanismo de activación o de modificación de la señal original. Dependiendo de los valores numéricos que aparecen en la máscara se pueden conseguir efectos de realzado o de suavizado, o bien una combinación de ambos.

Conectando capas sucesivas.

La entrada de datos se alimenta a través de capas convolucionales sucesivas en lugar de sobre una capa aislada, donde no todos los nodos están conectados a todos los nodos. Cada nodo sólo está conectado inicialmente a células vecinas cercanas. De este modo, es posible desarrollar modelos de activación locales con objeto de evaluar posibles correlaciones que tengan en cuenta la estructura planar (en fases más avanzadas, espacial para redes 3D ó espacio-temporal para redes 2D+1d como las vinculadas a vídeo digital, p.e.).

Las capas convolucionales también tienden a disminuir el tamaño de la máscara con la profundidad; para ello, se utilizan factores fácilmente divisibles de la entrada inicial. Este procedimiento recibe el nombre de “abstracción piramidal” y es de uso común en Visión Computacional; dos casos típicos están relacionados con pirámides gaussianas o pirámides laplacianas para procesos de suavizado o de realzado.

Agrupamiento y muestreo.

Además de estas capas convolucionales, a menudo también cuentan con procedimientos de agrupamiento o de muestreo por capas. La capa de muestreo filtra los datos utilizando alguna función definida sobre la máscara para procesos de agrupamiento; un ejemplo típico utiliza criterios de maximización; así, p.e. para una máscara de tamaño 2×2 píxeles, se toma el píxel que presenta un mayor valor para la función de intensidad que se está considerando.

Otras funciones de uso común que no dan lugar a las discontinuidades asociadas a tomar el máximo de los valores que aparecen en cada celda tras la aplicación de la máscara están dadas por la función $\tanh(x)$ (o más bien su módulo) y por la sigmoide $\sigma(x) := (1 + e^{-x})^{-1}$.

Invariancia.

Es conveniente que el entrenamiento de una CNN sea invariante con respecto a las transformaciones geométricas ó radiométricas (condiciones de iluminación, p.e.). Las transformaciones geométricas más simples son las del grupo de semejanza: traslaciones, rotaciones, escala. Para ello, se introducen “vectores de hechos” asociados a descriptores de acuerdo con las ideas básicas presentadas en la primera sección o su reformulación (como secciones de fibrados) en la segunda sección.

Para formalizar esta idea, se utilizan transformaciones entre referencias dentro del espacio de “hechos” que deben reformularse en términos de transformaciones a describir sobre la capa. Un ejemplo típico muy simple está asociado a las transformaciones tipo compás para detectar la dirección predominante (mayor crecimiento de la derivada direccional discreta) de un segmento. Este procedimiento se puede aplicar asimismo en relación con los histogramas de gradientes orientados (HOG) de interés para estimar las direcciones predominantes de movimiento para cada una de las regiones móviles (con movimiento real o aparente) en una secuencia de vídeo.

3.3.2. Incorporando la topología a las CNN.

Escalas y formas.

La selección del tamaño de la máscara para la convolución condiciona los resultados a obtener. Frecuentemente, se adopta una estrategia de tipo piramidal partiendo de una plantilla de elevado tamaño que se subdivide progresivamente dependiendo del hallazgo o no de la información buscada o bien del grado de precisión requerido.

Dos *estrategias típicas* procedentes del enfoque piramidal en Visión Computacional consideran pirámides gaussianas o laplacianas para el suavizado o el realzado de las posibles discontinuidades que puedan aparecer. Así, p.e. si se parte de una plantilla de tamaño 16×16 , la subdivisión en 4 partes da lugar a 4 plantillas de tamaño 4×4 que a su vez se subdividen en 4 plantillas de tamaño 2×2 . De este modo, se obtiene un mapa con tres niveles de profundidad en el que la mayor parte de las células pueden ser etiquetadas como vacías E (empty)

debido a la no-presencia de los elementos buscados. De hecho, el proceso de subdivisión sólo se continúa para las células marcadas como llenas F (full).

Las sub-células llenas pueden ser adyacentes o bien estar desconectadas entre sí (a distancia superior a un umbral).

Revisitando relaciones entre componentes.

La traslación de la plantilla asociada a una red convolucional permite detectar elementos en cada super-célula a enlazar con los correspondientes a otras supercélulas. Una vez detectados y enlazados, se enlazan para detectar regiones o hechos significativos de acuerdo con características globales o locales de la forma a localizar y estimar.

El enlazado de regiones con características similares se realiza mediante aristas de un grafo de acuerdo con criterios de adyacencia. Estas aristas representan posible interacciones entre los agentes representados por las regiones con características similares. Como la localización de los diferentes agentes puede cambiar (típico en escenas de tráfico, p.e.) se tiene una representación dada por multicaminos (ver **Capítulo 1**).

Las componentes pueden presentar características similares y pertenecer al mismo objeto (discontinuidades debidas a oclusiones parciales, p.e.) o, por el contrario, presentar características diferentes en términos de propiedades geométricas (profundidad ó aspecto, pe.) o radiométricas (intensidad en grises, color, reflectancia, p.e.).

En cualquier caso, pueden estar separadas por obstáculos (fijos o móviles) que dan lugar a que algunos caminos cerrados no sean contractibles a un punto. Este hecho da lugar a representaciones topológicas equivalentes a la suma conexas de $\#^g \mathbb{T}^2$ de g toros $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$ utilizada en cuestiones de Planificación de Movimientos para Robots descritas por Latombe (1991).

Operadores y restricciones.

De acuerdo con la descripción de los Sistemas Multiagente presentada más arriba, los aspectos topológicos de MAS a formalizar desde el punto de vista topológico conciernen a

- *Objetivos* del sistema para llevar a cabo las tareas. Se pueden representar como dados por una variedad topológica que representa la anulación de las funciones objetivo.
- *Planes* asociados a la utilización de una biblioteca de planes para gestionar eventos y alcanzar los objetivos. Esta biblioteca representa posibles métodos que se visualizan como trayectorias. Cada plan o estrategia se visualiza como un camino. La competencia entre caminos (de cara a posible optimización) se visualiza en un espacio de multicaminos introducido en el primer capítulo.
- *Eventos* que aparecen al procesar y actualizar las “creencias” de acuerdo con las acciones a llevar a cabo por los diferentes agents. Los eventos se describen como “discontinuidades” del sistema o de su evolución temporal.

En presencia de k agentes a_1, \dots, a_k operando de forma independiente en el espacio, podemos representar dicha acción como un multicamino correspondiente a colecciones finitas de caminos (eventualmente cerrados). Su gestión utiliza un espacio topológico X con k puntos base b_1, \dots, b_k (representing the reduction of agents to points). Los invariantes topológicos se describen por el grupo fundamental $\pi_1(X; b_1, \dots, b_k)$, que es la extensión obvia del grupo fundamental ordinario $\pi_1(X; x)$ de lazos correspondientes a las clases de homotopía de caminos cerrados. La primera dificultad a resolver está relacionada con la descripción de todas las posibles configuraciones de (el complementario de) k puntos móviles (agentes) $a_1(t), \dots, a_k(t)$. La dificultad procede de que algunos de ellos pueden presentar comportamientos correlacionados o incluso solaparse.

3.3.3. Memoria y aprendizaje.

Buena parte de los mamíferos superiores tienen un memoria a corto y a largo plazo. La primera tiene un carácter efímero (duración de pocos segundos), mientras que la segunda presenta caracteres “irreversibles” en el horizonte temporal. Para consolidar el carácter

irreversible se desarrollan mecanismos de reforzamiento. Además, existen memorias de tipo asociativo que pueden funcionar indistintamente para las memorias a corto y largo plazo.

Las Redes Neuronales Recurrentes (RNN) permiten incorporar procesos que tienen lugar a lo largo del tiempo, mediante una realimentación asociada a mecanismos de propagación hacia adelante y hacia atrás. Por ello, se están convirtiendo en una de las herramientas más potentes para el modelado espacio-temporal. Desde el punto de vista computacional, ello requiere una gestión más eficiente de diferentes tipos de memoria que puedan proporcionar mecanismos de reforzamiento, no sólo sobre elementos aislados (modelables como trayectorias eventualmente discretas), sino sobre cómo evoluciona su distribución espacial a lo largo del tiempo.

Por falta de espacio y por coherencia con el resto de materiales, en este trabajo no se desarrollan las RNN (Recurrent Neural Networks). Sin embargo, a la vista de su utilidad en procesos temporales es conveniente tenerlas presente como una posible extensión de los contenidos presentados en este TFG.

Empezamos recordando algunos aspectos básicos de las memorias a corto y largo plazo.

Memoria a corto plazo.

Requiere un número bajo de items correspondientes a eventos que, habitualmente, no están correlacionados. Por ello, a menudo siguen comportamientos reactivos que afectan sólo a partes aisladas de la red neuronal. Un típico ejemplo está dado por la concatenación de letras para formar una palabra que no tiene por qué tener un significado, como p.e. una clave de acceso temporal a un sistema.

Es de utilidad para un etiquetado, pero una vez realizado el proceso no se retiene en memoria. Se puede modelar como un fenómeno disipativo o como un fenómeno en el que no hay realimentación de ningún tipo.

Memoria a largo plazo.

Requiere la formación de una representación (procedente del procesamiento y análisis como la información acústica o visual p.e.), *almacenamiento* con restricciones estrictas (de tipo métrico, p.e.) o más relajadas (de tipo afín o asociada a deformaciones más generales, p.e.) y reutilización (en función del contexto).

Mecanismos de aprendizaje.

Algunos de los más frecuentes están dados por reforzamiento o, a un nivel superior (de tipo semántico) por memorias asociativas. Para verificar si el proceso de aprendizaje está dando los resultados esperados, es necesario que el sujeto del aprendizaje responda de forma positiva con respecto a los estímulos favorables e ignore los desfavorables.

Una reformulación en RNN.

Hochreiter y Schmidhuber (1997) resolvieron el problema de conseguir un RNN (Recurrent Neural Network) para memoria a largo plazo mediante la construcción de lo que se conoce como *Long Short-Term Memory* (LSTMs). Las redes LSTM tratan de combatir el problema de anulación (o de explosión, como inversa) del gradiente mediante la introducción de “puertas” y una célula de memoria definidos explícitamente.

La célula de memoria almacena los valores anteriores y se aferra al actual salvo que una “puerta a olvidar” mande a la célula olvidar esos valores. LSTMs también tienen una “puerta de entrada” que añade datos nuevos a la célula y una “puerta de salida” que decide cuándo pasar a lo largo de los vectores de la célula al siguiente estado de oculto. En términos diferenciales, juegan un papel similar a un máximo vs un mínimo local para un funcional o bien un repulsor vs un atractor de uso común en Sistemas Dinámicos.

LSTMs simplemente añade una capa de la célula para asegurarse de que la transferencia de información (mediante iteración) desde un estado oculto al siguiente es “razonablemente alta” (por encima de un umbral u_0). Dicho de otro modo, queremos recordar estados o propiedades de iteraciones anteriores a lo largo del tiempo que sea necesario; las células de LSTMs proporcionan un soporte para que esto suceda. LSTMs han demostrado ser capaces

de aprender secuencias complejas, como escribir al estilo de Shakespeare o componer música primitiva.

3.3.4. ¿Qué red elegir?

A la vista de la enorme diversidad de ANN, la elección de la red más apropiada es un problema no-trivial. Formalmente, una red está compuesta por un número finito de capas, cada uno con un elevado número de nodos encargados del procesamiento. Las neuronas se representan como variables de estado que, a su vez, se expresan como combinaciones lineales “pesadas” de variables input y otras variables output próximas, dependiendo de la función de activación (tipo umbral o sigmoide, p.e.) y del tipo de conexividad, sobre todo.

Una primera taxonomía.

Dos criterios iniciales para clasificar ANN utilizan el tipo de activación o bien el tipo de conexividad. En relación con el *tipo de activación* podemos distinguir entre

- *Redes deterministas* dadas por una función a valores reales monótona creciente, habitualmente, como p.e. el Perceptrón inicial y la red binaria de Hopfield. Las extensiones a Perceptrón multicapa (con la función sigmoide que da lugar a una distribución logística) y las redes continuas de Hopfield (ver más abajo) son algunas de las más utilizadas a lo largo de los ochenta y noventa.
- *Redes estocásticas* con activación asociada a funciones de densidad de probabilidad (pdf). La variante estocástica de estas redes y de redes de Hopfield se etiqueta como redes de Boltzmann y ha sido de uso creciente desde mediados de los noventa.

En relación con el *tipo de conexividad* podemos distinguir entre

- *Redes de retro-alimentación (feedforward)*; En este caso todas las conexiones son unidireccionales tal y como ocurre en el Perceptrón Multicapa, no existiendo conexiones entre las neuronas de la misma capa. Proporcionan un modelo universal [Hor89]¹⁰. Sin embargo, presentan un carácter muy estático en relación con las necesidades de rápida adaptación para un número creciente de aplicaciones
- *Redes Recurrentes Neuronales (RNN)*: En este caso, puede haber conexiones dentro de la misma capa, lo cual da lugar a que el estado cambie a medida que pasa el tiempo. Fueron introducidas asimismo por Hopfield (1982, 1984) y permiten abordar problemas complicados de optimización combinatoria y NP-completos que otras redes clásicas no pueden abordar.

Debido precisamente a su mayor flexibilidad, las Redes Recurrentes (RNN) de unidades no lineales son generalmente muy difíciles de analizar. Puede comportarse de diferentes maneras: converger a un estado estable, oscilar o seguir trayectorias caóticas que no se puede predecir el momento en el futuro.

Redes de Hopfield.

Son significativas tanto para el tipo de activación como el tipo de conexividad. En las redes de Hopfield (HN) cada neurona de una capa está conectada a cualquier otra neurona de la capa siguiente. Esta arquitectura da lugar a una elevadísima complejidad del soporte físico.

Las redes se entrenan de acuerdo con el valor que toman las funciones en las neuronas con respecto al patrón deseado una vez que los pesos se pueden calcular invirtiendo las matrices correspondientes. Una vez calculado el peso, dicho valor permanece inalterado después del entrenamiento. Una vez entrenadas las redes para uno o varios patrones, la red se orienta siempre a uno de los patrones aprendidos porque la red es solamente estable en esos Estados.

Las redes de Hopfield juegan otro papel en relación con la memoria. En lugar de utilizar la red para almacenar recuerdos, se pueden utilizar para construir interpretaciones de entrada sensorial. La entrada está representada por las unidades accesibles, la interpretación está representada por los estados de las unidades ocultas, y el desajuste con respecto a la interpretación está representada por la energía. Esta observación las hace especialmente indicadas en relación con tópicos de la Teoría Clásica de Morse presentada más arriba.

¹⁰K.Hornik, M.Stinchcombe and H.White: “Multilayer Feedforward Networks are Universal Approximators”, *Neural Networks*, 2., 359-366, 1989.

Redes de Boltzmann.

Lamentablemente, la red de Hopfield tiene muchas limitaciones debido a una escasa capacidad. Una red de N unidades de Hopfield sólo puede memorizar patrones $0.15N$ debido a los llamados falsos mínimos en función de su energía. En este contexto se puede aplicar el método del descenso del gradiente más rápido que es fácilmente discretizable.

Sin embargo y como es habitual en este tipo de métodos se puede caer en “falsos mínimos”. La idea es la siguiente: Como la función de energía es continua en el espacio de sus pesos, si dos mínimos locales están muy cercanos, se puede “caer” en uno que es sólo un mínimo local que no corresponde a ninguna muestra ó patrón utilizado en el entrenamiento, al tiempo que se olvida de las dos muestras introducidas para memorizar. Este fenómeno limita significativamente el número de muestras que puede aprender una red de Hopfield.

Este problema se resuelve mediante una perturbación estocástica en las máquinas de Boltzmann, inicialmente motivado por el problema del viajante. Como es bien sabido, se trata de un problema con dificultad NP para el que se han desarrollado diferentes estrategias de tipo heurístico (incorporando hipótesis ad hoc) o bien de tipo estocástico (redes de Boltzmann propiamente dichas).

Las primeras dan lugar a una casuística sin resultados estructurales, mientras que las segundas requieren una potencia de cálculo formidable que hace muy difícil su entrenamiento en tiempo razonable. Cabe esperar que la aplicación de herramientas tipo DeepLearning permita avanzar en este problema, pero ello requiere desarrollar herramientas para el aprendizaje de los “flujos entre cantidades” completamente generales (representados en nuestro caso por tensores). Estas construcciones se abordan en la última sección.

Redes no-lineales.

Los desarrollos presentados en los dos capítulos anteriores sugieren aproximaciones basadas en extensiones no-lineales de los métodos desarrollados para las primeras ANN. Básicamente, hay dos aproximaciones:

- *Redes Neuronales Semi-lineales* que utilizan sumas pesadas de estados y variables input como argumentos de las funciones de activación. Se han utilizado para modelar procesos biológicos sencillos. En el contexto de los fibrados vectoriales presentados en la sección §2,3 corresponderían a funciones elementales definidas sobre el espacio total E de un fibrado vectorial $\xi = (E, \pi, B, F)$.
- *Redes Neuronales de orden superior* que utilizan funciones no-lineales para la activación. Permiten acelerar de forma espectacular la tasa de convergencia dependiendo de las funciones elegidas y proporcionan un enfoque adaptativo a fenómenos cambiantes que otras redes no ofrecen. Por ello, cabe esperar que la adaptación de las herramientas de Deep Learning a esta estrategia proporcione resultados mucho más eficientes.

3.4. Flujos tensoriales en Variedades y TensorFlow en ANN.

El marco de los tensores sobre variedades es bien conocido en Matemáticas, pero su utilización en Inteligencia Artificial es aún muy limitada. En las subsecciones siguientes se esbozan los pasos a dar para facilitar las conexiones entre el marco diferencial del Cálculo Tensorial sobre Variedades y el marco computacional dado por TensorFlow en relación con Redes Neuronales Artificiales.

3.4.1. Cálculo Tensorial en el marco suave.

Recordemos que dado un espacio vectorial V , un tensor de tipo (s, r) sobre V está dado por una forma multilineal, es decir, se expresa en términos de combinaciones lineales de la forma

$$\sum_{I, J} a_{I, J}^I \varphi_I \otimes \mathbf{v}^J = a_{j_1 \dots j_r}^{i_1 \dots i_s} (\varphi_{i_1} \otimes \dots \otimes \varphi_{i_s}) \otimes v^{j_1} \otimes \dots \otimes v^{j_r}$$

producto tensorial de s covectores $\varphi_i : V \rightarrow \mathbb{R}$ para $1 \leq i \leq s$) y r vectores $v_j \in V$ para $1 \leq j \leq r$.

Como el soporte de φ_i es un subespacio de codimensión 1, podemos visualizar dicho tensor (módulo coeficientes) como una colección de s hiperplanos H_i y de r rectas ℓ_j cuyo vector director es v_j . En Geometría, todos esos datos dependen del punto base $p \in \mathbb{M}$ de una variedad \mathbb{M} que, inicialmente, suponemos diferenciable para simplificar.

Tensores sobre una variedad suave.

Si M es una variedad diferenciable, un tensor de tipo (s, r) es una sección local del fibrado vectorial $(\tau_M^*)^{\otimes s} \otimes (\tau_M)^{\otimes r}$ de s copias del fibrado cotangente $\tau_M^* = \Omega_M^1$ y de r copias del fibrado tangente τ_M . Por ello, sus secciones son expresiones de la forma

$$\sum_{I, J} a_J^I dx_I \otimes \frac{\partial}{\partial J} = \sum a_{j_1 \dots j_r}^{i_1 \dots i_s} [dx_{i_1} \otimes \dots \otimes dx_{i_s}] \otimes \left[\frac{\partial}{\partial x^{j_1}} \otimes \dots \otimes \frac{\partial}{\partial x^{j_r}} \right]$$

producto tensorial de s formas diferenciales $dx_{i_k} : T_p(M) \rightarrow \mathbb{R}$ para $1 \leq k \leq s$ (secciones locales del fibrado cotangente) y r campos vectoriales $X_\ell = \partial/\partial x^{j_\ell}$ para $1 \leq \ell \leq r$ (secciones locales del fibrado tangente). De forma intuitiva y de cara a las aplicaciones, estamos evaluando de forma simultánea r trayectorias $\gamma_j(t)$ asociadas a r campos vectoriales X_j y s “cantidades numéricas variables” asociadas a s formas diferenciables.

Un campo vectorial arbitrario X se transforma por multiplicación a la izquierda por la matriz jacobiana $Jac(\psi_\beta \circ \psi_\alpha^{-1})$ del cambio de base $\psi_\beta \circ \psi_\alpha^{-1} : \psi_\alpha(U_\alpha \cap U_\beta) \rightarrow \psi_\beta(U_\alpha \cap U_\beta)$. Por dualidad, una forma diferencial arbitraria ω se transforma por multiplicación a la izquierda por la transpuesta de la matriz jacobiana $Jac(\psi_\beta \circ \psi_\alpha^{-1})^T$.

Como consecuencia, por cambio de carta un tensor de tipo (r, s) se transforma por multiplicación a la izquierda de un producto de r matrices jacobianas y un producto de s matrices transpuestas de las Jacobianas. Esta descripción es crucial para entender las transformaciones genéricas que tienen lugar sobre los nodos del retículo que representa la herramienta computacional TensorFlow.

Transformaciones tensoriales del mismo tipo.

Una convolución es básicamente un producto tensorial¹¹. Para tensores de grado total $r+s = 2$ se representa mediante una operación de producto tensorial (componente a componente) por una matriz de pesos $W = (w_{ij})$ que habitualmente (CNN, RNN) se supone cuadrada y con coeficientes constantes.

En Matemáticas no tiene por qué ser así: Las matrices pueden ser rectangulares y con pesos variables dados por funciones que ni siquiera tienen por qué ser continuas. El carácter rectangular de las matrices permite relacionar información que se procesa sobre capas consecutivas de una red neuronal, tal y como ocurre en las redes recurrente (RNN). Además, la generación de los mapas auto-organizantes se expresa en términos de submersiones locales. Por otro lado, los pesos variables se consideran de forma implícita en cuanto se introducen funciones no-lineales (de tipo sigmoideo, p.e.) que toman valores sobre las celdas de la máscara.

Para simplificar, en este apartado suponemos que las matrices de pesos a las que se llama *núcleo de la máscara*, son cuadradas y con pesos constantes, aunque con tamaño $k \times k$ ó variable. La aplicación de esta máscara sobre una red cuadrangular regular (correspondiente a los píxeles de una imagen ó células en la misma capa de una red neuronal, p.e.) se lleva a cabo de una forma iterativa mediante un procedimiento de barrido (habitualmente horizontal ó vertical).

Dos ejemplos básicos.

Dos “ejemplos” básicos para máscaras están asociados a procesos de suavizado o de realzado de las discontinuidades que aparecen en la señal. El ejemplo más simple corresponde a imágenes digitales en las que el proceso de suavizado genera un efecto de emborronamiento (dependiendo del tamaño de la máscara, lo cual permite suprimir outliers) y el proceso de realzado acentúa los perfiles (obteniendo “mejor definición” en los bordes, al tiempo que

¹¹En la literatura utilizada en Ingeniería adopta múltiples denominaciones como producto de Hadamard, Kronecker, Schur, entre otros, dependiendo de la tradición en la que se enmarca la escuela del autor y con diferentes símbolos lo cual contribuye a incrementar la confusión.

incrementa el “ruido” en el resto). Habitualmente, interesa aplicar los dos de forma secuencial¹².

Los diferentes procesos de filtrado (como producto tensorial) son bien conocidos y ampliamente utilizados para cualquier tipo de señales desde los años ochenta. El marco conceptual para estos desarrollos corresponde a los diferentes tipos de funciones que se pueden definir sobre un retículo regular. En otras palabras, los expertos en este área aún no han descubierto que, además de los campos escalares (funciones sobre variedades o funcionales sobre espacios de funciones), existen otros tipos de campos (co)vectoriales o con más generalidad, tensoriales; y eso, a pesar de que las operaciones están definidas por productos tensoriales.

3.4.2. Conservando el grado total.

Llamamos bigrado de un tensor de tipo (r, s) al par (r, s) asociado al número r de vectores y s de covectores. Llamamos “grado total” a la suma $r + s$. Algunas de las operaciones más relevantes tanto en Álgebra Multilineal como en Geometría Diferencial conciernen a transformaciones que conservan o alteran el grado. En esta subsección nos ocupamos de las primeras, mientras que en la siguiente nos ocupamos de las segundas.

La primera extensión significativa (que no aparece descrita en la literatura) consistiría en mostrar el significado de tensores de bigrado (s, r) para cada valor fijo del grado total $r + s$, en términos de las particiones de $s + r$. A pesar de la diversidad en las descripciones que aparecen más abajo, el nexo común está proporcionado por la existencia de un soporte común dado por convoluciones (autónomas o no, es decir, dependientes del tiempo) sobre el producto tensorial de un número creciente de s copias de V^* y de r copias de V .

Grado total 1.

El primer caso (casi-trivial) corresponde a $s + r = 1$ con dos particiones correspondientes a $(1, 0)$ (formas lineales φ^H sobre el espacio vectorial V ambiente) y $(0, 1)$ (vectores v o direcciones en el espacio ambiente); aunque son trivialmente duales entre sí, su significado es diferente pues en el primer caso estamos evaluando “atributos” (cuyo valor proporciona un número), mientras que en el segundo estamos evaluando “direcciones” (en términos de diferencias finitas como versión discreta de derivadas). En otras palabras, los datos son “puntuales” o “direccionales”.

Grado total 2.

El segundo caso corresponde a $s + r = 2$ con tres particiones correspondientes a $(2, 0)$ (pares de formas lineales φ_1^H, φ_2^H sobre el espacio vectorial V ambiente), $(1, 1)$ (formas lineales φ_1^H sobre el espacio vectorial V ambiente y vectores $v \in V$) y $(0, 2)$ (pares de vectores (v_1, v_2) o direcciones en el espacio ambiente). El significado es una extensión inmediata del comentado en el ítem anterior.

Grado total 3.

El tercer caso corresponde a $s + r = 3$ con cuatro particiones y un significado similar al expuesto más arriba. El interés de esta extensión radica en que actúa sobre mallas cúbicas “pesadas” gestionadas en términos de productos de hipermatrices (con un cubo de Rubik “pesado” como soporte para el caso más simple). Esta situación es clave para gestionar contenidos tridimensionales (asociados a una reconstrucción 3D, p.e.) o para gestionar contenidos 2D+1d (como los correspondientes a las secuencias de vídeo).

Aplicación a Redes Neuronales.

Desde el punto de vista de las redes neuronales, el caso correspondiente a grado total 2 se gestiona computacionalmente en términos de CNN (Convolutional Neural Networks). El caso correspondiente a grado total 3 sugiere la utilización de RNN (Recurrent Neural Networks) para incorporar mecanismos de retroalimentación más complejos basados en tracking que tengan en cuenta las capas L_0 (actual), L_{-1} (anterior para facilitar predicción) y L_{+1} (siguiente para validar la predicción) en tareas de seguimiento.

¹²El filtro de Canny (1984) proporciona una solución óptima para esta idea.

3.4.3. Incorporando pesos variables.

El modelo tradicional supone que, tras un proceso de aprendizaje, los pesos son constantes; en otras palabras, el dispositivo “inteligente” se vuelve “tonto” una vez aprendida la lección. Este comportamiento es incapaz de incorporar funcionalidades superiores relacionadas con la capacidad de modificación de pesos y la capacidad de auto-organización del conocimiento en función de nueva información. Las nuevas generaciones de Redes Neuronales requieren incorporar procedimientos más flexibles en la asignación de pesos en los mecanismos de realimentación interna.

Para incorporar la variabilidad de pesos en las plantillas, es necesario adoptar el enfoque de la Geometría Diferencial en la que las entradas de las hipermatrices se modifican en función de la variabilidad de campos vectoriales o de formas. Los diferentes tipos de “curvaturas” (tasas de variación de segundo orden) permiten representar dicha variabilidad.

Para representar dicha variabilidad se recurre a una representación intrínseca de la noción de derivada direccional que extiende a objetos geométricos (co-vectores o con más generalidad tensores) la noción bien conocida en Análisis para funciones (campos escalares). La evolución espacio-temporal de tensores en el espacio ambiente M se describe mediante la *derivada de Lie* \mathcal{L}_X de un tensor a lo largo de un campo vectorial X . Para fijar ideas, se describe la derivada de Lie para cada una de los tipos particulares de tensores.

Derivada de Lie de una función.

La derivada de Lie de una función es la extensión geométrica de la noción de derivada direccional utilizada en Análisis. La única diferencia radica en que para evitar la dependencia con respecto a la elección de coordenadas locales, se formula en términos vectoriales. Empezamos recordando la noción de derivada direccional que suponemos conocida de Análisis:

Definición 3.2 *Dado un campo vectorial X y una función $f : M \rightarrow \mathbb{R}$ sobre una variedad diferenciable M , se define la derivada direccional de f con respecto al campo X en el punto $p \in M$ mediante*

$$\partial_X f(p) = \lim_{t \rightarrow 0} \frac{f(p + tX(p)) - f(p)}{t}$$

Una versión geométrica más adaptada al contexto geométrico está dada por

Definición 3.3 *Dado un campo vectorial X y una función $f : M \rightarrow \mathbb{R}$ sobre una variedad diferenciable M , se define la derivada de Lie \mathcal{L} de f con respecto al campo X en el punto $p \in M$ mediante*

$$\mathcal{L}_X f(p) := \frac{d}{dt} f(\gamma(t))|_{t=0}$$

donde $\gamma(t)$ es una curva con un contacto de orden ≥ 2 con M en p y d es el operador diferencial. Formalmente, se representa la derivada de Lie como

$$\mathcal{L}_X(f) = X(f)$$

Derivada de Lie de un campo vectorial.

Fijado un campo vectorial $X \in \Theta_M$ sobre una variedad diferenciable, la *derivada de Lie* de cualquier campo vectorial $Y \in \Theta_M$ se define como el conmutador de ambos campos

$$\mathcal{L}_X(Y) = [X, Y] \quad \forall Y \in \Theta_M$$

Supongamos que $U \subset M$ es un abierto de trivialización para un atlas \mathcal{A} sobre M , con coordenadas locales $\underline{x} = (x_1, \dots, x_m)$ en un entorno de $p \in U \subset M$, y que

$$X|_U = \sum_{i=1}^m f_i(\underline{x}) \frac{\partial}{\partial x_i} \quad , \quad Y|_U = \sum_{j=1}^m g_j(\underline{x}) \frac{\partial}{\partial x_j}$$

son las expresiones locales de los campos X, Y sobre el abierto U . Entonces, el corchete de Lie $[X, Y]$ se escribe localmente como

$$[X, Y]|_U := \sum_{i,j=1}^m [f_i(\underline{x}) \frac{\partial g_j}{\partial x_i} \frac{\partial}{\partial x_j} - g_j(\underline{x}) \frac{\partial f_i}{\partial x_j} \frac{\partial}{\partial x_i}]$$

que es un campo vectorial bien definido sobre X verificando la propiedad antisimétrica. Utilizando el Teorema de la Función Implícita (regularidad de la matriz Jacobiana) para el cambio de carta, se obtiene una expresión similar. Por ello, la derivada de Lie $\mathcal{L}_X(Y)$ está definido globalmente como operador diferencial, es decir,

$$\mathcal{L}_X(Y) = X(Y(f)) - Y(X(f)) \quad \forall f \in \mathcal{E}_M$$

El carácter global es significativo para mantener la coherencia de los pegados locales o facilitar la transferencia de información entre diferentes agentes que estén capturando simultáneamente información desde diferentes localizaciones.

La identidad de Jacobi $[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$ se puede reformular en términos de derivadas de Lie observando que

$$\mathcal{L}_X[Y, Z] = [X, [Y, Z]] = [Y, [X, Z]] - [Z, [X, Y]] = Y\mathcal{L}_X Z - Z\mathcal{L}_X Y$$

lo cual muestra que la derivada de Lie es una anti-derivación sobre el álgebra de Lie de los campos vectoriales.

Derivada de Lie de una forma diferencial.

Denotemos mediante ϕ_t el flujo correspondiente a un campo vectorial $X \in \Theta_M$ (dado por las curvas integrales). Para cualquier forma diferencial $\alpha \in \Omega_M^1$ la derivada de Lie $\mathcal{L}_X(\alpha)$ de α en $p \in M$ se define como

$$\mathcal{L}_X(\alpha(p)) := \lim_{t \rightarrow 0} \frac{\phi_t^*(\alpha)(p) - \alpha(p)}{t}$$

que, intuitivamente corresponde a comparar dos hiperplanos (representando el soporte de las 1-formas) “infinitamente próximos”. Obviamente, la comparación debe hacerse sobre el mismo punto, lo cual justifica la utilización del “cambio de base” representado por $\phi_t^*(\alpha)$.

Con más generalidad, para cualquier par de campos vectoriales $X, Y \in \Theta_M$ y cualquier tensor T , la derivada de Lie de T con respecto al corchete $[X, Y]$ se define como

$$\mathcal{L}_{[X, Y]}T = \mathcal{L}_X\mathcal{L}_Y(T) - \mathcal{L}_Y\mathcal{L}_X T.$$

Hablando estrictamente no es propiamente una definición, sino el resumen de una colección de operaciones que, para ahorrar espacio no se llevan a cabo aquí.

Aplicando la derivada de Lie de un tensor.

El formalismo para la derivada de Lie de un tensor de tipo (s, r) es similar al descrito en los dos apartados anteriores. Debido a la notación explícita tan engorrosa, aquí no lo reproducimos; ver [29] para detalles.

La derivada de Lie de cualquier tensor de tipo (s, r) conserva el bigrado. Por ello, se trata de una operación interna que modifica el “aspecto” del tensor, pero no sus propiedades geométricas. En consecuencia, etiquetamos como “regulares” a este tipo de transformaciones que tienen lugar en los nodos dentro del retículo correspondiente a TensorFlow. Además de estas operaciones regulares, es interesante analizar transformaciones que alteran el bigrado conservando el grado total del tensor.

3.4.4. Contracción y expansión.

En el apartado anterior se ha descrito la derivada de Lie de un tensor como una transformación que conserva el tipo (s, r) del tensor. En particular, transforma funciones en funciones, campos en campos, y formas diferenciales en formas diferenciales del mismo grado total. Además de estas operaciones existen otras que permiten modificar tanto el tipo como el grado total $r + s$ del tensor. En algunos libros antiguos se denominaban a estas operaciones inducidas por una métrica ds^2 como “subida” y “bajada” de índices.

Transformaciones en grado total 1.

La transformación de un tensor de tipo $(1, 0)$ (covector) en uno de tipo $(0, 1)$ (vector) se realiza por dualidad. La conversión de un tensor de tipo $(2, 0)$ (métrica) en uno de tipo $(1, 1)$ (forma bilineal) se lleva a cabo contrayendo una 1-forma (mediante integración, p.e.) y expandiendo a lo largo de una dirección (dual de la 1-forma contraída, p.e.). Con más generalidad, es posible mantener el grado total $s+r$ de un tensor de tipo (s, r) ; en particular, estamos interesados en operaciones sobre los tensores de grado total 3 (para el análisis de secuencias de vídeo, p.e.) o bien 4 (para el análisis de volúmenes cambiantes, p.e.).

La contracción de la 1-forma dx_j se puede interpretar como una integración a lo largo de una fibración cuyas curvas integrales γ_i en la dirección de la variable contraída x_j (estamos reparametrizando la curva integral en términos de dicha variable). La expansión a lo largo de una 1-forma se interpreta como un producto exterior.

Transformaciones en grado superior.

Con más generalidad, se pueden contraer k -formas diferenciales de manera simultánea mediante integración sobre la fibra en el espacio total correspondiente al fibrado de las k -formas diferenciales. De forma inversa, el producto exterior por formas cuyo soporte yace en variedades transversales permite expandir el grado de la parte contravariante.

Las operaciones similares con campos vectoriales se interpretan como las duales de las dos operaciones descritas; en este caso, afectan a la aplicación fibración de una proyección.

Una aplicación a SOM.

Esta simple observación permite extender el enfoque de los mapas auto-organizantes de Kohonen en el marco tensorial; en este caso, el carácter auto-organizante procede de una evaluación de una submersión (“proyección no-lineal” en la terminología de Kohonen) sobre el espacio base dado por la capa neuronal siguiente. En el esquema original de Kohonen, el carácter auto-organizante se puede describir en términos de fibraciones sucesivas. En nuestro esquema, además de las contracciones sucesivas, también pueden aparecer expansiones locales que permiten “pegar” datos asociados a cilindroides o conoides.

Una especulación sobre tensores de grado 4.

Uno de los “ejemplos clásicos” más importantes de contracción de un tensor es la conversión del tensor $(3, 1)$ de curvatura de Riemann (en el espacio tiempo) en el tensor $(2, 0)$ de curvatura de Ricci (sobre una superficie). Esta operación es un “ejemplo” en el que se contrae un tensor de tipo $(1, 1)$, es decir, una forma bilineal a un punto; en particular, permite evaluar la *deformación real* que se presenta en la superficie de un objeto como resultado de una interacción (gravitatoria en el caso de Relatividad General).

En el análisis simultáneo de la forma y el movimiento, podemos partir de forma análoga de un tensor de tipo $(1, 3)$ en el que la 1-forma representa la sección temporal en el espacio-tiempo 4D que proporciona una representación volumétrica cambiante. La información relativa a la puesta en correspondencia de elementos homólogos contenidos en dos vistas próximas está “codificada” por una forma bilineal (matriz fundamental, en el caso afín) que es un tensor de tipo $(1, 1)$. Por consiguiente, la contracción del tensor de tipo $(1, 3)$ en otro de tipo $(1, 1)$ se lleva a cabo mediante una métrica sobre el espacio de campos vectoriales. La cuestión inmediata es: ¿Qué representa dicha 2-forma diferencial? La solución es sencilla: Es la contracción del plano tangente al objeto en la dirección del cono visual (envolvente de los rayos ópticos) que, algebraicamente, corresponde al cono tangente al lugar discriminante de la proyección sobre el plano de imagen.

La cuestión siguiente que se plantea es cómo estimar estos datos. Para ello, recurrimos a un ejemplo de estimación de la forma controlando las condiciones radiométricas. En términos estadísticos, podemos interpretar la matriz de covarianza (asociada a diferentes condiciones de iluminación sobre una superficie, p.e.) como una matriz de curvatura asociada a la respuesta a condiciones cambiantes “controladas”.

El movimiento relativo de un objeto con respecto a las fuentes de iluminación cambia la apariencia. Por ello, el esquema basado en la reinterpretación de las condiciones de iluminación controlada debería poderse extender a situaciones en las que no se dispone de dicho

control. La solución es clara: Basta estimar el mapa de reflectancia utilizando herramientas de IBR (Image Based Rendering) para el caso estático o de VBR (Video Based Rendering) para el caso dinámico. El enfoque tradicional que se hace en Visión Computacional o en Informática Gráfica utiliza funciones que es preciso estimar en cada punto por cada frame.

El enfoque basado en funciones (campos escalares) es tedioso y muy poco práctico. Para resolver el problema de forma dinámica, basta con reinterpretar el mapa de reflectancia como una 2-forma diferencial que se propaga sobre la superficie de cada objeto. Sus derivadas de Lie a lo largo de las direcciones planares del movimiento en la superficie de la escena proporcionan una distribución generada por dos campos vectoriales X_1, X_2 sobre un soporte suave a trozos.

La distribución $\mathcal{D} = \langle X_1, X_2 \rangle$ no tiene por qué ser integrable, pero cada curva integral corresponde a una trayectoria factible en la escena realizada por un agente móvil. De este modo, la solución al problema anterior se describe en términos de un tensor de tipo $(2, 2)$ que almacena información radiométrica (2-forma de reflectancia) y cinemática (campos vectoriales asociados al desplazamiento). Mediante transformaciones elementales podemos reconvertir en un tensor de tipo $(1, 3)$ ó de tipo $(3, 1)$ que, dependiendo de la información sobre la iluminación o sobre la escena sean más fáciles de reinterpretar. Cabe esperar que en un futuro próximo, TensorFlow proporcione una solución de forma automática a esta descripción.

Capítulo 4

Conclusiones

A lo largo de este TFG se han presentado varias líneas de trabajo que justifican el interés en la utilización de métodos topológicos para el análisis y la caracterización tanto de objetos, como de tareas. El modelado de objetos deformables se ha desarrollado en las categorías lineales a trozos y suaves a trozos, dentro del marco de la Topología Algebraica y de la Topología Diferencial, respectivamente. Se ha puesto especial interés en aspectos que puedan ser aplicados desde el punto de vista computacional, mostrando algunos de los métodos que se están desarrollando en ambos campos.

La utilización de PL y PS-complejos graduados para el modelado de objetos es común desde los años noventa. En nuestro caso, se ha puesto especial hincapié en el enfoque basado en complejos cuboidales (debido a una mayor facilidad en la parametrización y, por consiguiente, en la actualización de la información) y en algunos desarrollos recientes de Topología Diferencial como extensión de la Teoría de Morse. En ambos casos, se han introducido representaciones simbólicas basadas en diferentes tipos de grafos. Esto es básicamente lo que hemos tratado en el **Capítulo 1**.

En el **Capítulo 2**, de forma complementaria, se han introducido también las α -formas como un método de modelado que permite combinar PL-aproximaciones (sobre todo de tipo simplicial) con un tratamiento computacional próximo a la Geometría Computacional, lo cual permite una adaptación más natural de estrategias bien conocidas para el tratamiento computacional de la información.

El tratamiento de las tareas o procesos llevados a cabo por diferentes tipos de agentes es más novedoso. El enfoque habitual utiliza multi-caminos o multi-funciones (no confundir con funciones multivaloradas) que se pueden remitir a la Topología Algebraica tradicional (en términos de trenzas, p.e.) o bien a la cohomología (en términos de funciones definidas sobre complejos).

El tratamiento que se da a la información contenida en complejos graduados variables sugiere la introducción de diferentes tipos de campos que extiendan el enfoque cohomológico basado simplemente en clases de funciones verificando “buenas propiedades” (condición de ciclo, p.e.). Nuestro enfoque adopta el punto de vista de campos vectoriales y co-vectoriales, así como su extensión a campos tensoriales para representar cualquier tipo de flujo o de interacción dado sobre variedades o, con más generalidad, espacios topológicos.

El enfoque basado en campos es bien conocido en Física donde lleva utilizándose de forma sistemática desde los años cincuenta del siglo XX. Sin embargo, en Ingeniería y, de forma más específica, en las Tecnologías de Información y Comunicaciones es bastante menos frecuente, a pesar de estar subyacente a una gran cantidad de desarrollos recientes en Visión Computacional, Robótica y Sistemas Expertos, entre otras áreas. Por esta razón, esta memoria también trata de establecer puentes entre el marco de las Matemáticas y el marco de la Ingeniería (en lo referente al **Capítulo 3**).

Para ilustrar el alcance de los métodos basados en campos (escalares, vectoriales o tensoriales) se han esbozado algunos “ejemplos” procedentes de la Visión Computacional relacionados con la Visión Computacional y los Sistemas Expertos. El reconocimiento automático de objetos ó de tareas realizadas por agentes inteligentes proporciona un terreno común en el que se solapan ambas áreas de conocimiento. El marco natural para la superposición de ambas áreas está dado por las Redes Neuronales Artificiales (ANN: Artificial Neural Networks).

Algunos de los problemas iniciales presentados por las ANN en los años sesenta del pasado siglo estaban relacionados con la convergencia extremadamente lenta del aprendizaje, la rigidez de las redes una vez que han sido entrenadas y, por consiguiente, la incapacidad de auto-adaptarse a condiciones cambiantes (lo que se conoce como “*Los inviernos de la IA*”). Estos problemas han ido resolviéndose en los últimos veinte años del siglo XX, en términos de Sistemas Expertos que utilizan variantes de aprendizaje supervisado (basado en modelos geométricos) con respecto a no-supervisado (emergencia de patrones a partir del análisis estadístico de datos).

Una realimentación entre ambas estrategias de aprendizaje se da en el marco de Aprendizaje Profundo (Deep Learning) donde la utilización de dispositivos físicos con un número creciente de capas de neuronas y de distribución de procesos ha proporcionado un avance exponencial en la rapidez de cálculos y la convergencia de procesos. Las arquitecturas actualmente disponibles permiten entrenar procesos con varios millones de muestras a partir de un número relativamente reducido (unos pocos miles) de muestras etiquetadas correctamente.

Obviamente, la topología proporciona un lugar de encuentro para estas estrategias, pero para demostrarlo de forma explícita es necesario desarrollar un enfoque computacional de los campos. El tipo más general de campos en Geometría, Análisis o Topología está dado por los campos tensoriales. De forma complementaria, en el marco computacional se ha desarrollado una librería llamada TensorFlow que se puede representar simbólicamente como un retículo regular cuyos nodos son transformaciones y cuyas aristas representan tensores (el tipo más general de campos). Cada vez que un tensor atraviesa un nodo, el tensor se modifica de acuerdo con la operación representada por dicho nodo. Si el tensor resulta incompatible con la operación, el tensor atraviesa el nodo sin modificación alguna.

Ninguna de las transformaciones regulares (derivadas de Lie, p.e.) o irregulares (contracción vs expansión) de un tensor se ha considerado en la literatura sobre Sistemas Expertos. Por ello, el enfoque adoptado en este trabajo abre un montón de posibilidades para representar comportamientos que puedan incluir cambios de “tendencia” (“estado” en el caso de variedades, “fase” cuando analizamos la Cinemática).

Lo realmente relevante no es esta descripción novedosa, sino la posibilidad de que pueda ser aprendida de forma semi-automática por un dispositivo artificial: Este tópico va bastante más allá de la demostración automática de teoremas de finales de los ochenta y principios de los noventa, pues incorpora tipos de lógica más allá de la lógica de clases o de la lógica proposicional de primer orden, incorporando la lógica descriptiva (razonamiento bajo incertidumbre) que es característica de actividades humanas más complejas (como p.e. vinculadas al uso correcto del lenguaje hablado o simbólico). Esto es una idea importante que dejamos reflejado en este TFG.

Se ha vaticinado que hacia el año 2045, la Inteligencia Artificial habrá sobrepasado la inteligencia humana. Obviamente, esto plantea problemas éticos de otro tipo, pero están más allá del alcance de esta Memoria.

Bibliografía

- [1] Kunio Murasugi and Bohdan I. Kurpita; A Study of Braids (Vol.484); Kluwer; 1999.
- [2] J. W. Milnor; Morse Theory; Princeton University; 1973.
- [3] Marvin J. Greenberg; Lectures on Algebraic Topology; Benjamin; 1967.
- [4] William S. Massey; Singular Homology Theory; Springer-Verlag; 1980.
- [5] Albrecht Dold; Lectures on Algebraic Topology; Springer-Verlag; GMW; 1980.
- [6] Joseph O'Rourke; Computational Geometry in C; Cambridge University Press; 1994.
- [7] Franco P. Preparata and Michael Ian Shamos; Computational Geometry-An Introduction; Springer; 1985.
- [8] G. F. Voronoi; Nouvelles applications des paramètres continus à la théorie de formes quadratiques; 1908.
- [9] Afra J. Zomorodian; Topology for Computing, Cambridge University Press; 2005.
- [10] P. J. Giblin; Graphs, Surfaces and Homology; Chapman and Hall; 1976.
- [11] Herbert Edelsbrunner and John L. Harer; Computational Topology, An introduction; AMS; 2010.
- [12] T. Kohonen; Self-Organizing Maps (2nd. Edition); Springer-Verlag; 1997.
- [13] N. D. Gilbert and T. Porter; Knots and Surfaces; Oxford, 1994.
- [14] Ralph L. Cohen; The Topology of Fiber Bundles, Lecture Notes; Stanford University; 1998.
- [15] Ausgustin Banyaga and David Hurtubise; Lectures on Morse Homology; Kluwer Academic Publishers; 2004.
- [16] Michael Nielsen; Neuronal Networks and Deep Learning; web book; 2018.
- [17] T. Kaczynski, K. Mischaikow, M. Mrozek; Algebraic Topology: A Computational Approach; 2000.
- [18] L. Landau and E. Lifschitz; Théorie de l'Elasticité; Vol. 7 de Physique Théorique, Eds. Mir; 1967.
- [19] Joseph J. Rotman; An Introduction to Algebraic Topology; Springer; 1998.
- [20] H. Edelsbrunner, D.G. Kirkpatrick, and R. Seidel; On the shape of a set of points in the plane; IEEE Transactions on Information Theory; 1983.
- [21] H. Edelsbrunner and E.P. Mücke; Three-dimensional alpha shapes. ACM Transactions on Graphics; 1994.
- [22] Charles A. Weibel; An Introduction to Homological Algebra; Cambridge University Press; 1994.
- [23] Richard L. Bishop and Samuel I. Goldberg; Tensor Analysis on Manifolds; Dover Publications, Inc; 1980.
- [24] Luis A. Santaló; Vectores y Tensores con sus aplicaciones; Eudeba, B. Aires; 1981.

- [25] M. Martín y M. Pastor; Álgebra y cálculo tensorial, Ibergaceta Publicaciones, S.L.; Madrid, 2014.
- [26] R.Thom; Ensembles et morphisms stratifiés; Bull. Amer.Math Society; 1969.
- [27] P. K. Rasheski; Geometría Riemanniana y Análisis tensorial (Tomo 2); M. V. Lomonósov; 2017.
- [28] M. K. Hu; Visual Pattern Recognition by Moment Invariants; IRE Trans. Info. Theory, vol. IT-8; pp.179.187; 1962.
- [29] Abraham, Marsden and Raitu; manifolds, Tensor Analysis and Applicaitons (2nd ed), Springer-Verlag; 1983.
- [30] Rodney M. J. Cotterill; Computer simulation in brain science; Cambridge University Press; 1988.