Universidad de Valladolid

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Matemáticas

Modelos generativos profundos: Autocodificadores Variacionales

Autor: Álvaro Baños Izquierdo

Tutor/es: Eustasio del Barrio Tellado

Me gustaría agradecer por haberme acompañado en este largo camino a todos mis compañeros del grado, en especial a mis compañeros Ignacio, Jose y Gonzalo.

25 de Junio de 2022

ÍNDICE GENERAL

Resumen

En la actualidad, los modelos generativos basados en el aprendizaje profundo mediante técnicas de machine y deep learning han cobrado gran importancia debido a los resultados y avances que se están consiguiendo con su uso. Estos modelos se basan en tratamientos de grandes cantidades de datos mediante arquitecturas y técnicas de entrenamiento inteligente, teniendo una gran capacidad para generar nuevas distribuciones de datos muy realistas, como imágenes, textos o sonidos. Dentro de estos modelos, uno de los más conocidos son los Autocodificadores Variacionales, conocidos como VAEs, el cual es un autocodificador cuya distribución de codificaciones se regulariza durante el entrenamiento para garantizar que su espacio latente tenga buenas propiedades que nos permitan generar datos nuevos. Este trabajo se centra en el estudio de los fundamentos de estos modelos, partiendo de la técnica de reducción de dimensionalidad más sencilla, como es el Análisis de Componentes Principales, (ACP), hasta llegar a un análisis del estudio empírico de un modelo VAE, analizando y comparando los resultados obtenidos.

Abstract

Currently, generative models based on deep learning through machine and deep learning techniques have gained great importance due to the results and advances that are being achieved with their use. These models are based on the processing of large amounts of data through intelligent training architectures and techniques, with a great capacity to generate new and very realistic data distributions, such as images, texts or sounds. Among these models, one of the best known is the Variatonal Auto-Encoders, VAEs, which is an autoencoder whose distribution of encodings is regularised during training to ensure that its latent space has good properties that allow us to generate new data. This paper focuses on the study of the fundamentals of these models, starting from the simplest dimensionality reduction technique, Principal Component Analysis (ACP), to an analysis of the empirical study of a model VAE, analysing and comparing the results obtained.

CAPÍTULO 1

INTRODUCCIÓN AL TRABAJO DE FIN DE GRADO

Este trabajo de fin de grado, presenta un estudio sobre los modelos generativos profundos, los cuales son arquitecturas profundas dotadas de algoritmos de aprendizaje que tienen como misión aprender cualquier tipo de distribución de datos, pudiendo generar nuevas distribuciones similares a la inicial.

Para llegar a ellos, este trabajo sigue una estructura lineal, partiendo de lo más sencillo y conceptos más elementales, hasta llegar a los propios modelos generativos, estudiándolos tanto teóricamente como exponiendo algunas de sus múltiples aplicaciones.

Dentro de los modelos generativos profundos, estudiaremos los VAEs, conocidos como autocodificadores variacionales. Un VAE es un autocodificador que emplea algoritmos de aprendizaje profundo mediante los que se infiere una distribución continua de los datos de entrenamiento a través de un campo de la estadística que se conoce como inferencia variacional.

Los autocodificadores son redes neuronales artificiales, entrenadas de manera no supervisada, que tienen como objetivo aprender primero las representaciones codificadas de nuestros datos y luego generar los datos de entrada, lo más cercano posible, a partir de las representaciones codificadas aprendidas. Estos tienen una estructura codificador-decodificador, siendo el codificador el encargado de proyectar los datos iniciales a una dimensión menor, y el decodificador generará la nueva

distribución de datos en el espacio inicial a partir de los datos codificados en el espacio reducido.

En la propia definición surge un concepto muy importante el cual será nuestro punto de partida, la reducción de dimensionalidad.

El capítulo 2 comienza introduciendo este concepto explicando una de las técnicas más usada y sencilla pero a la vez más importante: el análisis de componentes principales, ACP. En este capítulo se hace un desarrollo teórico del ACP basado principalmente en Shai Shalev-Shwartz y Shai Ben-David [1]. Posteriormente, se exponen sus principales usos y limitaciones por los que surgirá la necesidad de usar modelos más complejos como son los autocodificadores. Antes de introducirlos, es necesario explicar la inferencia bayesiana como se hace en la sección 2.2.

En esta sección, se introduce y se desarrolla teóricamente el marco estadístico de la inferencia variacional, siendo un método estadístico muy importante para aproximar densidades de probabilidad.

Se explica en qué consiste el problema, exponiendo la familia variacional de campo medio y el algoritmo de inferencia variacional por ascenso de coordenadas (CAVI) como herramienta de optimización del problema. Este marco teórico se aplica a un caso concreto, la mezcla de distribuciones normales, desarrollado en la sección 2.3. Posteriormente, se expone un estudio empírico de Blei et al. [11] donde observamos los resultados de aplicar la inferencia variacional a un caso real.

Ya explicada la inferencia bayesiana, tenemos las herramientas para introducir a los autocodificadores, introducidos en la sección 2.4. En esta sección se explica el qué consiste un autocodificador exponiendo a su vez los problemas que nos encontramos si usamos modelos sencillos únicamente con redes neuronales. Para nuestro objetivo, la generación de nuevo contenido, necesitamos complicar un poco más el problema. Es por ello por lo que introducimos los autocodificadores variacionales.

Los autocodificadores variacionales se desarrollan en el capítulo 3, siendo una de las principales partes de este trabajo de fin de grado. Se enfoca de manera similar a las anteriores secciones, introduciendo la idea general y continuando con su desarrollo teórico, hasta llegar a exponer el trabajo empírico expuesto en Diederik P. and Max W. [18].

En este capítulo se introduce un nuevo estimador del límite inferior variacional, el gradiente estocástico VB (SGVB), para una inferencia aproximada eficiente con variables latentes continuas. Se expone el caso de conjuntos de datos i.i.d. y variables latentes continuas por punto de datos, introduciendo un algoritmo eficiente para la inferencia y el aprendizaje, el Auto-Encoding VB (AEVB), el cual aprende un modelo de inferencia aproximado utilizando el estimador SGVB. Finalmente, observamos las ventajas que proporciona este método en los resultados empíricos expuestos.

Con este capítulo finalizamos este trabajo de fin de grado. Se han asimilado tanto su funcionamiento como los algoritmos que se usan para su implementación, llegando a la conclusión de la amplia utilidad que tienen estos modelos generativos, así como de las múltiples líneas de investigación que existen actualmente, siendo sin lugar a dudas, una de las principales herramientas para la generación de contenido.

CAPÍTULO 2

REDUCCIÓN DE DIMENSIONALIDAD

La reducción de dimensionalidad es el proceso de tomar datos en un espacio de alta dimensión y proyectarlos a un espacio de menor dimensión, de tal manera que se pierda la menor cantidad de información posible. Existen varias razones para reducir la dimensionalidad de los datos. En primer lugar, la alta dimensionalidad puede dar lugar a una escasa capacidad de generalización del algoritmo de aprendizaje, por ello la reducción de dimensionalidad reduce de forma severa los costes del aprendizaje automático y permite la resolución de problemas complejos con modelos simples. Puede usarse también para la interpretabilidad de los datos, para encontrar una estructura significativa en los mismos, o con fines ilustrativos.

En esta parte del trabajo describiremos los principales métodos de reducción de dimensionalidad, comenzando con simples transformaciones lineales y finalizando con los métodos que usaremos en el estudio de los autocodificadores variacionales, (en inglés variational autoencoders, VAEs).

En la siguiente sección se mostrarán principalmente los resultados expuestos en Shai Shalev-Shwartz y Shai Ben-David [1].

2.1. Análisis de componentes principales.

Se usará cómo distancia la ya conocida norma euclídea, $\|\cdot\|$. Sea $\mathbf{x}_1, \cdots, \mathbf{x}_m$ m vectores en $\mathbb{R}^d$. Vamos a reducir la dimensionalidad de estos vectores mediante una

transformación lineal, por lo tanto, queremos una matriz $\boldsymbol{W}$ perteneciente a $\mathbb{R}^{n,d}$, con $n < d$, la cual nos permita obtener la proyeción en el espacio de menor dimensión. También queremos otra matriz $\boldsymbol{U}$ perteneciente a $\mathbb{R}^{d,n}$ que nos permita recuperar con la menor pérdida de información posible el vector original en el espacio de mayor dimensión.

En el *análisis de componentes principales*, (ACP), esto se traduce en la búsqueda de las matrices $\boldsymbol{W}$ y $\boldsymbol{U}$ tales que minimicen la distancia entre el vector original y el obtenido.

Por lo tanto, lo que buscamos es resolver el problema:

$$\underset{\mathbf{W}\in\mathbb{R}^{n,d},U\in\mathbb{R}^{d,n}}{argmin} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{U}\mathbf{W}\mathbf{x}_i\|_2^2. \tag{2.1}$$

Para la solución de este problema enunciamos el siguiente lema:

Lema 2.1. Sea ($\boldsymbol{W}$,$\boldsymbol{U}$) las matrices solución de la ecuación 2.1, entonces las columnas de $\boldsymbol{U}$ son ortonormales y $\boldsymbol{W}$=$\boldsymbol{U}^T$.

Demostración. Fijamos cualquier matriz $\boldsymbol{U}$ y $\boldsymbol{W}$, y consideramos una función $f : \mathbb{R}^d \to \mathbb{R}^d$ donde $f(\mathbf{x}) = \boldsymbol{U}\boldsymbol{W}\mathbf{x}$. El rango de esta función es $\mathbb{R} = \left\{\boldsymbol{U}\boldsymbol{W}\mathbf{x} : \mathbf{x} \in \mathbb{R}^d\right\}$, el cual es un subespacio lineal de $\mathbb{R}^d$.

Sea ahora $\boldsymbol{V} \in \mathbb{R}^{d,n}$, una matriz cuyas columnas forman una base del subespacio lineal $\mathbb{R}$, por lo tanto, sabemos que para todo $\mathbf{x} \in \mathbb{R}$ se puede escribir como combinación lineal de las columnas de la matriz $\boldsymbol{V}$, es decir, sean $\mathbf{v}_1, \cdots, \mathbf{v}_2$ las columnas de $\boldsymbol{V}$, existen $a_1, \cdots, a_n \in \mathbb{R}$ de tal manera que:

$$\mathbf{x} = a_1 \cdot v_1 + \cdots + a_n \cdot v_n; \quad \mathbf{x} \in \mathbb{R}^d. \tag{2.2}$$

LLamaremos $\mathbf{a} \in \mathbb{R}^n$ al vector formado por los anteriores coeficientes de tal manera que $\mathbf{x} = \boldsymbol{V}\mathbf{a}$. Para cada vector $\mathbf{x} \in \boldsymbol{R}^d$ y $\mathbf{a} \in \boldsymbol{R}^d$ tenemos que :

$$\|\mathbf{x} - \boldsymbol{V}\mathbf{a}\|_2^2 = \|\mathbf{x}\|^2 + \mathbf{a}^T\boldsymbol{V}^T\boldsymbol{V}\mathbf{a} - 2\mathbf{a}^T\boldsymbol{V}^T\mathbf{x} = \|\mathbf{x}\|^2 + \|\mathbf{a}\|^2 - 2\mathbf{a}^T\left(\boldsymbol{V}^T\mathbf{x}\right), \tag{2.3}$$

donde sabemos que $\boldsymbol{V}^T\boldsymbol{V} = 1$. Para minimizar la expresión anterior se realiza el gradiente con respecto a $\mathbf{a}$ igualándolo a 0, para lo cual se obtiene que $\mathbf{a} = \boldsymbol{V}^T\mathbf{x}$, lo

cual se cumple para todo vector $\mathbf{x} \in \mathbb{R}^d$, por lo que finalmente obtenemos para las matrices iniciales $\boldsymbol{U}$ y $\boldsymbol{W}$:

$$\sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{V}\mathbf{V}^T\mathbf{x}_i\|_2^2 \leq \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{U}\mathbf{W}\mathbf{x}_i\|_2^2, \tag{2.4}$$

probando lo que queríamos ver.

□

Aún podemos hacer una mayor simplificación del problema usando manipulaciones algebraicas elementales, de tal manera que para $\mathbf{x} \in \mathbb{R}^d$ y $\boldsymbol{V} \in \mathbb{R}^{d,n}$ tal que $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, tenemos:

$$\begin{aligned} \|\mathbf{x} - \boldsymbol{V}\boldsymbol{V}^T\|^2 &= \|\mathbf{x}\|^2 - 2\mathbf{x}^T\boldsymbol{V}\boldsymbol{V}^T\mathbf{x} + \mathbf{x}^T\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{V}^T\mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathbf{x}^T\boldsymbol{V}\boldsymbol{V}^T\mathbf{x} \\ &= \|\mathbf{x}\|^2 - \text{traza}\left(\boldsymbol{V}^T\mathbf{x}\mathbf{x}^T\boldsymbol{V}\right). \end{aligned} \tag{2.5}$$

Omitimos el subíndice 2 en la norma, ya que siempre nos referiremos a la norma euclídea, salvo que se indique lo contrario.

La traza hace referencia a la suma de los elementos de la diagonal de la matriz.

Teniendo en cuenta el lema anterior, y la simplificación mencionada, reescribiríamos el problema inicial 2.1 como sigue:

$$\underset{\boldsymbol{V} \in \mathbb{R}^{d,n}: \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}}{argmax} \text{traza}\left(\boldsymbol{V}\sum_{i=1}^{m}\mathbf{x}_i\mathbf{x}_i^T\boldsymbol{V}\right). \tag{2.6}$$

Definamos ahora la matriz simétrica $\mathbf{A} = \sum_{i=1}^{m}\mathbf{x}_i\mathbf{x}_i^T$, la cual al ser simétrica admite la descomposición como $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^T$, donde la matriz $\mathbf{D}$ es una matriz diagonal y $\mathbf{P}^T\mathbf{P} = \mathbf{I}$. Esto es posible, ya que es una matriz simétrica con coeficientes reales, por lo que se puede aplicar el teorema de descomposición-espectral.

Nota 2.2. El teorema espectral dice lo siguiente. Sea $\boldsymbol{A}$ una matriz de dimensión $n \times n$, de coeficientes reales y simétrica que verifica las siguientes propiedades:

- Sea $\boldsymbol{A}$ una matriz real simétrica de dimensión $n \times n$ que podemos escribir de

la forma $\boldsymbol{A} = \mathbf{P}\mathbf{D}\mathbf{P}^T$, donde las columnas de P son las coordenadas de los vectores propios ortonormales $\mathbf{v}_1, \cdots, \mathbf{v}_n$ de $\boldsymbol{A}$ y los correspondientes valores propios $\lambda_1, \cdots, \lambda_n$ están en la matriz diagonal $\mathbf{D}$. Antes de seguir recordamos que:

- $\boldsymbol{A}$ tiene n valores propios reales.
- Si λ es un valor propio de $\boldsymbol{A}$ con multiplicidad k, entonces el espacio propio asociado a λ es k-dimensional.
- Los espacios propios son mutualmente ortogonales, es decir: dos vectores propios que corresponden a dos valores propios distintos son ortogonales.

Entonces:

$$\begin{aligned} \boldsymbol{A} &= \sum_{i=1}^{n} \lambda_i \mathbf{A}_i \\ &= \lambda_1 \mathbf{A}_1 + \cdots \lambda_n \mathbf{A}_n, \end{aligned} \tag{2.7}$$

donde las matrices $\mathbf{A}_i = \mathbf{v}_i \mathbf{v}_i^T$, con $1 \leq i \leq n$, son simétricas, idempotentes y verifican:

$$\begin{aligned} \sum_{i=1}^{n} \mathbf{A}_i &= \sum_{i=1}^{n} \mathbf{v}_i \mathbf{v}_i^T = \mathbf{I}_n, \\ \mathbf{A}_i \mathbf{A}_j &= \left(\mathbf{v}_i \mathbf{v}_j^T\right)\left(\mathbf{v}_i \mathbf{v}_j^T\right) = \mathbf{0}_{nxn}, \quad \mathrm{i} \neq j. \end{aligned} \tag{2.8}$$

Con la nota anterior podemos garantizar que los elementos de la diagonal de la matriz $\mathbf{D}$ son los autovalores de la matriz $\boldsymbol{A}$ y las columnas de la matriz $\mathbf{P}$ son sus correspondientes autovectores.

Supondremos sin pérdida de generalidad que $\mathbf{D}_{1,1} \geq \mathbf{D}_{2,2} \cdots \mathbf{D}_{n,n} \geq 0$. Esto último lo podemos asegurar al ser la matriz $\mathbf{A}$ semidefinida positiva.

Teorema 2.3. Sea $\mathbf{x}_1, \cdots, \mathbf{x}_m$ vectores cualesquiera de $\mathbb{R}^d$, sea $\mathbf{A} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$ y sean $\mathbf{u}_1, \cdots, \mathbf{u}_n$ los n autovectores de la matriz $\mathbf{A}$ correspondientes a los n mayores auto-

valores de $\mathbf{A}$. Entonces, la solución al problema de optimización del ACP dado por la ecuación 2.1 es la matriz $\boldsymbol{U}$ cuyas columnas son los autovectores $\mathbf{u}_1, \cdots, \mathbf{u}_n$ y la matriz $\boldsymbol{W} = \boldsymbol{U}^T$.

Demostración. Sea $\mathbf{PDP}^T$ la descomposición espectral de la matriz $\mathbf{A}$, sea ahora $\mathbf{U} \in \mathbb{R}^{d,n}$ una matriz con columnas ortonormales. Definimos $\mathbf{B} = \mathbf{P}^T\mathbf{U}$, por lo que $\mathbf{PB} = \mathbf{PP}^T\mathbf{U} = \mathbf{U}$ lo que implica

$$\mathbf{U}^T\mathbf{AU} = \mathbf{B}^T\mathbf{V}^T\mathbf{VDV}^T = \mathbf{B}^T\mathbf{DB}, \tag{2.9}$$

por lo tanto

$$\text{traza}\left(\mathbf{U}^T\mathbf{AU}\right) = \sum_{j=1}^{d} \mathbf{D}_{j,j} \sum_{i=1}^{n} \mathbf{B}_{j,i}^2. \tag{2.10}$$

Ahora hay que tener en cuenta que las columnas de $\mathbf{B}$ son ortonormales, ya que $\mathbf{B}^T\mathbf{B} = \mathbf{U}^T\mathbf{PP}^T\mathbf{U} = \mathbf{U}^T\mathbf{U} = \mathbf{I}$, lo que implica que la suma de la norma al cuadrado de las columnas de la matriz $\mathbf{B}$ es n, es decir, $\sum_{j=1}^{d}\sum_{i=1}^{n} \mathbf{B}_{j,i}^2 = n$. Usaremos ahora una matriz auxiliar $\mathbf{G} \in \mathbb{R}^{d,d}$, cuyas primeras n columnas son las columnas de $\mathbf{B}$ y además $\mathbf{G}^T\mathbf{G} = \mathbf{I}$, por lo tanto, para cada j tenemos que $\sum_{i=1}^{d} \mathbf{G}_{j,i}^2 = 1$ lo que implica que $\sum_{i=1}^{n} \mathbf{B}_{j,i}^2 \leq 1$. Volviendo a la primera ecuación tendríamos

$$\text{traza}\left(\mathbf{U}^T\mathbf{AU}\right) \leq \sum_{j=1}^{d} \mathbf{D}_{j,j}. \tag{2.11}$$

Escogiendo como matriz $\mathbf{U}$ la matriz cuyas n columnas son los n primeros autovectores de $\mathbf{A}$, obtenemos que

$$\text{traza}\left(\mathbf{U}^T\mathbf{AU}\right) = \sum_{j=1}^{d} \mathbf{D}_{j,j}, \tag{2.12}$$

lo que conluye finalmente la prueba. □

Nota 2.4. Es común centralizar las muestras antes de aplicar el ACP, es decir, normalizar los vectores iniciales, $\mu = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i$, denotando también $\mathbf{r}_i = (\mathbf{x}_i - \mu)$ y posteriormente aplicar el ACP en los vectores $\mathbf{r}_1, \cdots, \mathbf{r}_n$.

Ya hemos resuelto el problema de análisis de componentes principales, ahora trataremos una situación particular en la cual podremos encontrar una solución más eficiente.

2.1.1. Solución óptima para el caso $m \ll d$.

En la práctica, es un caso muy común que la dimensión del espacio sea mucho mayor que el número de vectores que estemos tratando. Hay que tener en cuenta que el coste computacional en el problema del ACP es del orden de $O(d^3)$, correspondiente al cálculo de los autovalores de $\mathbf{A}$, más $O(md^2)$ para el cálculo de construcción de la matriz $\mathbf{A}$. Vamos a ver como podemos reducir considerablemente el coste computacional aprovechando esta particular situación.

Definiremos primero la matriz $\mathbf{X} \in \mathbb{R}^{m,d}$ como la matriz cuyas i-ésima fila es $\mathbf{x}_i^T$. De esta manera redefiniremos la matriz $\mathbf{A}$ definida en el apartado anterior, como $\mathbf{A} = \mathbf{X}^T\mathbf{X}$. Consideramos la matriz $\mathbf{B} = \mathbf{X}\mathbf{X}^T \in \mathbb{R}^{m,m}$.

Ahora supongamos que u es un autovector de $\mathbf{B}$, es decir, existe $\lambda \in \mathbb{R}$ tal que $Bu = \lambda u$, multiplicando a ambos lados de la igualdad por $\mathbf{X}^T$ y usando la definición de $\mathbf{A}$ obtenemos que $\mathbf{A}\left(\mathbf{X}^T u\right) = \lambda\left(\mathbf{X}^T u\right)$. Observando esta expresión se deduce directamente que $\dfrac{\mathbf{X}^T u}{\|\mathbf{X}^T u\|}$ es un autovector de $\mathbf{A}$ con autovalor asociado λ.

De esta manera reducimos la solución del problema de ACP al simple cálculo de multiplicación de vectores. Mediante este procedimiento el coste computacional sería del orden de $O(m^3)$, correspondiente al cálculo de los autovalores de $\mathbf{b}$, más $O(m^2 d)$ para la construcción de la matriz $\mathbf{B}$, por lo que se consigue una reducción importante en el coste computacional, lo que es esencial en la aplicación de los métodos en proyectos reales.

2.1.2. Principales usos y limitaciones del ACP.

Como ya se ha mencionado, mediante el ACP se pueden identificar las variables o características que aportan mayor información, pudiendo descartar aquellas que son menos relevantes y trabajar con un menor número de variables, simplificando el problema y agilizando el proceso de modelado. Por lo tanto, su principal aplicación es la **reducción de dimensionalidad**, sin embargo, no es su única aplicación.

Otra aplicación para la cual se puede aplicar el ACP es para la **detección de anomalías**, lo cual se podría aplicar en casos como la detección de fraude en transacciones bancarias. Mediante el uso del ACP se analizan las características o atributos que definen lo que corresponden un comportamiento "bueno", pudiendo posteriormente aplicar métricas que permitan detectar los casos que se alejan de ese comportamiento, es decir, los casos "malos".

El Análisis de Componentes Principales es una buena técnica de reducción de dimensionalidad, la cual nos permite proyectar una serie de datos en un espacio de menor dimensión mediante operadores lineales perdiendo la menor cantidad de información posible. Se podría entender el ACP como un caso particular de autocodificador, usando, por lo tanto, codificadores-decodificadores como redes neuronales de una sola capa y lineales. Sin embargo, con los autocodificadores podemos abarcar un marco mayor, ya que podemos utilizar codificadores y decodificadores con un número mayor de capas tanto lineales como no lineales.

El propósito de este estudio es la generación de nuevos datos a partir de un conjunto inicial. En este punto hay que mencionar una importante limitación de los autocodificadores, (como el ACP), que es la dificultad de garantizar una cierta regularidad en el espacio reducido, (espacio que se obtiene al aplicar la matriz que proyecta los datos a un espacio de menor dimensión), a causa del sobreajuste que se puede producir en el proceso.

Como consecuencia, el espacio reducido puede ser muy irregular, (puntos cercanos en el espacio reducido pueden dar datos decodificados muy diferentes, o puntos que no se encuentren en el espacio inicial), por lo que no se puede definir un proceso generativo que consista en coger puntos del espacio reducido y decodificarlos. Aquí surge la necesidad de implementar métodos algo más complejos que solventen este problema, para ello, primero explicaremos conceptos matemáticos que necesitaremos en desarrollos posteriores.

2.2. Inferencia Bayesiana.

Uno de los problemas centrales de la estadística moderna es la aproximación de densidades de probabilidad difíciles de calcular. Este problema es especialmente importante en la estadística bayesiana, que enmarca toda la inferencia sobre canti-

dades desconocidas como un cálculo que implica la densidad a posteriori.

En este trabajo de fin de grado, revisamos la inferencia variacional (VI, en inglés variational inference), un método de aprendizaje automático que aproxima las densidades de probabilidad mediante la optimización. La VI se ha utilizado en muchas aplicaciones y tiende a ser más rápido que los métodos clásicos, como el muestreo de cadena de Markov Montecarlo.

La idea detrás de VI es plantear primero una familia de densidades y luego encontrar el miembro de esa familia que se acerque al objetivo. La cercanía se mide por la divergencia de Kullback-Leibler.

Comenzaremos introduciendo el problema de inferencia bayesiana, el cual conlleva una serie de dificultades computacionales que nos llevarán a la necesidad de tener que usar el método de Inferencia Variacional.

Sea $\mathbf{x}$ una variable generada por una distribución de propabilidad p_θ, la cual depende de un parámetro desconocido θ. Asumiremos que poseemos conocimientos previos sobre el parámetro θ el cual puede expresarse como una densidad de probabilidad, $p(\theta)$. Cuando observamos el valor de la variable x, podemos actualizar nuestros conocimientos previos sobre este parámetro aplicando el teorema de Bayes:

$$p\left(\theta|x\right) = \frac{p\left(x|\theta\right)p(\theta)}{p(x)}, \tag{2.13}$$

Observamos que la computación anterior requiere tres términos:

- La probabilidad $p\left(x|\theta\right)$, es la distribución de probabilidad del dato observado con un valor dado del parámetro θ.
- La distribución de probabilidad del parámetro, $p(\theta)$, independiente de cualquier observación.
- La distribución de probabilidad marginal, $p(x)$, a la que llamaremos evidencia, es la distribución de probabilidad del dato observado independiente de cualquier valor del parámetro.

Los dos primeros términos son fácilmente expresados y se pueden dar por conocidos ya que en la mayoría de casos se conocen a priori. Sin embargo, el tercer

término se calcula de la siguiente manera:

$$p(x) = \int_{\theta} p\left(x|\theta\right) p(\theta) d\theta. \tag{2.14}$$

En dimensiones bajas esta integral puede ser calculada sin muchas dificultades, sin embargo, en dimensiones altas se hace prácticamente imposible. Para poder calcularla, se aplican ténicas de aproximación, entre las cuales las principales son **La Cadena de Markov Monte Carlo** , (MCMC), y la **Inferencia Variacional**, (VI). En este trabajo nos centraremos en la Inferencia Variacional (Jordan, M. I. et al. [2]).

2.2.1. Inferencia variacional

El objetivo de la inferencia variacional es aproximar una densidad condicional de las variables latentes dadas las variables observadas. La idea clave es resolver este problema mediante la optimización. Utilizamos una familia de densidades sobre las variables latentes, parametrizadas por parámetros variacionales libres. La optimización encuentra el miembro de esta familia, es decir, el ajuste de los parámetros, que más se acerca en la divergencia Kullback-Leibler, (KL), a la condicional de interés.

Se ha mencionado en el anterior párrafo la divergencia KL, la cual es es una medida no simétrica de la similitud o diferencia entre dos funciones de distribución de probabilidad P y Q. Se define formalmente a continuación:

Definición 2.5. Sean P y Q las distribuciones de probabilidad de una variable aleatoria discreta, su divergencia KL se define como:

$$KL(P||Q) = \sum_{i} P(i) \ln \frac{P(i)}{Q(i)}. \tag{2.15}$$

Es el promedio ponderado de la diferencia logarítmica entre las probabilidades P y Q. La divergencia KL solamente se define si P y Q suman 1 y si $Q(i) > 0$ para cualquier i tal que $P(i) > 0$. Si la cantidad $0 \cdot \ln 0$ aparece en la fórmula, se interpreta como cero.

Definición 2.6. Sean P y Q las distribuciones de probabilidad de una variable aleatoria continua, su divergencia KL se define como:

$$KL(P||Q) = \int_{-\infty}^{\infty} P(x) ln \frac{p(x)}{q(x)} dx. \tag{2.16}$$

En la sección 3.1.2.1 se muestra un ejemplo de cálculo analítico de la divergencia KL aplicado al caso más común, entre distribuciones normales, dentro del marco de los autocodificadores variacionales.

Continuamos con el problema de inferencia variacional. Sea $\mathbf{x} = x_{1:n}$ un vector de variables observadas, y sea $\mathbf{z} = x_{1:m}$ un vector de variables latentes, desconocidas, con una densidad de probabilidad conjunta $p(\mathbf{z}|\mathbf{x})$. El objetivo de la inferencia es hallar la densidad condicional de las variables latentes dadas las condicionadas, $p(\mathbf{z}|\mathbf{x})$. Esta la podemos reescribir como:

$$p\left(\mathbf{z}|\mathbf{x}\right) = \frac{p\left(\mathbf{x}|\mathbf{z}\right)p(\mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{z},\mathbf{x})}{p(\mathbf{x})}, \tag{2.17}$$

el denominador de la expresión anterior definido anteriormente como la *evidencia*, se puede calcular como:

$$p(\mathbf{x}) = \int p(\mathbf{z},\mathbf{x})d\mathbf{z}. \tag{2.18}$$

Como ya se indicó en la introducción, el cálculo de esta integral en grandes dimensiones puede llegar a ser inabarcable. Necesitamos la evidencia para hallar la condicional a partir del conjunto, es por ello que la aplicación de la inferencia en estos modelos es complicada.

Asumiremos que todas las cantidades desconocidas de interés se representan como variables aleatorias latentes, esto incluye los parámetros que podrían gobernar todos los datos, y las variables latentes que son locales para los puntos de datos individuales.

Sea ς una familia de densidades sobre el espacio de variables latentes. Cada $q(\mathbf{z}) \in \varsigma$ es una aproximación a la condicional exacta, siendo nuestro objetivo encontrar el mejor candidato, es decir, el candidato que esté más cerca en términos de divergencia de KL de la condicional exacta. Por lo tanto, nuestro problema se reduce a resolver la siguiente optimización:

$$q^*(\mathbf{z}) = \underset{q(\mathbf{z})\in\varsigma}{argmin}\ \ KL\left(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})\right). \tag{2.19}$$

Sin embargo, este cálculo tampoco es posible ya que requiere realizar el cálculo de $\log p(x)$ en la ecuación 2.18.

Recordemos que podemos escribir la divergencia KL como (Jordan, M. I. y Wainwright, M. [3]):

$$\begin{aligned} KL\left(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})\right) &= \mathbb{E}[\log q(\mathbf{z})] - \mathbb{E}[\log p(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}[\log q(\mathbf{z})] - \mathbb{E}[\log p(\mathbf{z},\mathbf{x})] + \log p(\mathbf{x}). \end{aligned} \tag{2.20}$$

Observamos que aparece la expresión que estamos intentando evitar, lo que nos impide poder calcular explícitamente la divergencia KL, por ello se usará la función conocida como *límite inferior de la evidencia* (en inglés evidence lower bound, ELBO), (Diederik P. y Welling, Max [4]) introducida en la siguiente definción:

Definición 2.7. Sean X y Z dos variables aleatorias con distribución conjunta $p(X, Z)$, la cual depende de z, siendo esta un parámetro que parametriza la distribución. Z es una variable aleatoria de la que no se tiene información, la variable aleatoria latente.
Sea q la función de distribución que sigue la variable Z. Se conoce como límite inferior de la evidencia a

$$ELBO = \mathbb{E}_{Z\sim q}\left[\log \frac{p(X,Z)}{q(Z)}\right]. \tag{2.21}$$

Retomando la ecuación 2.20 observamos que

$$\begin{aligned} KL\left(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})\right) &= \mathbb{E}[\log q(\mathbf{z})] - \mathbb{E}[\log p(\mathbf{z},\mathbf{x})] + \log p(\mathbf{x}) \\ &= -\mathbb{E}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right] + \log p(\mathbf{x}). \end{aligned} \tag{2.22}$$

Directamente se deduce que maximizar la expresión de ELBO es equivalente a minimizar la divergencia KL.

Observamos que $\log p(\mathbf{x}) \geq ELBO(q)$ para cualquier densidad $q(\mathbf{z})$. También se observa que el primer término de la ELBO, $\log p(\mathbf{x}, \mathbf{z})$, se puede optimizar mediante el algoritmo de esperanza-maximización EM[1] (Dempster et al, [5]). El algoritmo EM fue diseñado para encontrar estimaciones de máxima verosimilitud en modelos

[1]El algoritmo EM es un proceso iterativo para maximizar verosimilitud en presencia de datos faltantes, alterna pasos de esperanza (E), donde se computa la esperanza de la verosimilitud mediante la inclusión de variables latentes como si fueran observables, y un paso de maximización (M), donde se computan estimadores de máxima verosimilitud de los parámetros mediante la maximización de la verosimilitud esperada del paso E

con variables latentes.

A diferencia de la inferencia variacional, EM asume que la esperanza bajo $p(\mathbf{z}|\mathbf{x})$ es computable y la utiliza en problemas de estimación de parámetros, además la inferencia variacional no estima parámetros fijos del modelo, sino que se utiliza a menudo en un entorno bayesiano en el que los parámetros clásicos se tratan como variables latentes. La inferencia variacional se aplica a los modelos en los que no podemos calcular la condicional exacta de las variables latentes.

El problema de inferencia, que inicialmente consistía en la optimización de la ecuación 2.19, finalmente consiste en la optimización de la función ELBO anteriormente expuesta. Para ello especificaremos la familia variacional ς para completar los detalles del problema de optimización variacional.

2.2.1.1. La Familia Variacional del Campo Medio.

Es importante elegir una familia de densidades adecuada, ya que la complejidad de la familia elegida determinará la complejidad del problema de optimización.

En este estudio nos centraremos en la *familia variacional del campo medio*.

Definición 2.8. La familia variacional de campo medio es una familia de densidades de probabilidad sobre el espacio de variables latentes, en el cual todas las variables son independientes entre sí y cada una esta gobernada por un factor distinto en la densidad variacional. Toda densidad perteneciente a esta familia se puede expresar de la siguiente manera:

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j). \tag{2.23}$$

Cada variable latente z_j se rige por su propio factor variacional, la densidad $q_j(z_j)$, estos factores variacionales se eligen para maximizar la ecuación ELBO.

La familia de campo medio es relevante porque puede capturar cualquier densidad marginal de las variables latentes. Sin embargo, no puede capturar correlación entre ellas. Ver esto en acción revela algunas de las intuiciones y limitaciones de la inferencia variacional de campo medio.

Utilizando la ELBO y la familia de campo medio, hemos planteado la inferencia variacional como un problema de optimización. En la siguiente sección, describimos uno de los algoritmos más utilizados para resolver este problema de optimización, la inferencia variacional de ascenso por coordenadas (CAVI).

2.2.1.2. Inferencia Variacional de campo medio por ascenso de coordenadas.

El algoritmo CAVI es una técnica de optimización matemática que pertenece a la familia de los algoritmos de búsqueda local. Es un algoritmo iterativo que comienza con una solución arbitraria a un problema, y continúa intentando para encontrar una mejor solución variando incrementalmente un único elemento de la solución.

Algoritmo: Primero llamaremos *densidad condicional de* z_j a la densidad dada por el resto de variables latentes y las observadas, es decir $p(z_j|\mathbf{z}_{-j}, \mathbf{x})$.

Fijando el resto de factores variacionales $q_l(z_l),\ l \neq j$, el óptimo $q_j(z_j)$ es entonces proporcional a la condicional esperada del logaritmo de la distribución condicional completa, lo que significa (Bishop [6]):

$$q_j^*(z_j) \propto exp\left(\mathbb{E}_{-j}[\log p(z_j|\mathbf{z}_{-j}, \mathbf{x})]\right) \propto exp\left(\mathbb{E}_{-j}[\log p(z_j, \mathbf{z}_{-j}, \mathbf{x})]\right). \tag{2.24}$$

Esta última proporción se debe al hecho de que estamos usando la familia de campo medio, en la cual habíamos asumido la independencia entre sí de las variables latentes.

Estas ecuaciones son la base del algoritmo CAVI. Mantendremos un conjunto de factores $q_l(z_l)$, e iteraremos a través de ellos actualizando los factores $q_j(z_j)$ usando la ecuación 2.24.

Mostraremos primero la estructura del algoritmo para a continuación explicar como efectivamente se van actualizando los términos $q_j(z_j)$ llegando finalmente a la maximización de la función ELBO.

Algorithm 1 Ascenso por coordenadas, *CAVI*.

Input: La densidad $p(\mathbf{x}, \mathbf{z})$ y el conjunto de datos $\mathbf{x}$.

Output: La densidad variacional $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$.

1: Los factores variacionales $q_j(z_j)$
2: **while** *ELBO no converja* **do**
3: **for** $j \in 1, \cdots, m$ **do**
4: $q_j(z_j) \propto exp\left(\mathbb{E}_{-j}[\log p(z_j, \mathbf{z}_{-j}, \mathbf{x})]\right)$
5: **end for**
6: **end while**
7: **return** $q_j(z_j)$

Tal como se ve en la ecuación 2.25, el algoritmo CAVI es un algoritmo de ascenso a un máximo local. Ahora, derivamos las coordenadas actualizadas en la ecuación 2.24 y escribimos la definición del ELBO en función del j-ésimo factor variacional $q_j(z_j)$, juntando todos los términos que no dependen de este factor en una sola constante C. Reescribiremos el primer término de la ecuación 2.22 usando la anterior expresión iterada, el segundo término lo descompondremos usando la suposición de independencia de las variables latentes, manteniendo solo los términos que dependen de $q_j(z_j)$:

$$ELBO(q_j) = \mathbb{E}_j[\mathbb{E}_{-j}[\log p(z_j, \mathbf{z}_{-j}, \mathbf{x})]] - \mathbb{E}_j[\log q_j(z_j)] + C, \tag{2.25}$$

como la densidad q es producto de las q_j, dejamos el resto fijo y maximizamos esta ecuación respecto $q_j(z_j)$ para cada paso, garantizando que cada vez obtengamos un resultado más próximo al máximo local de la función ELBO. De esta manera estaremos resolviendo el problema inicial, finalizando cuando lleguemos a $q_j(z_j) = q_j^*(z_j)$.

2.2.2. Aspectos Prácticos.

En esta sección vamos a explicar algunos aspectos que hay que tener en cuenta al aplicar la inferencia variacional.

La función ELBO, generalmente es una función objetivo no convexa. El algoritmo CAVI, explicado en la sección anterior, únicamente nos garantiza encontrar un máximo local, por lo que no tiene por qué ser el máximo absoluto. Se puede comprobar

para el caso de un modelo de distribuciones normales, como la función ELBO tiene diferentes óptimos locales, dentro de los cuales, los mejores máximos locales darán lugar a densidades variacionales más próximas a la distribución posterior exacta. Esto realmente no tiene por qué ser siempre una desventaja, lo cual lo mostraremos en el caso práctico que se explicará en la siguiente sección.

Otro aspecto importante a tener en cuenta es el hecho de que en el algoritmo CAVI, se evalúa la convergencia una vez que el cambio en la función ELBO ha entrado dentro de un determinado umbral. Sin embargo, podemos encontrar algún inconveniente en este proceso, ya que evaluar la función ELBO en todo el conjunto de datos puede llegar a ser inoperable o demasiado costoso. En su lugar se puede calcular la media de la predicción logarítmica de un pequeño conjunto de datos representativos de la muestra total, a esto se le conoce como la probabilidad de predicción retenida.

Por último, hay que tener en cuenta que en el manejo de probabilidades estaremos tratando valores en el intervalo $[0, 1]$, lo que implica que en muchas ocasiones estaremos usando valores muy pequeños, lo que conlleva un cuidado adicional, por ello, es recomendable trabajar con logaritmos de probabilidades. Será útil la siguiente identidad:

$$\log\left\{\sum_i exp(x_i)\right\} = a + \log\left\{\sum_i exp(x_i - a)\right\}. \tag{2.26}$$

Donde la constante a se fija en un máximo de x_i. De esta manera obtendremos estabilidad numérica en los procedimientos de inferencia variacional.

Establecidos los aspectos a tener en cuenta y la teoría sobre la inferencia variacional, vamos a exponer en la siguiente sección el ejemplo práctico con el *modelo de mezcla bayesiana de distribuciones normales*.

2.3. Mezcla bayesiana de distribuciones normales.

Los resultados expuestos en esta sección están fundamentalmente ilustrados en el artículo de Blei et al. [11].

Definición 2.9. Llamaremos *mezcla bayesiana de distribuciones normales univariantes*

de varianza unitaria a un conjunto de K distribuciones normales con medias $\boldsymbol{\mu} = \{\mu_1, \cdots, \mu_K\}$.

Los parámetros medios se extraen de forma independiente de una distribución a priori, (previa), común $p(\mu_K)$, la cual asumiremos que es una distribución normal $\mathcal{N}(o, \sigma^2)$, siendo la varianza previa σ^2 un hiperparámetro.

Definición 2.10. Denotaremos como *cluster* c_i a un vector K-indicador, es decir, un vector con todas sus componentes nulas excepto la i-ésima posición.

Definición 2.11. La *distribución categórica* o *distribución de Bernoulli generalizada* es una distribución de probabilidad discreta que describe los posibles resultados de una variable aleatoria que puede tomar una de K categorías posibles, con la probabilidad de cada categoría especificada por separado. Los parámetros que especifican las probabilidades de cada resultado posible están limitados solo por el hecho de que cada uno debe estar en el rango de 0 a 1, y todos deben sumar 1. (Murphy, K.P. [10]).

Para generar una observación x_i del modelo, primero elegiremos una asignación de clústeres c_i, el cual nos indicará que conglomerado x_i se usará extrayendo una distribución categórica sobre $\{1, \cdots, K\}$. Por lo tanto, extraeremos x_i de la distribución normal correspondiente $\mathcal{N}(c_i^T \boldsymbol{\mu}, 1)$.

De esta manera, el modelo jerárquico completo sería el siguiente:

$$\begin{aligned} &\mu_k \sim \mathcal{N}(0, \sigma^2) \ ; && k = 1, \cdots, K, \\ &c_i \sim \text{Categorica}(1/K, \cdots, 1/K) \ ; && i = 1, \cdots, n, \\ &x_i | c_i, \boldsymbol{\mu} \sim \mathcal{N}(c_i^T \boldsymbol{\mu}, 1) \ ; && i = 1, \cdots, n. \end{aligned} \tag{2.27}$$

Siendo n el tamaño de la muestra. La densidad conjunta de las variables latentes y observadas quedaría de la siguiente manera:

$$p(\boldsymbol{\mu}, \mathbf{c}, \mathbf{x}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \boldsymbol{\mu}) \tag{2.28}$$

Siendo en este modelo las variables latentes $\mathbf{z} = \{\boldsymbol{\mu}, \mathbf{c}\}$, es decir, las medias de K clases, y las asignaciones de n clases.

La ya conocida *evidencia*, en este modelo sería:

$$p(\mathbf{x}) = \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} \sum_{c_i} p(c_i) p(x_i|c_i, \boldsymbol{\mu}) d\boldsymbol{\mu}.$$

Hay que tener en cuenta que en la integral anterior cada μ_k aparece en todos los n factores del integrando, por lo que no tiene un factor separado para cada μ_k, y por lo tanto no se puede reducir a un producto de integrales unidimensionales sobre las μ_k, siendo el coste operacional de evaluar numéricamente esta integral del orden de $O(K^n)$.

Sin embargo, si distribuimos el producto seobre la suma en la ecuación anterior y reordenamos, podemos escribir la anterior expresión como:

$$p(\mathbf{x}) = \sum_{\mathbf{c}} p(\mathbf{c}) \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(x_i|c_i, \boldsymbol{\mu}) d\boldsymbol{\mu}. \tag{2.29}$$

Hemos reducido la compleja y costosa integral anterior a un conjunto de integrales computables. No obstante, sigue habiendo un total de K^n integrales, una para cada configuración de las asignaciones de los clusters, por lo que este cálculo seguiría siendo intratable.

La densidad conjunta de las variables latentes y observadas es la ilustrada en la ecuación 2.28, solo nos quedaría determinar la familia variacional de este modelo. Para ello usaremos la ya ilustrada familia variacional de campo medio, la cual contendría las siguientes densidades posteriores aproximadas:

$$q(\boldsymbol{\mu}, \mathbf{c}) = \prod_{k=1}^{K} q(\mu_k; m_k, s_k^2) \prod_{i=1}^{n} q(c_i; \phi_i). \tag{2.30}$$

Basándonos en lo expuesto en la sección 2.2.1.1, cada variable latente se rige por su propio factor variacional variante. El factor $q(\mu_k; m_k, s_k^2)$ es una distribución normal en el parámetro de la media de la k-ésima componente de la mezcla, es decir, su media es m_k y su varianza s_k^2. El factor $q(c_i; \phi_i)$ es una distribución sobre la asignación de la mezcla de la i-ésima observación, es decir, sus probabilidades de asignación son un K-vector ϕ_i.

Los componentes de la mezcla son distribuciones normales con parámetros va-

riacionales (media y varianza) específicos del k-ésimo clúster. Recordamos también que la asignación de clústeres son categóricos con parámetros variacionales específicos para el i-ésimo punto de datos. Esta sería la forma óptima de la densidad variacional del campo medio para la mezcla de distribuciones normales.

Habiendo establecido ya la familia variacional, ya hemos especificado completamente el problema de inferencia variacional. La ELBO está definida por la definición del modelo en la ecuación 2.28 y la familia de campo medio en la ecuación 2.30. A continuación, explicaremos el correspondiente problema de optimización de la función ELBO para este modelo.

Como hemos visto, en este modelo tenemos dos tipos de parámetros variacionales: los parámetros categóricos ϕ_i para aproximar la asignación posterior de clústeres de la i-ésima observación, y los parámetros Gausssianos m_k y s_k^2 para aproximar la distribución posterior del k-ésimo componente de la mezcla. De esta manera vamos a combinar la familia conjunta y la del campo medio para formar la función ELBO de parámetros variacionales $\mathbf{m}$, $\mathbf{s^2}$ y ϕ, para el modelo de mezcla de Gaussianos:

$$\begin{aligned} ELBO(\mathbf{m}, \mathbf{s}^2, \boldsymbol{\phi}) = &\sum_{k=1}^{K} \mathbb{E}\left\{\log p(\mu_k); m_k, s_k^2\right\} \\ &+ \sum_{i=1}^{n} \left(\mathbb{E}\left\{\log p(c_i); \phi_i\right\} + \mathbb{E}\left\{\log p(x_i|c_i, \boldsymbol{\mu}); \phi_i, \mathbf{m}, \mathbf{s}^2\right\}\right) \\ &- \sum_{i=1}^{n} \mathbb{E}\left\{\log q(c_i; \phi_i)\right\} - \mathbb{E}\left\{\log q(\mu_k; m_k, s_k^2)\right\}. \end{aligned} \tag{2.31}$$

En la anterior ecuación cada esperanza se puede calcular de forma cerrada. Mediante el algoritmo CAVI vamos actualizando cada parámetro variacional.

Primero, derivaríamos la actualización para el factor variacional de asignación de clústeres, y posteriormente derivaríamos la actualización para el factor variacional de componente de la mezcla, lo cual se especifica en los siguientes apartados:

2.3.1. Densidad variacional de la asignación de la mezcla.

Como hemos indicado, en primer lugar derivamos la actualización variacional para la asignación del clúster c_i, para ello usaremos la ecuación 2.24,

$$q^*(c_i, \phi_i) \propto exp\left\{\log p(c_i) + \mathbb{E}\left[\log p(x_i|c_i, \boldsymbol{\mu}); \mathbf{m}, \mathbf{s}^2\right]\right\}. \tag{2.32}$$

Los términos de la función exponencial son los componentes de la densidad conjunta que dependen de c_i. El segundo término es la esperanza sobre los componentes de la mezcla μ.

El primer término de la ecuación anterior es la log-probabilidad a priori de c_i. Para todos los posibles valores de c_i esta tiene el mismo valor, $\log p(c_i) = -\log K$. El segundo término es la log-esperanza de la densidad de la distribución normal, y teniendo en cuenta que c_i es un vector indicador, podemos escribir:

$$p(x_i|c_i, \boldsymbol{\mu}) = \prod_{k=1}^{K} p(x_i|\mu_k)^{c_{ik}}. \tag{2.33}$$

Usamos la expresión anterior para hallar la esperanza de la probabilidad logarítmica:

$$\begin{aligned}
\mathbb{E}\left\{\log p(x_i|c_i, \boldsymbol{\mu})\right\} &= \sum_k c_{ik} \mathbb{E}\left\{\log p(x_i|\mu_k); m_k, s_k^2\right\} \\
&= \sum_k c_{ik} \mathbb{E}\left\{-(x_i - \mu_k)^2/2; m_k, s_k^2\right\} + Cons \\
&= \sum_k c_{ik} \left(\mathbb{E}\left\{\mu_k; m_k, s_k^2\right\} x_i - \mathbb{E}\left\{\mu_k^2; m_k, s_k^2\right\}/2\right) + Cons,
\end{aligned} \tag{2.34}$$

donde *Cons* representa una constante.

En cada línea de la ecuación anterior hemos eliminado los componentes constantes respecto de c_i. El cálculo que obtenemos requiere $\mathbb{E}(\mu_k)$ y $\mathbb{E}(\mu_k^2)$, para cada componente de la mezcla, ambos computables a partir de la distribución normal variacional en la k-ésima componente de la mezcla.

De esta manera la actualización de la i-ésima asignación de clúster sería

$$\phi_{ik} \propto exp\left[\mathbb{E}\left\{\mu_k; m_k, s_k^2\right\} x_i - \mathbb{E}\left\{\mu_k^2; m_k, s_k^2/2\right\}\right]. \tag{2.35}$$

Nos quedaría una función que depende únicamente de los parámetros variacionales de los componentes de la mezcla.

2.3.2. Densidad variacional de las medias de los componentes de la mezcla.

Nos centramos ahora en la densidad variacional $q(\mu_k; m_k, s_k^2)$ de la k-ésima componente de la mezcla. Volvemos a usar la ecuación 2.24 y escribimos la densidad conjunta hasta una constante de normalización,

$$q(\mu_k) \propto exp\left[\log p(\mu_k) + \sum_{i=1}^{n} \mathbb{E}\left\{\log p(x_i|c_i, \boldsymbol{\mu}); \phi_i, m_{-k}, \mathbf{s}_{-k}^2\right\}\right], \tag{2.36}$$

recordemos que ϕ_{ik} es la probabilidad de que la i-ésima observación provenga del k-ésimo clúster.

Ahora calcularemos el logaritmo no normalizado de esta coordenada óptima $q(\mu_k)$. Teniendo en cuenta que c_i es un vector indicador, podemos establecer que $\phi_{ik} = \mathbb{E}\{c_{ik}; \phi_i\}$. Por lo que nos quedaría:

$$\begin{aligned}
\log q(\mu_k) &= \log p(\mu_K) + \sum_i \mathbb{E}\left\{\log p(x_i|c_i, \boldsymbol{\mu}); \phi_i, \mathbf{m}_{-k}, \mathbf{s}^2\right\} + Cons \\
&= \log p(\mu_K) + \sum_i \mathbb{E}\{c_{ik} \log p(x_i|\mu_k); \phi_i\} + Cons \\
&= -\mu_k^2/2\sigma^2 + \sum_i \mathbb{E}\{c_{ik}; \phi_i\} \log p(x_i|\mu_k) + Cons \\
&= -\mu_k^2/2\sigma^2 + \sum_i \phi_{ik}(-(x_i - \mu_k)^2/2) \log p(x_i|\mu_k) + Cons \\
&= -\mu_k^2/2\sigma^2 + \sum_i \phi_{ik} x_i \mu_k - \phi_{ik}\mu_k^2/2 + Cons \\
&= \left(\sum_i \phi_{ik} x_i\right) \mu_k - \left(1/2\sigma^2 + \sum_i \phi_{ik}/2\right) \mu_k^2 + Cons.
\end{aligned} \tag{2.37}$$

Con el cálculo anterior observamos que la densidad variacional óptima por coordenadas de μ_k es una familia exponencial con estadísticos $\{\mu_k, \mu_k^2\}$ y parámetros

naturales $\left\{ \sum_{i=1}^{n} \phi_{ik} x_i, -1/2\sigma^2 - \sum_{i=1}^{n} \phi_{ik}/2 \right\}$, es decir, una normal.

Las acutalizaciones para $q(\mu_k)$ en términos de la media y varianza variacional quedarían:

$$m_k = \frac{\sum_i \phi_{ik} x_i}{1/\sigma^2 + \sum_i \phi_{ik}}, \qquad s_k^2 = \frac{1}{1/\sigma^2 + \sum_i \phi_{ik}}. \tag{2.38}$$

Estas actualizaciones están estrechamente relacionadas con la densidad condicional completa de la k-ésima componente en el modelo de la mezcla. Sabemos que la condicional completa es una distribución posterior normal dados los datos asignados a la k-ésima componente.

Finalmente, la actualización variacional sería una condicional completa ponderada, donde cada punto de datos se pondera por su probabilidad de ser asignado al componente k.

2.3.3. Algoritmo CAVI para el modelo de Mezcla de Gaussianos.

El algoritmo presentado en la sección 2.2.1.2 presenta la inferencia variacional por ascenso de coordenadas para un modelo genérico. Mediante lo visto en esta sección, vamos a particularizar el algoritmo CAVI para el modelo mezcla bayesiana de distribuciones normales. Para ello, combinaremos las ecuaciones 2.32 y 2.38. También, el algoritmo requiere computar la ecuación ELBO 2.31. Utilizaremos la ecuación ELBO para seguir el progreso del algoritmo y evaluar cuándo ha convergido.

Algorithm 2 CAVI para el modelo de mezcla de normales.

Input: Set de datos $x_{1:n}$, número de componentes K y la varianza previa de las medias de las componentes σ^2.

Output: Densidad variacional normal $q(\mu_k; m_k, s_k^2)$, y la densidad variacional K-Categórica $q(c_i; \phi_i)$.

Parámetros variacionales $\mathbf{m} = m_{1:K}, \mathbf{s}^2 = s_{1:K}^2$ y $\boldsymbol{\phi} = \phi_{1:n}$.

while *ELBO no converja* **do**

  for $i \in 1, \cdots, n$ **do**

$$\phi_{ik} \propto exp\left(\mathbb{E}\left\{\mu_k; m_k, s_k^2\right\} x_i - \mathbb{E}\left\{\mu_k^2; m_k, s_k^2\right\} / 2\right).$$

  end for

  for $k \in 1, \cdots, K$ **do**

$$\begin{aligned} m_k &\leftarrow \frac{\sum_i \phi_{ik} x_i}{1/\sigma^2 + \sum_i \phi_{ik}}, \\ s_k^2 &\leftarrow \frac{1}{1/\sigma^2 + \sum_i \phi_{ik}}. \end{aligned} \tag{2.39}$$

  end for

end while

Computa $ElBO(\mathbf{m}, \mathbf{s}^2, \boldsymbol{\phi})$.

return $q(\mathbf{m}, \mathbf{s}^2, \boldsymbol{\phi})$.

Una vez que tenemos una densidad variacional ajustada, podemos utilizarla como utilizaríamos la distribución a posteriori. Por ejemplo, podemos obtener una descomposición posterior de los datos. Asignamos los puntos a su asignación de mezcla más probable $c_i^* = \text{argmax}\, x_k \phi_{ik}$ y estimamos las medias de los clústeres con sus medias variacionales m_k.

También podemos usar la densidad variacional ajustada para aproximar la densidad predictiva de los nuevos datos. Esta predicción aproximada es una mezcla de normales,

$$p(x_{new}|x_{1:n}) \approx \frac{1}{K} \sum_{k=1}^{K} p(x_{new}|m_k), \tag{2.40}$$

donde $p(x_{new}|x_{1:n})$ es una distribución normal con media m_K y varianza unitaria.

2.3.4. Estudio empírico.

Para ilustrar lo explicado en las anteriores secciones, mostraremos el estudio práctico expuesto en Blei et al. [11]. Nos centraremos en el primer análisis de este artículo observando y comprobando el funcionamiento del algoritmo de mezcla de distribuciones normales.

Estudio de simulación: Se considera los datos bidimensionales reales **x**. Se simuló con $K = 5$ normales con medias, covarianzas y asignaciones de mezcla aleatorias, mostrados en la siguiente figura:

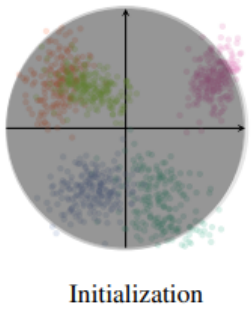

Figura 2.1: Datos iniciales del estudio de simulación en 2 dimensiones del modelo de Mezcla de normales
Source: variacionales Inference: A Review for Statisticians, David M. et al. [11].

Cada punto está coloreado según su verdadero cluster. La figura anterior también ilustra la densidad inicial de los componentes de la mezcla: cada uno es una normal, casi centrada y con una amplia varianza.

En las siguientes imágenes observamos los valores anteriores a medida que progresa el algoritmo CAVI.

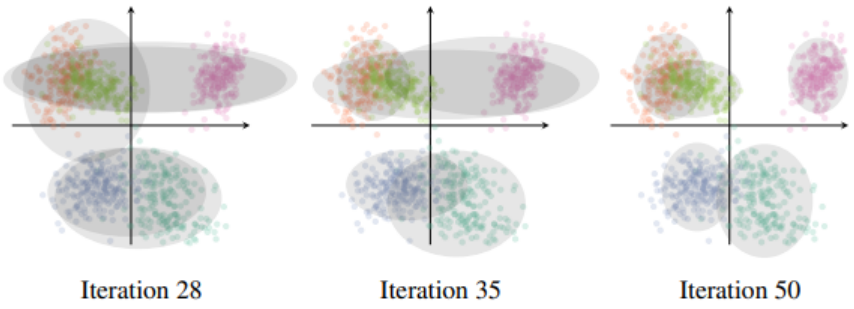

Figura 2.2: Datos del estudio de simulación en 2 dimensiones del modelo de mezcla de normales a durante el algoritmo CAVI
Source: variacionales Inference: A Review for Statisticians, David M. et al. [11].

Es importante interpretar el progreso de la función ELBO. Se pueden observar los puntos clave en los que la ELBO desarrolla *codos*, fases de la maximización en las que la aproximación variacional cambia de forma. Estos *codos* surgen porque la ELBO no es una función convexa en términos de los parámetros variacionales, es decir, el CAVI alcanza iterativamente mejores mesetas.

2.3.5. Discusión y problemas abiertos

Recapitulando, hemos descrito la inferencia variacional, un método que utiliza la optimización para realizar cálculos probabilísticos. El objetivo es aproximar la densidad condicional de las variables latentes $\mathbf{z}$ dadas las variables observadas $\mathbf{x}$, $p(\mathbf{z}|\mathbf{x})$.

La idea como hemos visto es plantear una familia de densidades Q y luego encontrar el miembro $q^*(\cdot)$ que más se acerque en divergencia KL a la condicional de interés. La minimización de la divergencia KL es el problema de optimización, y su complejidad se rige por la complejidad de la familia de aproximación.

A continuación hemos descrito la familia de campo medio, es decir, la familia de densidades totalmente factorizadas de las variables latentes. Utilizando esta familia, la inferencia variacional es particularmente susceptible de optimizarse por ascenso de coordenadas, algoritmo que optimiza iterativamente cada factor.

Finalmente mostramos cómo utilizar el campo medio de inferencia variacional para aproximar la densidad posterior de una mezcla bayesiana de normales, exponiendo los resultados del análisis práctico presentado en la anterior sección.

Aunque todavía no se ha desarrollado mucha teoría en torno a la inferencia variacional, hay varios hilos de investigación sobre las garantías teóricas de las aproximaciones variacionales que se muestran a continuación:

- You, C. et al.[12] estudian el posterior variacional para un modelo lineal bayesiano clásico.
- Hall, P. et al. [13] examinan un modelo simple de efectos mixtos de Poisson, uno con un solo predictor y un intercepto aleatorio. Utilizan una aproximación variacional gaussiana y estiman los parámetros con EM variacional.
- Celisse, A. et al. [14] analizan los datos de la red utilizando modelos de bloques estocásticos. Muestran la normalidad asintótica de las estimaciones de los parámetros obtenidas mediante una aproximación variacional de campo medio.
- Westling, T. y McCormick, T. [15] estudian la consistencia de VI a través de una conexión con M-estimación. Se centran en una clase más amplia de modelos (con soporte posterior en el espacio de coordenadas real espacio de coordenadas reales) y analizan una técnica de VI automatizada que utiliza una aproximación variacional Gaussiana.
- Wang, B. y Titterington, D. [16] analizan aproximaciones variacionales a mezclas de normales. Específicamente, consideran mezclas bayesianas con densidades previas conjugadas, la aproximación variacional del campo medio y un estimador que es la media posterior variacional. Confirman que CAVI converge a un óptimo local, que el estimador VI es consistente y que la estimación VI y la estimación de máxima verosimilitud se aproximan entre sí en un rango del orden de $O(1/n)$.

En general, todos los estudios anteriores tratan las medias a posteriori de la inferencia variacional como estimaciones puntuales y confirman que tienen la asintótica frecuentista habitual. Cada resultado gira en torno a un único modelo y una única familia de aproximaciones variacionales.

Además de todos los estudios mencionados, existen muchas otras vías abiertas para la investigación estadística en la inferencia variacional. Por ejemplo, hasta ahora nos hemos centrado en la optimización de la divergencia $KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$ como función objetivo. Una prometedora vía de investigación es el desarrollo de métodos de inferencia variacional que optimicen otras medidas como la propagación de esperanzas mostrada en Minka, T. P. [17].

Las propiedades estadísticas de la inferencia variacional aún no se comprenden bien, especialmente en contraste con la riqueza del análisis de las técnicas MCMC. Sin embargo, hemos podido comprobar en esta sección su funcionamiento y su fuerte potencial en múltiples aplicaciones, como es en el caso de los autocodificadores, sobre lo que se tratará en la siguiente sección.

2.4. Autocodificadores.

Vamos a hablar ahora de los autocodificadores y a ver cómo podemos utilizar las redes neuronales para la reducción de la dimensionalidad. La idea general es bastante sencilla y consiste en configurar un codificador y un decodificador como redes neuronales y aprender el mejor esquema de codificación-decodificación mediante un proceso de optimización iterativo. Así, en cada iteración alimentamos la arquitectura del autocodificador (codificador-decodificador) con algunos datos, comparamos la salida codificada-decodificada con los datos iniciales y retropropagamos el error a través de la arquitectura para actualizar los pesos de las redes.

Supongamos primero que nuestras arquitecturas de codificador y decodificador tienen una sola capa lineal, (autocodificador lineal). Entonces, dicho codificador y decodificador son simples transformaciones lineales que pueden expresarse como matrices. En esta situación, podemos ver un claro vínculo con el ACP, ya que al igual que en este caso el caso, estamos buscando el mejor subespacio lineal para proyectar los datos con la menor pérdida de información posible. Las matrices de codificación y decodificación obtenidas con ACP definen naturalmente una de las soluciones que alcanzaríamos mediante el descenso de gradiente, pero no es la única. De hecho, se pueden elegir varias bases para describir el mismo subespacio óptimo y, por tanto, varios pares de codificador/decodificador pueden dar el error de reconstrucción óptimo.

Ahora, supongamos que tanto el codificador como el decodificador son profundos y no lineales. En este caso, cuanto más compleja sea la arquitectura, mayor capacidad tendrá el autocodificador de proceder a una reducción de dimensionalidad elevada manteniendo una pérdida de reconstrucción baja.

Debemos tener en cuenta dos aspectos:

- En primer lugar, una importante reducción de la dimensionalidad sin pérdida de reconstrucción suele conllevar una falta de regularidad en el espacio reducido.
- En segundo lugar, como ya se ha mencionado anteriormente el objetivo final de la reducción de la dimensionalidad es reducir el número de dimensiones manteniendo la mayor parte de la información de la estructura de los datos en el espacio reducido.

Por estas dos razones, la dimensión del espacio latente y la complejidad de los autocodificadores deben controlarse y ajustarse cuidadosamente en función del objetivo final.

Nuestro objetivo es la generación de nuevos datos a partir de un conjunto inicial, sin embargo, hasta ahora no se ha introducido un proceso real que genere una nueva distribución de datos, simplemente se ha explicado los mecanismos de reducción de dimensionalidad de un espacio inicial perdiendo la menor cantidad de información posible.

En este punto hay que mencionar una importante limitación de los autocodificadores: la dificultad de garantizar una cierta regularidad en el espacio reducido, a causa del sobreajuste que se puede producir en el proceso. Esto se debe a que la regularidad del espacio reducido para los autocodificadores depende de la distribución de los datos en el espacio inicial, de la dimensión del espacio latente y de la arquitectura del codificador, por lo que el espacio reducido puede llegar a ser muy irregular (puntos cercanos en el espacio reducido pueden dar datos decodificados muy diferentes, o puntos que no se encuentren en el espacio inicial) por lo que no se puede definir un proceso generativo que consista en coger puntos del espacio reducido y decodificarlos.

Esto ocurre efectivamente en el ACP debido al alto grado de libertad del autocodificador que hace posible codificar y decodificar sin pérdida de información, lo

que conduce a un severo sobreajuste que implica que algunos puntos del espacio reducido generarán contenido sin sentido una vez decodificados, es decir, una vez nos salgamos de los datos iniciales, no estaremos obteniendo datos coherentes con el espacio inicial.

Aún usando redes neuronales en el autocodificador se sigue encontrando este problema, ya que no hay nada en la tarea de entrenar el autocodificador que haga que se mantenga una determinada estructura en el espacio reducido. Esto es porque el autocodificador se entrena únicamente para codificar y decodificar con la menor pérdida posible, sin importar cómo esté organizado el espacio reducido. Es por ello por lo que introducimos los *autocodificadores variacionales*, los cuales nos solucionanarán el problema de irregularidad del espacio reducido.

CAPÍTULO 3

AUTOCODIFICADORES VARIACIONALES

Hemos visto que mediante inferencia variacional bayesiana obtenemos una optimización de la aproximación de la distribución latente. Sin embargo, la inferencia de campo medio requiere soluciones analíticas de las esperanzas con respecto a la distribución posterior (o posteriori) aproximada, las cuales en muchos casos son intratables o demasiado ineficientes.

En este capítulo vamos a estudiar como una cota inferior variacional produce un simple estimador insesgado y diferenciable de la cota inferior. Este estimador se conoce como SGVB (Stochastic Gradient Variational Bayes). Se usa principalmente para obtener una inferencia posterior aproximada en la mayoría de modelos con variables latentes continuas, además es fácil de optimizar usando técnicas como el ascenso de gradiente estocástico.

Salvo que se indique lo contrario, supondremos un conjunto inicial de variables latentes continuas independientes igualmente distribuidas (i.i.d). Estudiaremos el algoritmo de Autocodificación VB, (AEVB), en el cual se consigue que la inferencia y el aprendizaje sean especialmente eficiente utilizando el estimador SGVB para optimizar un modelo de reconocimiento que nos permite realizar inferencia posterior aproximada muy eficiente, que a su vez nos permite aprender eficientemente los parámetros del modelo.

Este modelo de inferencia posterior aproximado también puede utilizarse para una

serie de tareas como el reconocimiento, la eliminación de ruido, la representación y la visualización. Usando una red neuronal para el modelo de reconocimiento, llegaremos a los *autocodificadores variacionales*, VAE, (en inglés variational autoencoders).

3.1. Presentación del problema.

Como ya se ha indicado, en este trabajo nos limitaremos al caso en el que tenemos un conjunto de datos i.i.d. con variables latentes por punto de datos. Realizaremos una inferencia de máxima verosimilitud (ML) o máxima a posteriori (MAP) sobre los parámetros globales, y la inferencia variacional sobre las variables latentes.

Consideraremos un conjunto de N muestras i.i.d de una variable aleatoria, continua o discreta, $\mathbf{x} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$. También supondremos que los datos son generados por un proceso aleatorio el cual implica una variable aleatoria latente $\mathbf{z}$.

Los valores $\mathbf{z}^{(i)}$ son generados por una distribución a priori $p_{\boldsymbol{\theta}^*}(\mathbf{z})$, los valores $\mathbf{x}^{(i)}$ son generados por una distribución condicional $p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{z})$. Supondremos que las distribuciones de probabilidad $p_{\boldsymbol{\theta}^*}(\mathbf{z})$ y $p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{z})$ vienen de una familia de distribuciones paramétricas $p_{\boldsymbol{\theta}}(\mathbf{z})$ y $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, diferenciables casi siempre respecto a $\boldsymbol{\theta}$ y $\mathbf{z}$. Hay que tener en cuenta que en este proceso los valores reales de los parámetros $\boldsymbol{\theta}^*$ y $\mathbf{z}^{(i)}$ son desconocidos para nosotros.

Nuestro objetivo inicial es la búsqueda de un algoritmo que sea eficiente en un marco general en el que:

- La integral de la distribución de probabilidad marginal $p(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\, p_{\boldsymbol{\theta}}(\mathbf{z}) d\mathbf{z}$ y la distribución de probabilidad condicional $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \dfrac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\, p_{\boldsymbol{\theta}}(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$ no sea posible llevar a cabo su cálculo.
- La dimensión del conjunto de datos inicial sea demasiado grande, lo que conlleva a que no sea posible la aplicación de soluciones basadas en el muestreo como el algoritmo de Monte Carlo, ya que su tiempo de ejecución sería demasiado alto.

De esta manera estamos estudiando la solución de los tres problemas siguientes:

- Estimar de manera óptima los parámetros $\boldsymbol{\theta}$ mediante el método de ML o MAP. Esto nos permitirá imitar el proceso aleatorio oculto y generar datos artificiales que se parezcan a los reales.
- Estimar eficientemente la variable latente $\mathbf{z}$ mediante inferencia posterior aproximada dado un valor $\mathbf{x}$ para una elección determinada de parámetros $\boldsymbol{\theta}$. Nos permitirá realizar tareas de codificación o de representación de datos.
- Estimar la variable $\mathbf{x}$ de manera eficiente mediante inferencia marginal aproximada. Nos permitirá realizar todo tipo de tareas de inferencia. Un ejemplo sería la eliminación de ruido en imágenes o el repintado.

Ahora introducimos el modelo de reconocimiento $q_{\phi}(\mathbf{z}|\mathbf{x})$. Esta distribución de probabilidad es una aproximación a la distribución $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. A diferencia de la aproximación en la inferencia variacional de campo medio, sus parámetros ϕ no se calculan a partir de una esperanza de forma cerrada. En su lugar, introduciremos un método para aprender los parámetros ϕ del modelo de reconocimiento conjuntamente con los parámetros del modelo generativo $\boldsymbol{\theta}$.

Este problema esta directamente relacionado con los autocodificadores explicados en la sección anterior. Las variables no observadas $\mathbf{z}$ tienen una interpretación como latente o código.

También se entiende el modelo de reconocimiento $q_{\phi}(\mathbf{z}|\mathbf{x})$ como un codificador probabilístico, ya que dado un punto del conjunto de datos $\mathbf{x}$ produce una distribución sobre los posibles valores del código $\mathbf{z}$ a partir de los cuales se podría haber generado ese punto $\mathbf{x}$. En la misma línea entendemos $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ como un decodificador probabilístico, ya que dado un valor del código $\mathbf{z}$ produce una distribución sobre los posibles correspondientes valores de $\mathbf{x}$.

3.1.1. El límite variacional.

En esta sección nos basaremos en los resultados que aparecen en Blei et al. [7], y en los vistos en la sección 2.2.1 de inferencia variacional.

Sabemos que podemos escribir el logaritmo de las probabilidades marginales de la siguiente manera:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{N}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}). \tag{3.1}$$

Haciendo referencia a la ecuación 2.22, podemos reescribir esta expresión como

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + ELBO(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}). \tag{3.2}$$

Ya hemos visto que el término $KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)}))$ es no negativo, y por ello el término $ELBO(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ lo denominamos *límite inferior variacional*. Podemos asegurar que:

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) &\geq ELBO(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left\{ -\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \right\} = \\ &= -KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left\{ \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) \right\}. \end{aligned} \tag{3.3}$$

El problema de inferencia variacional se reduce en optimizar la expresión $ELBO$ respecto al parámetro variacional $\boldsymbol{\phi}$ y $\boldsymbol{\theta}$. Una solución común suele usar el estimador de gradiente estocástico Monte Carlo, el cual consistiría en calcular (Blei et al. [7]):

$$\nabla \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})} \left\{ f(\mathbf{z}) \right\} = \nabla \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})} \left\{ f(\mathbf{z}) \nabla_{q_{\boldsymbol{\phi}}(\mathbf{z})} \log q_{\boldsymbol{\phi}}(\mathbf{z}) \right\} \mathbf{x} \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}) \nabla_{q_{\boldsymbol{\phi}}(\mathbf{z}^{(l)})} \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(l)}) \tag{3.4}$$

Donde $\mathbf{z}^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$, sin embargo, este estimador de gradiente presenta una varianza muy alta y es poco práctico para nuestros propósitos.

3.1.2. El estimador SGVB y el algoritmo AEVB.

Hemos explicado en el capítulo anterior como resolver la inferencia variacional de campo medio mediante el ascenso de coordenadas, en esta sección se propone el autocodificador VB, (AEVB).

En el algoritmo AEVB hacemos que la inferencia y el aprendizaje sean especialmente eficiente utilizando el estimador SGVB para optimizar un modelo de reconocimiento que nos permite realizar inferencia posterior aproximada muy eficiente utilizando un simple muestreo previo, que a su vez nos permite aprender eficien-

temente los parámetros del modelo. Inicialmente introduciremos el estimador de límite inferior, (*lower bound*). Para ello, asumiremos una distribución posterior conocida $q_\phi(\mathbf{z}|\mathbf{x})$.

Nota 3.1. Aunque asumamos por conocida la distribución $q_\phi(\mathbf{z}|\mathbf{x})$, es importante tener en cuenta que el algoritmo explicado en esta sección se podría aplicar también a la misma distribución $q_\phi(\mathbf{z})$. El método bayesiano variacional completo está detallado en la sección 3.1.2.2.

En esta sección reparametrizaremos la variable aleatoria continua $\mathbf{z}^* \sim q_\phi(\mathbf{z}|\mathbf{x})$ mediante el método conocido como truco de reparametrización, a través del cual expresaremos la variable aleatoria $\mathbf{z}$ como una variable determinista $\mathbf{z} = g_\phi(\epsilon, \mathbf{x})$, donde ϵ es una variable auxiliar con una distribución marginal $p(\epsilon)$, y $g_\phi(.)$ es una función vectorial parametrizada por ϕ.

Antes de seguir con el problema general, explicaremos en qué consiste el truco de reparametrización.

Esta reparametrización es útil ya que nos permite reescribir la esperanza con respecto a $q_\phi(\mathbf{z}|\mathbf{x})$ de tal manera que la estimación de Monte Carlo de la esperanza es diferenciable con respecto a ϕ.

Dado el muestreo determinista $\mathbf{z} = g_\phi(\epsilon, \mathbf{x})$ podemos escribir que $q_\phi(\mathbf{z}|\mathbf{x})d\mathbf{z} = q_\phi(\mathbf{z}|\mathbf{x}) \prod_i d\mathbf{z}_i$, lo que es igual a $p(\epsilon) \prod_i d\epsilon_i$. Por lo tanto, siendo $f(\mathbf{z})$ una función real suficientemente regular dependiente de $\mathbf{z}$, tenemos que:

$$\int q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} = \int p(\epsilon) f(\mathbf{z}) d\epsilon = \int p(\epsilon) f(g_\phi(\epsilon, \mathbf{x})) d\epsilon, \tag{3.5}$$

observamos de esta última ecuación que podemos construir el estimador diferencial

$$\int q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_\phi(\mathbf{x}, \epsilon^{(l)})),$$

donde $\epsilon^{(l)} \sim p(\epsilon)$.

Para comprender la idea explicada aplicamos el truco de parametrización para el caso de una distribución normal univariante. Sea $\mathbf{z} \sim p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu, \sigma^2)$. En este caso una parametrización válida sería $\mathbf{z} = \mu + \sigma\epsilon$, siendo epsilon la variable auxiliar

con distribución normal, $\epsilon \sim \mathcal{N}(0,1)$. De esta manera tendríamos

$$\mathbb{E}_{\mathcal{N}(\mathbf{z};\mu,\sigma^2)}\left\{f(\mathbf{z})\right\} = \mathbb{E}_{\mathcal{N}(\epsilon;0,1)}\left\{f(\mu+\sigma\epsilon)\right\} \simeq \frac{1}{L}\sum_{l=1}^{L} f(\mu+\sigma\epsilon^{(l)}),$$

donde $\epsilon \sim \mathcal{N}(0,1)$.

Mediante el truco de parametrización hemos establecido la distribución $p(\epsilon)$ y la función $g_\phi(\epsilon,\mathbf{x})$ convenientes reparametrizando la variable aleatoria $\mathbf{z}^* \sim q_\phi(\mathbf{z}|\mathbf{x})$.

$$\mathbf{z}^* = g_\phi(\epsilon,\mathbf{x}) \;\; ; \;\; \epsilon \sim p(\epsilon).$$

Recordamos que podemos usar la estimación de Monte Carlo para hallar la esperanza de una función $f(\mathbf{z})$ cualquiera con respecto a $q_\phi(\mathbf{z}|\mathbf{x})$.

$$\begin{aligned}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left\{f(\mathbf{z})\right\} &= \mathbb{E}_{p(\epsilon)}\left\{f(g_\phi(\epsilon,\mathbf{x}^{(i)}))\right\} = \\ &\simeq \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\epsilon^{(l)},\mathbf{x}^{(i)})),\end{aligned} \tag{3.6}$$

donde $\epsilon^{(l)} \sim p(\epsilon)$.

Aplicaremos esta técnica al límite inferior variacional 3.4 (ELBO) obteniendo nuestro estimador bayesiano variacional del gradiente, (SGVD), es decir, obtendríamos $ELBO^*(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) \simeq ELBO(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)})$ de tal manera que:

$$ELBO^*(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) = \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)},\mathbf{z}^{(i,l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(i,l)}|\mathbf{x}^i), \tag{3.7}$$

donde $\mathbf{z}^{(i,l)} = g_\phi(\epsilon^{(i,l)},\mathbf{x}^{(i)})$ y $\epsilon \sim p(\epsilon)$.

Ya tenemos el estimador SGVD que buscamos, antes de aplicar el algoritmo AEVB para resolver el problema, presentaremos otra variante del estimador que se puede usar.

En muchas ocasiones, la KL divergencia $KL(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}))$ puede ser calculada analíticamente, de tal manera que únicamente $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left\{\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right\}$ requeriría estimación por muestreo. En este caso la divergencia KL se puede interpretar como

un regularizador ϕ, provocando que la distribución posterior aproximada sea próxima a la distribución a priori $p_{\boldsymbol{\theta}}(\mathbf{z})$.

Con este proceso obtendríamos una segunda versión del estimador SGVB, $ELBO^B$, con el que conseguimos una menor vrianza que la que obteníamos con el estimador $ELBO^*$.

$$ELBO^B(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = -KL(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}), \tag{3.8}$$

donde $\mathbf{z}^{(i,l)} = g_\phi(\epsilon^{(i,l)}, \mathbf{x}^{(i)})$ y $\epsilon^{(l)} \sim p(\epsilon)$.

Supongamos ahora que tenemos un conjunto inicial $\boldsymbol{X}$ con N puntos, podemos construir un estimador de la probabilidad marginal del límite inferior del conjunto de datos completo, basado en minilotes:

$$ELBO^B(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) \simeq ELBO^M(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = \frac{N}{M}\sum_{i=1}^{M} ELBO^*(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) \tag{3.9}$$

Donde el minilote de puntos $\boldsymbol{X}^M = \{\mathbf{x}^{(i)}\}_{i=1}^M$ es un muestreo aleatorio de M puntos del conjunto total con N puntos $\mathbf{x}$.

Se pueden hallar las derivadas $\nabla_{\boldsymbol{\theta},\phi} ELBO^*(\boldsymbol{\theta}, \mathbf{x}^M)$, y los gradientes resultantes se pueden utilizar junto con métodos de optimización estocástica como SGD o Adagrad (John D. et al. [8]).

Finalmente el algoritmo AEVB nos quedaría de la siguiente manera:

Algorithm 3 Algoritmo AEVB versión minilotes del autocodificador VB

Input: Parámetros iniciales $\boldsymbol{\theta}$, ϕ.

Output: Los parámetros $\boldsymbol{\theta}$ y ϕ actualizados.

for $\boldsymbol{\theta}$ y ϕ no converja **do**

Obtenemos $\boldsymbol{X}^M$. El muestreo aleatorio de M datos extraído del conjunto inicial de datos.

Obtenemos las muestras de la variable aleatoria auxiliar ϵ de la distribución $p(\epsilon)$.

Hacemos el gradiente del estimador ELBO, $\nabla_{\boldsymbol{\theta},\phi} ELBO^*(\boldsymbol{\theta}, \phi; \mathbf{x}^m, \epsilon)$ obteniendo la función g.

Actualizamos los parámetros θ y ϕ usando la función g.

end for

return θ, ϕ.

Observando la ecuación 3.8 nos damos cuenta de la relación con los autocodificadores.

El primer término de la ecuación, (la divergencia KL), actúa como regularizador mientras que el segundo término sería el error de reconstrucción negativo esperado.

La función g_ϕ se elige de tal manera que proyecta los datos $\mathbf{x}^{(i)}$ y un vector aleatorio de ruido $\epsilon^{(l)}$ a una muestra de la distribución posterior aproximada para ese conjunto de datos. Es decir, $\mathbf{z}^{(i,l)} = g_\phi(\epsilon^{(l)}, \mathbf{x}^{(i)})$ donde $\mathbf{z}^{(i,l)} \simeq q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$.

Posteriormente $\mathbf{z}^{(i,l)}$ será el input de la función $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$, que es igual a la densidad de probabilidad del punto de los datos $\mathbf{x}^{(i)}$ del modelo generativo. Este término se le conoce como el error negativo de reconstrucción del autocodificador.

3.1.2.1. Cálculo analítico de la divergencia KL.

Se ha mencionado en el apartado anterior la posibilidad de hallar analíticamente la divergencia $KL(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}))$. Vamos a ver en este apartado como hacerlo para el caso más común, el de la distribución normal.

Asumiremos que $p_{\boldsymbol{\theta}}(\mathbf{z}) = N(0, \mathbf{I})$ y $q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$ siguen distribuciones normales. Sea J la dimensión de $\mathbf{z}$. Sea μ y σ los vectores de medias y desviaciones típicas evaluados en el punto i, y sea μ_j y σ_j la componente j-ésima de esos vectores. Partimos de:

$$-KL(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) = \int q_\theta(\mathbf{z})(\log p_\theta(\mathbf{z}) - \log q_\theta(\mathbf{z}))d\mathbf{z}. \tag{3.10}$$

Calculamos la integral por separado:

$$\begin{aligned}\int q_\theta(\mathbf{z}) \log p_\theta(\mathbf{z}) d\mathbf{z} &= \int N(\mathbf{z}; \mu, \sigma^2) \log N(\mathbf{z}; \mathbf{0}, \mathbf{I}) d\mathbf{z} \\ &= -\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2).\end{aligned} \tag{3.11}$$

Por otro lado

$$\begin{aligned}\int q_\theta(\mathbf{z}) \log q_\theta(\mathbf{z}) d\mathbf{z} &= \int N(\mathbf{z}; \mu, \sigma^2) \log N(\mathbf{z}; \mu, \sigma^2) d\mathbf{z} \\ &= -\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(1 + \log \sigma_j^2),\end{aligned} \tag{3.12}$$

juntando ambas expresiones finalmente nos quedaría:

$$\begin{aligned}-KL(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) &= \int q_\theta(\mathbf{z})(\log p_\theta(\mathbf{z}) - \log q_\theta(\mathbf{z}))d\mathbf{z} \\ &= \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2).\end{aligned} \tag{3.13}$$

Obteniendo la expresión buscada.

3.1.2.2. El método Bayesiano Variacional Completo.

Como se ha mencionado al principio de esta sección, es posible realizar una inferencia variacional tanto en los parámetros $\boldsymbol{\theta}$ como en las variables latentes $\mathbf{z}$, en lugar de solo las variables latentes como hicimos en este trabajo. En este apartado derivaremos nuestro estimador para ese caso.

Sea $p_{\boldsymbol{\alpha}}(\boldsymbol{\theta})$ una distribución previa para los parámetros introducidos en el párra-

fo anterior, parametrizada por α. La probabilidad marginal puede ser escrita como:

$$\log p_{\boldsymbol{\alpha}}(\boldsymbol{X}) = KL(q_{\phi}(\boldsymbol{\theta})||p_{\alpha}(\boldsymbol{\theta}|\boldsymbol{X})) + ELBO(\phi, \boldsymbol{X}), \tag{3.14}$$

siendo como hasta ahora tanto $\boldsymbol{X}$ el conjunto de datos iniciales y $ELBO(\phi, \mathbf{x})$ el límite inferior variacional de la probabilidad marginal:

$$ELBO(\phi, \boldsymbol{X}) = \int q_{\phi}(\boldsymbol{\theta})(\log p_{\boldsymbol{\theta}}(\boldsymbol{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\phi}(\boldsymbol{\theta}))d\boldsymbol{\theta}. \tag{3.15}$$

Observamos que se trata efectivamente de un límite inferior, ya que la divergencia KL no es negativa.. El términolog $p_{\boldsymbol{\theta}}(\mathbf{x})$ está compuesto de una suma sobre las probabilidades marginales de los puntos de datos individuales, $\log p_{\boldsymbol{\theta}}(\boldsymbol{X}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$, el cual reescribimos como sigue:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = KL(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\alpha}(\mathbf{z}|\mathbf{x}^{(i)}) + ELBO(\boldsymbol{\theta}, \phi, \mathbf{x}^{(i)}), \tag{3.16}$$

siendo en este caso $ELBO(\boldsymbol{\theta}, \phi, \mathbf{x}^{(i)})$ el límite inferior variacional de la probabilidad marginal del punto de datos i:

$$ELBO(\boldsymbol{\theta}, \phi, \mathbf{x}^{(i)}) = \int q_{\phi}(\mathbf{z}|\mathbf{x})(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x}))d\mathbf{z}. \tag{3.17}$$

Las esperanzas en el lado derecho de las ecuaciones 3.15 y 3.17 pueden escribirse obviamente como una suma de tres esperanzas separadas, de las cuales la segunda y la tercera componente pueden a veces resolverse analíticamente, por ejemplo cuando tanto $p_{\boldsymbol{\theta}}(\mathbf{x})$ como $q_{\phi}(\mathbf{z}|\mathbf{x})$ son normales, (siendo este el caso de mayor relevancia).

Por razones de generalidad, supondremos aquí que cada una de estas esperanzas es intratable.

Bajo ciertas condiciones para las distribuciones posteriores aproximadas elegidas $q_{\phi}(\boldsymbol{\theta})$ y $q_{\phi}(\mathbf{z}|\mathbf{x})$ podemos reparametrizar las muestras condicionales $\mathbf{z}^{*} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$ como

$$\mathbf{z}^{*} = q_{\phi}(\epsilon, \mathbf{x}) \qquad \epsilon \sim p(\epsilon), \tag{3.18}$$

donde elegimos la distribución previa $p(\epsilon)$ y una función $g_\phi(\epsilon, \mathbf{x})$ tal que se cumple lo siguiente:

$$\begin{aligned} ELBO(\boldsymbol{\theta}, \phi, \mathbf{x}^{(i)}) &= \int q_\phi(\mathbf{z}|\mathbf{x})(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}))d\mathbf{z} \\ &= \int p(\epsilon) \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right) |_{\mathbf{z}=g_\phi(\epsilon, \mathbf{x}^{(i)})} d\epsilon. \end{aligned} \tag{3.19}$$

Y lo mismo se puede hacer para la distribución aproximada posterior $q_\phi(\boldsymbol{\theta})$:

$$\boldsymbol{\theta}^* = h_\phi(\zeta) \qquad \zeta \sim p(\zeta). \tag{3.20}$$

En la cual elegimos como en el anterior la distribución previa $p(\zeta)$ y la función $h_\phi(\zeta)$ tal que cumplen lo siguiente:

$$\begin{aligned} ELBO(\phi, \mathbf{x}) &= \int q_\phi(\boldsymbol{\theta})(\log p_{\boldsymbol{\theta}}(\mathbf{x}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_\phi(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &= \int p(\epsilon) \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_\phi(\boldsymbol{\theta})\right) |_{\boldsymbol{\theta}=h_\phi(\zeta)} d\zeta. \end{aligned} \tag{3.21}$$

Por comodidad en la notación llamaremos a $f_\phi(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})$ como :

$$f_\phi(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta}) = N \cdot \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_\phi(\boldsymbol{\theta}). \tag{3.22}$$

Usando las ecuaciones 3.19 y 3.21, la estimación de Monte Carlo del límite inferior variacional, dado el punto $\mathbf{x}^{(i)}$ es:

$$ELBO(\phi, \mathbf{x}) \approx \frac{1}{L} \sum_{l=1}^{L} f_\phi(\mathbf{x}^{(l)}, g_\phi(\epsilon^{(l)}, x^{(l)}), h_\phi(\zeta^{(l)})), \tag{3.23}$$

donde $\epsilon^{(l)} \sim p(\epsilon)$ y $\zeta^{(l)} \sim p(\zeta)$.

Se observa que el estimador sólo depende de las muestras de $p(\epsilon)$ y $p(\zeta)$ las cuales no dependen de ϕ, por lo que el estimador puede diferenciarse con respecto a ϕ.

3.2. Autocodificadores Variacionales, VAEs.

Continuaremos explicando cómo usar una red neuronal en el codificador probabilístico $q_\phi(\mathbf{z}|\mathbf{x})$, donde los parámetros $\boldsymbol{\theta}$ y ϕ son optimizados mediante el algoritmo AEVB, este es el caso al que queríamos llegar, los *Autocodificadores Variacionaless*, VAEs.

Asumiremos que la densidad de probabilidad sobre las variables latentes será la distribución normal normalizada, es decir centrada y con matriz de covarianzas la identidad, $p_{\boldsymbol{\theta}}(\mathbf{z}) = N(\mathbf{z}; \mathbf{0}, \mathbf{I})$.

Observamos que esta densidad de probabilidad carece de parámetros, además, establecemos que la distribución $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ será una distribución normal multivariante para un caso de datos reales, o una Bernoulli para conjuntos de datos binarios, cuyos parámetros de distribución se calculan a partir de $\mathbf{z}$ con un MLP, red neuronal perceptron multicapa, lo que se explicará en el siguiente apartado.

También observamos que en este caso la distribución verdadera a posteriori $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ no se puede calcular explícitamente, por lo que haremos en este punto una pequeña simplificación.

Asumiremos que se corresponde a una distribución aproximadamente normal con una covarianza aproximadamente diagonal, y por lo tanto, debido a la libertad en la forma de $q_\phi(\mathbf{z}|\mathbf{x})$ podemos establecer que:

$$q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \log N(\mathbf{z}; \mu^{(i)}, \sigma^{2(i)}\mathbf{I}), \tag{3.24}$$

donde la media y al desviación estándar $\mu^{(i)}$ y $\sigma^{(i)}$ los obtendremos mediante el código mencionado anteriormente MLP.

Muestreamos de la distribución posterior $\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$ usando

$$\mathbf{z}^{(i,l)} = g_\phi(\mathbf{x}^{(i)}, \epsilon^{(l)}) = \mu^{(i)} + \sigma^{(i)} * \epsilon^{(l)} \;\; con \;\; \epsilon^{(l)} \sim N(\mathbf{0}, I).$$

En este modelo tenemos que ambas distribuciones, a priori y posteriori, $p_{\boldsymbol{\theta}}(\mathbf{z})$ y $q_\phi(\mathbf{z}|\mathbf{x})$ son distribuciones normales, por lo tanto, usaremos el estimador 3.8, en el cual, la divergencia KL se halla explítitamente mediante el método explciado en la sección 3.1.2. Por lo que finalmente el estimador para este modelo y el punto de la

muestra $\mathbf{x}^{(i)}$ sería:

$$\begin{aligned} ELBO(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) &\simeq -KL(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}) \\ &= \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right) + \\ &+ \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}), \end{aligned} \tag{3.25}$$

donde $\mathbf{z}^{(i,l)} = \mu^{(i)} + \sigma^{(i)} * \epsilon^{(l)}$ y $\epsilon^{(l)} \sim N(0, \mathbf{I})$.

Finalmente, el decodificador $p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$ será una distribución de Bernoulli o Gaussiana MLP, dependiendo del tipo de datos que estemos modelando. Los cuales se explican a continuación.

3.2.1. MLP como codificadores y decodificadores probabilísticos.

En este apartado se darán por conocidos los conocimientos sobre redes neuronales elementales que se pueden consultar en (R. Rojas [9]).

En los autocodificadores variacionales, las redes neuronales se utilizan como codificadores y decodificadores probabilísticos.

Existen infinidad de posibilidades de codificadores y decodificadores, dependiendo del tipo de datos y del modelo. En el ejemplo tratado en esta sección tratamos redes neuronales perceptrones multicapa, MLP. Para el codificador utilizamos un MLP con salida Gaussiana, mientras que para el decodificador utilizamos MLP con salidas Gaussianas o Bernoulli, dependiendo del tipo de datos. Explicaremos brevemente los siguientes casos:

-MLP Bernoulli como decofidicador: En este caso la distribución $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ será una distribución de Bernoulli multivariante cuyas probabilidades se calculan a partir de

la variable latente $\mathbf{z}$ con una red neuronal que posee únicamente una capa oculta:

$$\log p(\mathbf{x}|\mathbf{z}) = \sum_{i=1}^{D} \mathbf{x}_i \log y_i + (1 - \mathbf{x}_i) \cdot \log(1 - y_i),$$

donde $\mathbf{y} = f_\sigma(\mathbf{W}_2 \tanh(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2)$, siendo la función f_σ la función de activación sigmoidea, y $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$ son los pesos y sesgos correspondientes a la red perceptron multicapa correspondiente.

-MLP Gaussiano como decofidicador y codificador: El codificador y decodificador será una distribución probabilística normal multivariante con una estructura de covarianza diagonal, por lo tanto

$$\log p(\mathbf{x}|\mathbf{z}) = \log N(\mathbf{x}; \mu, \sigma^2 \cdot \mathbf{I}),$$

donde $\mu = \mathbf{W}_4\mathbf{h} + \mathbf{b}_4$, $\log \sigma^2 = \mathbf{W}_5\mathbf{h} + \mathbf{b}_5$ y $\mathbf{h} = \tanh(\mathbf{W}_3\mathbf{z} + \mathbf{b}_3)$. Siendo $\{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$ los pesos y sesgos correspondientes a la red MLP, y parten de $\boldsymbol{\theta}$ cuando se usa como decodificador, como en el caso anterior.

Cuando esta red se usa como codificador $q_\phi(\mathbf{z}|\mathbf{x})$, los papeles de $\mathbf{z}$ y $\mathbf{x}$ se intercambian, siendo los pesos y los sesgos los parámetros variacionales ϕ.

3.3. Trabajo Empírico.

En esta sección se mostrarán los resultados y experimentos expuestos en *Auto-Encoding variatonal Bayes*, Diederik P. and Max W. [18].

En ellos se entrenaron modelos generativos de imágenes de los conjuntos de datos MNIST y Frey Face[1] y se comparan los algoritmos de aprendizaje en términos de la cota inferior variacional y la probabilidad marginal estimada.

Antes de ello, vamos a presentar dos apartados que se usan en el desarrollo empírico.

[1]http://www.cs.nyu.edu/ roweis/data.html

3.3.1. Estimador de probabilidad marginal.

En este apartado derivaremos el siguiente estimador de la distribución de probabilidad marginal el cual produce buenas estimaciones siempre que la dimensionalidad del espacio muestreado sea baja, (menos de 5 dimensiones), y se tomen suficientes muestras.

Sea $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ el modelo generativo del que estamos muestreando. Para un punto de datos $\mathbf{x}^{(i)}$ queremos estimar la probabilidad marginal $p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$.

El proceso de estimación consta de tres etapas:

1. Muestreamos L valores $\{\mathbf{z}^{(l)}\}$ de la distribución a posteriori usando MCMC basado en el gradiente, como podría ser el Monte Carlo Híbrido, usando $\nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.
2. Ajustamos un estimador $q(\mathbf{z})$ a las muestras $\{\mathbf{z}^{(l)}\}$.
3. De nuevo, se muestrean L nuevos valores de la distribución posterior. Introducimos estas muestras, así como la $q(\mathbf{z})$, en el siguiente estimador:

$$p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \approx \left(\frac{1}{L} \sum_{l=1}^{L} \frac{q(\mathbf{z}^{(l)})}{p_{\boldsymbol{\theta}}(\mathbf{z}) p(\mathbf{x}^{(i)}|\mathbf{z}^{(l)})} \right)^{-1} \qquad donde \qquad \mathbf{z}^{(l)} \sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)}).$$

Derivando el estimador:

$$\begin{aligned} \frac{1}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} &= \frac{\int q(\mathbf{z}) d\mathbf{z}}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} = \frac{\int q(\mathbf{z} \dfrac{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})} d\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} \\ &= \int \frac{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} \cdot \frac{q(\boldsymbol{\theta})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})} d\mathbf{z} \\ &= \int p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)}) \cdot \frac{q(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})} \\ &\approx \frac{1}{L} \sum_{l=1}^{L} \frac{q(\mathbf{z}^{(l)})}{p_{\boldsymbol{\theta}}(\mathbf{z}) p(\mathbf{x}^{(i)}|\mathbf{z}^{(l)})} \qquad donde \qquad \mathbf{z}^{(l)} \sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)}). \end{aligned} \tag{3.26}$$

3.3.2. Monte Carlo EM.

El algoritmo EM de Monte Carlo no emplea un codificador, sino que muestrea a partir de la distribución a posteriori de las variables latentes utilizando gradientes de esta distribución calculados con $\nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$. El procedimiento EM de Montecarlo consiste en 10 pasos de salto con un tamaño de paso ajustado automáticamente de manera que la tasa de aceptación fuera del 90 %, seguido de 5 pasos de actualización de pesos utilizando la muestra adquirida. Para todos los algoritmos, los parámetros se actualizan utilizando los tamaños de paso de Adagrad [20].

La probabilidad marginal se estimó con los primeros 1000 puntos de datos de los conjuntos de entrenamiento y prueba, para cada punto de datos se muestrean 50 valores de la distribución a posteriori de las variables latentes utilizando el método de Monte Carlo Híbrido, Simon D. et al. [21], con 4 pasos de salto.

3.3.3. Desarrollo experimental.

Se utilizó el modelo generativo, (codificador), y la aproximación variacional, (decodificador), de la sección 3.2, donde el codificador y el decodificador descritos tienen el mismo número de capas ocultas.

Como los datos de Frey son continuos, se usó un decodificador con salidas de distribución normal, idéntico al codificador, excepto que las medias se limitaron al intervalo $(0, 1)$ utilizando una función de activación sigmoidal en la salida del decodificador.

Los parámetros se actualizan mediante el ascenso por gradiente estocástico, donde los gradientes se calculan diferenciando el estimador de límite inferior $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} ELBO$ mediante el algoritmo presentado en la sección 3.1.2, más un pequeño término de decaimiento del peso correspondiente a la distribución a previa $p(\boldsymbol{\theta}) = \mathcal{N}(0, \boldsymbol{I})$.

De esta manera se comparó el rendimiento de AEVB con el algoritmo wake-sleep presentado en Geoffrey E. et al. [19]. Para ello se empleó el mismo codificador para el algoritmo wake-sleep y el autocodificador variacional. Todos los parámetros, tanto variacionales como generativos, se inicializaron mediante un muestreo aleatorio procedente de una distribución $\mathcal{N}(0, 0.01)$, y se optimizaron conjuntamente de for-

ma estocástica utilizando el criterio MAP.

Los parámetros globales de tamaño de paso de Adagrad se eligieron entre $\{0.01, 0.02, 0.1\}$ basándose en el rendimiento del conjunto de entrenamiento en las primeras iteraciones. Se utilizaron minilotes de tamaño $M = 100$, con $L = 1$ muestras por punto del conjunto de datos.

Probabilidad de límite inferior: Se entrenó los modelos generativos, (decodificadores), y los correspondientes codificadores con 500 unidades ocultas en el caso de MNIST, y 200 unidades ocultas en el caso de datos Frey Face, para evitar el sobreajuste, ya que se trataba de un conjunto de datos considerablemente más pequeño. El rendimiento relativo de los diferentes algoritmos no fue muy sensible a estas elecciones. La siguiente figura muestra los resultados que se obtuvieron comparando los diferentes límites inferiores.

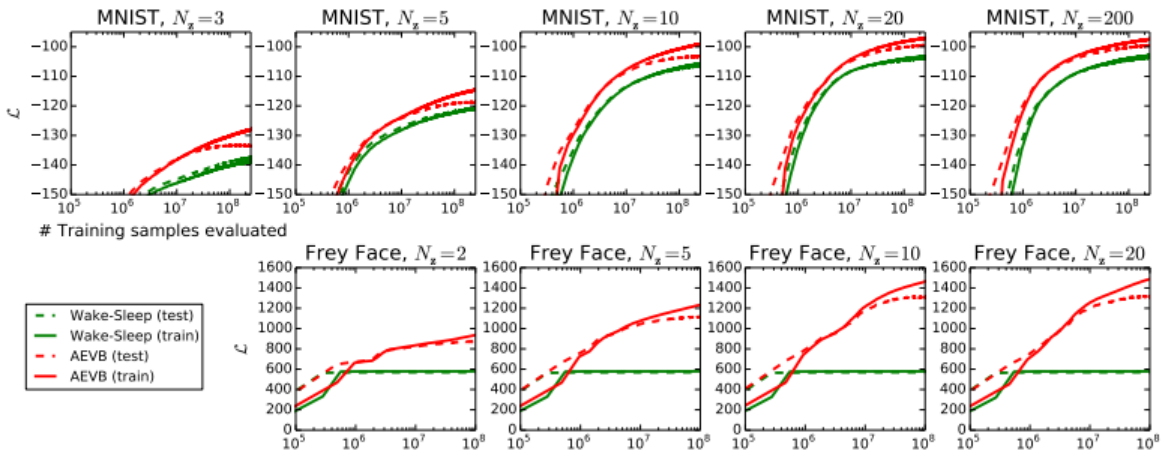

Figura 3.1: Comparación de los resultados obtenidos con el algoritmo AEVB y con el algoritmo wake-sleep, en términos de optimización del límite inferior, para diferente dimensionalidad del espacio latente ($N\mathbf{z}$).
El método aplicado converge considerablemente más rápido y se observa como alcanza una mejor solución en todos los experimentos.
Eje vertical: la media estimada del límite inferior variacional por punto de datos. La varianza del estimador era pequeña (< 1) y se omite.
Eje horizontal: cantidad de puntos de entrenamiento evaluados. El cálculo tardó entre 20 y 40 minutos por cada millón de muestras de entrenamiento con una CPU Intel xeon que funcionaba a 40 GFLOPS efectivos.
Source: Auto-Encoding Variatonal Bayes, Diederik P. and Max W. [18].

Se observó que las variables latentes superfluas no dieron lugar a un sobreajuste,lo cual llama la tención pero se debe por la naturaleza regularizadora del límite variacional.

Estimador de la probabilidad marginal: Para un espacio latente de muy baja dimensión es posible estimar la máxima verosimilitud marginal de los modelos generativos aprendidos utilizando un estimador MCMC, según se ha explicado en la sección 3.3.1 y 3.3.2.

Para el codificador y el decodificador se utilizaron redes neuronales, con 100 unidades ocultas y 3 variables latentes. Se observó que para un espacio latente de mayor dimensión, las estimaciones se volvieron poco fiables.

De nuevo, se utilizó el conjunto de datos MNIST. Los métodos AEVB y Wake-Sleep se compararon con el Monte Carlo EM, (MCEM), y con un Monte Carlo Híbrido, (HMC), Simon D. et al. [21].

Se comparó la velocidad de convergencia de los tres algoritmos, para un conjunto de entrenamiento de tamaño pequeño y grande, cuyos resultados se muestran en la figura siguiente:

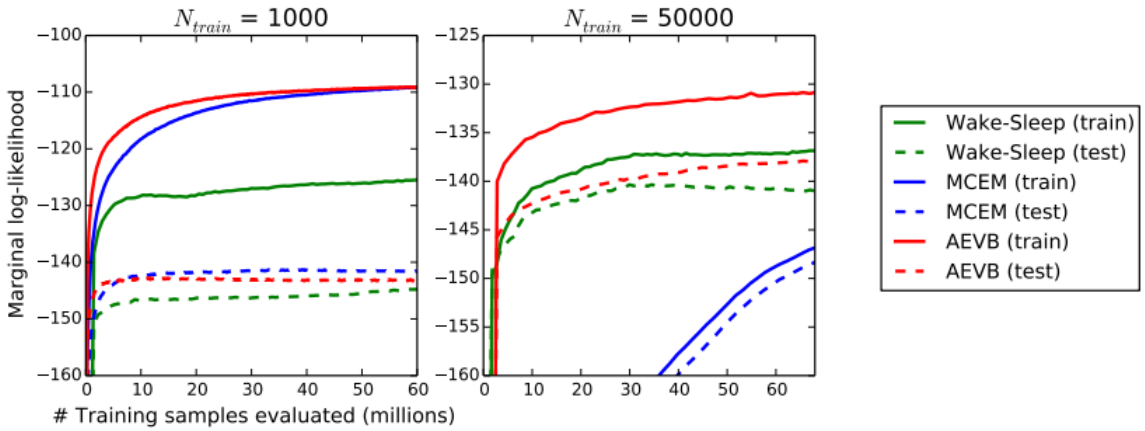

Figura 3.2: Comparación del algoritmo AEVB con el algoritmo wake-sleep y Monte Carlo EM, en términos de la probabilidad marginal estimada, para un número diferente de puntos de entrenamiento.
Monte Carlo EM no es un algoritmo en línea, y no puede aplicarse eficientemente el conjunto de datos MNIST completo a diferencia del algoritmo AEVB y del método wake-sleep.
Source: Auto-Encoding Variatonal Bayes, Diederik P. and Max W. [18].

3.3.4. Conclusiones y futuros trabajos.

En este último capítulo, hemos introducido un nuevo estimador del límite inferior variacional, el Gradiente Estocástico VB (SGVB), para una inferencia aproximada eficiente con variables latentes continuas.

El estimador propuesto puede diferenciarse y optimizarse directamente utilizando métodos estándar de gradiente estocástico. Para el caso de conjuntos de datos *i.i.d*. y variables latentes continuas por punto de datos hemos introducimos un algoritmo eficiente para la inferencia y el aprendizaje, el autocodificador VB (AEVB), que aprende un modelo de inferencia aproximado utilizando el estimador SGVB, las ventajas que proporciona este método se han podido observar y comentar en los resultados anteriores.

Dado que el estimador SGVB y el algoritmo AEVB pueden aplicarse a casi cualquier problema de inferencia y aprendizaje con variables latentes continuas, hay muchas posibles direcciones de investigación futuras, entre las cuales mencionamos las siguientes:

- Aprender arquitecturas generativas jerárquicas con redes neuronales profundas, como las redes convolucionales, utilizadas para los codificadores y decodificadores entrenados conjuntamente con AEVB.
- Modelos de series temporales, como las redes bayesianas dinámicas.
- Aplicación de SGVB a los parámetros globales.
- Modelos supervisados con variables latentes, útiles para aprender distribuciones de ruido complicadas.

Con este capítulo finalizamos este trabajo de fin de grado. Hemos comprobado como mediante el uso de estimadores eficientes, los autocodificadores variacionales representan uno de los mejores candidatos en los modelos generativos, solucionando los problemas de sobreajuste que presentaban autocodificadores más sencillos.

BIBLIOGRAFÍA

[1] Shai Shalev-Shwartz,Shai Ben-David , *Understanding Machine Learning*, Cambridge University Press, 2014. 2, 5

[2] Jordan, M. I., Ghahramani, Z., Jaakkola, T., Saul, L.Saul, L., *Introduction to variational methods for graphical models. Machine Learning*, 37:183–233, 1999 13

[3] Wainwright, M., Jordan, M. I. , *Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008. 15

[4] Kingma, Diederik P., Welling, Max , *Auto-Encoding Variational Bayes*, 2014. 15

[5] Dempster, A., Laird, N. , Rubin, D., *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, Series B, 39:1–38, 1977. 15

[6] Bishop, C., *Pattern Recognition and Machine Learning*, Springer New York, 2006. 17

[7] David M. Blei, Michael I. Jordan, John W. Paisley *Variational bayesian inference with stochastic search*, In Proceedings of the 29th International Conference on Machine Learning (ICML-12), pages 1367–1374, 2012. 36, 37

[8] John Duchi, Elad Hazan, Yoram Singer *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12:2121– 2159, 2010. 40

[9] R. Rojas, *A Systematic Introduction: Neural Networks*, Springer-Verlag, 1996. 46

[10] Murphy, K. P., *Machine learning: a probabilistic perspective*, p. 35. MIT press. ISBN 0262018020, 2012. 20

[11] David M. Blei, Alp Kucukelbir, Jon D. McAuliffe, *Variational Inference: A Review for Statisticians*, Columbia University and University of California, Berkeley, May 11, 2018. 2, 19, 27, 28

[12] You, C., Ormerod, J. and Muller, S., *On variational Bayes estimation and variational information criteria for linear regression models*, Australian New Zealand Journal of Statistics, 56(1):73–87, 2014. 29

[13] Hall, P., Ormerod, J. and Wand, M., *Theory of Gaussian variational approximation for a Poisson mixed model*, Statistica Sinica, 21:369–389, 2011. 29

[14] Celisse, A., Daudin, J. and Pierre, L., *Consistency of maximum-likelihood and variational estimators in the stochastic block model*, Electronic Journal of Statistics, 6:1847–1899, 2012. 29

[15] Westling, T., and McCormick, T. H., *Establishing consistency and improving uncertainty estimates of variational inference through M-estimation*, arXiv:1510.08151, 2015. 29

[16] Wang, B. , and Titterington, D., *Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model*, Bayesian Analysis, 1:625–650, 2006. 29

[17] Minka, T. P. , *Expectation propagation for approximate Bayesian inference*, In Uncertainty in Artificial Intelligence, 2001. 30

[18] Diederik P., Max W., *Auto-Encoding Variatonal Bayes*, Machine Learning Group, In University van Amsterdam, 2014. 2, 47, 51, 52

[19] Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, Radford M Neal, *The wakesleep algorithm for unsupervised neural networks*, SCIENCE, pages 1158–1158, 1995. 49

[20] John Duchi, Elad Hazan, Yoram Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12:2121–2159, 2010. 49

[21] Simon Duane, Anthony D. Kennedy, Brian J. Pendleton, Duncan Roweth, *Hybrid Monte Carlo*, Physics letters B, 195(2):216–222, 1987. 49, 52