Monografía sobre Aprendizaje Automático

Nicolás Caro

Departamento de Ingeniería Matemática, Universidad de Chile, Santiago.

ncaro@dim.uchile.cl

September 4, 2018

Contents

Preface

El presente trabajo reúne resultados referentes al aprendizaje automático, y busca servir como material de apoyo en el estudio de esta disciplina. En esta versión se estudian de manera autocontenida tópicos sobre modelos gráficos probabilísticos, métodos de inferencia aproximada y aprendizaje con kernels.

Nicolás Caro

CHAPTER 1

Modelos gráficos probabilísticos

Introducción

El eje central de este capitulo se basa en la búsqueda de una representación compacta, para distribuciones de probabilidad conjunta de la forma $p(\boldsymbol{x}|\boldsymbol{\theta})$. Esto, con la intención de realizar inferencia sobre variables y aprendizaje de parámetros de manera eficiente.

1.1. Modelos gráficos dirigidos

Toda distribución de probabilidad conjunta $p(\boldsymbol{x}) = p(x_1, x_2, \dots x_v)$ se puede representar de la forma:

$$p(\boldsymbol{x}) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \dots p(x_v \mid x_1, x_2, \dots, x_{v-1}) \tag{1.1}$$

El problema con esta expresión es la dificultad computacional subyacente al cálculo de distribuciones condicionales de la forma $p(x_t|x_1, \dots, x_{t-1})$ cuando el número de variables incidentes t aumenta.

No obstante, la representación (1.1) reduce su complejidad en presencia de **independencia condicional**.

En efecto, si se asume $x_{t+1} \perp x_1, \dots, x_{t-1} \mid x_t$. Es decir, las observaciones futuras x_{t+1} son independientes del pasado $x_1, \dots, x_{t-1}$, dado el estado presente x_t. La probabilidad conjunta se reduce entonces a:

$$p(\boldsymbol{x}) = p(x_1) \prod_{t=2}^{v} p(x_t|x_1, \dots, x_{t-1}) = p(x_1) \prod_{t=2}^{v} p(x_t|x_{t-1}) \tag{1.2}$$

De lo cual se obtiene una expresión más simple.

Modelar la independencia condicional entre las variables permite entonces reducir la complejidad de representación para la distribución conjunta. En particular, la elección tomada en (1.2) se conoce como **propiedad de Markov** de primer orden. En un contexto general, las relaciones de independencia condicional entre variables aleatorias de dimensión arbitraria, se modelan utilizando *diagramas de independencia* o **modelos gráficos**. Estos se valen de un grafo $G = (\mathcal{V}, \mathcal{E})$ [1] para representar mediante nodos $v = 1, \dots, \mathcal{V}$ las variables aleatorias del modelo, mientras que la presencia o ausencia de aristas entre estos nodos, permite modelar las relaciones de dependencia condicional subyacentes.

Una *red bayesiana* o **modelo gráfico dirigido** es un modelo gráfico probabilístico, cuyo grafo subyacente es un **grafo dirigido acíclico** (DAG por sus siglas en ingles). Todo DAG posee un *ordenamiento topológico*, es decir, los nodos de cualquier DAG pueden ser numerados de manera tal, que todo nodo padre posea una numeración inferior a sus nodos hijos. Esta característica permite enriquecer

[1] Conjunto consistente de $\mathcal{V} = \{1 \dots, V\}$ vértices (o nodos) y $\mathcal{E} = \{(s, t) : s, t \in \mathcal{V}\}$ aristas.

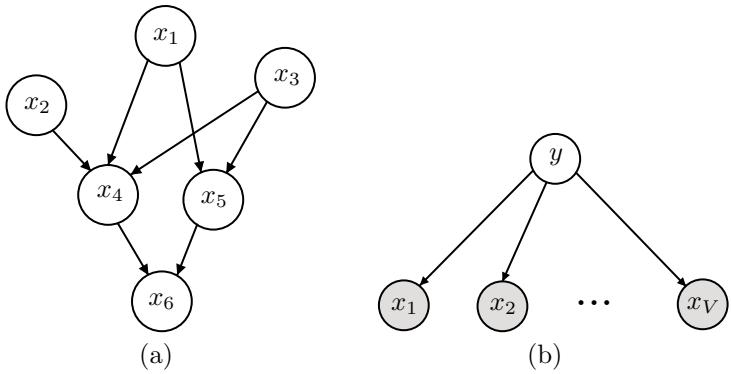

FIGURE 1. (a) Ejemplo de modelo gráfico dirigido. (b) Relaciones de dependencia condicional en el clasificador naive Bayes como un modelo gráfico dirigido, las variables aleatorias observadas se denotan por nodos grises.

la formulación de la propiedad de Markov (1.2), usando la estructura grafica como componente adicional. De esta forma, se puede formular la **propiedad ordenada de Markov** en modelos gráficos dirigidos:

$$x_s \perp \boldsymbol{x}_{pred(s)\setminus pa(s)} \mid \boldsymbol{x}_{pa(s)} \tag{1.3}$$

Es decir, un nodo x_s es independiente de aquellos predecesores, menores en orden topológico, a sus padres $\boldsymbol{x}_{pred(s)\setminus pa(s)}$, dados sus nodos padres $\boldsymbol{x}_{pa(s)}$. De manera equivalente, un nodo x_s solo depende de sus padres inmediatos $x_{pa(s)}$ y no de todos sus predecesores.

De esta forma, la probabilidad conjunta de un modelo gráfico dirigido, que cumple la propiedad ordenada de Markov, se puede descomponer de la forma:

$$p(\boldsymbol{x}) = \prod_{t=1}^{V} p(x_t|\boldsymbol{x}_{pa(t)}) \tag{1.4}$$

EJEMPLO 1.1 (Modelo grafico asociado a $p(\boldsymbol{x})$). *Si se estudia un modelo probabilístico, donde la probabilidad conjunta de las variables estudiadas $p(\boldsymbol{x})$ esta dada por:*

$$p(\boldsymbol{x}) = p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)p(x_5|x_1,x_3)p(x_6|x_4,x_5) \tag{1.5}$$

Entonces, un grafo dirigido asociado a tal factorización es el de la figura 1(a). Para construir dicho grafo, se consideran las relaciones de independencia condicional en la factorización (1.5), para luego establecer aristas $s \to t$ si la probabilidad condicional del nodo x_s depende de x_t. En este caso, no hay aristas incidentes hacia x_1, x_2 ni x_3. Por otra parte, se deben crear aristas desde x_1, x_2 y x_3 hacia x_4, desde x_1 y x_3 hacia x_5 y desde x_4, x_5 hacia x_6.

En general, es posible reconstruir la probabilidad conjunta subyacente a un modelo gráfico probabilístico conociendo el grafo y haciendo el proceso inverso al descrito anteriormente.

Con el fin de explorar las posibilidades de este tipo de modelos e introducir conceptos referentes a la notación de estos, se pasan a estudiar los siguientes ejemplos:

1.1.1. Naive Bayes. Dado un problema de clasificación de vectores $\boldsymbol{x} = (x_1, \ldots, x_V)$ en C clases. Es posible modelar las variables de decisión x_t como condicionalmente independientes dada la categoría de clasificación:

$$x_i \perp x_j \ | y = c, \ i \neq j \tag{1.6}$$

Si se usa este enfoque, se obtiene que la densidad condicional de clases toma la forma:

$$p(\boldsymbol{x} \ | y = c) = \prod_{t=1}^{V} p(x_t \ | y = c) \tag{1.7}$$

Al parametrizar las distribuciones de densidad condicional, es posible obtener un modelo de clasificación conocido como **clasificador naive Bayes**. La estructura de las relaciones de independencia inducidas por (1.6) se pueden expresar según (1.7) y el modelo gráfico dirigido de la figura 1(b).

1.1.2. Regresión polinomial. las variables aleatorias son el vector de coeficientes polinomiales $\boldsymbol{w}$ y los datos observados $\boldsymbol{y} = (y_1, \ldots, y_N)^T$. Adicionalmente, se parametriza el ruido del modelo a través de σ_ε^2 y la varianza de la distribución a priori [2] de $\boldsymbol{w}$ por σ_w^2. Finalmente, los datos de entrada se denotan por $\boldsymbol{x} = (x_1, \ldots, x_N)^T$.

La probabilidad conjunta de este modelo es el producto de la probabilidad a priori $p(\boldsymbol{w})$ con las distribuciones condicionales $p(y_i|\boldsymbol{w})$ para $i = 1, \ldots, N$:

$$p(\boldsymbol{y}, \boldsymbol{w}) = p(\boldsymbol{w}) \prod_{i=1}^{N} p(y_i|\boldsymbol{w}) \tag{1.8}$$

El grafo de tal factorización es similar al del clasificador naive Bayes 1(b). Para representarlo de manera compacta, se usa la notación de de placas o *plates*, en la figura 2(a) se muestra el grafo de (1.8) usando esta convención. Aquí N es la cantidad de nodos del modelo, de los cuales se muestra el representante y_i.

Si por otra parte, si se quiere estudiar la interacción de los parámetros en el modelo, es posible explicitarlos en la probabilidad conjunta para luego agregarlos al grafo:

$$p(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}, \sigma_\varepsilon^2, \sigma_w^2) = p(\boldsymbol{w}|\sigma_w^2) \prod_{i=1}^{N} p(y_i|\boldsymbol{w}, x_i, \sigma_\varepsilon^2) \tag{1.9}$$

La figura 2(b) muestra el grafo correspondiente a (1.9). Por convención, las variables deterministas se incluyen en el grafo como círculos pequeños, mientras que las variables aleatorias observadas se muestran como nodos grises, los nodos incoloros representan variables latentes o no observadas, finalmente las aristas, al igual que en los ejemplos anteriores, representan la dependencia condicional en la factorización de la probabilidad conjunta.

[2] Considerándose una distribución a priori, gaussiana y esférica sobre $\boldsymbol{w}$.

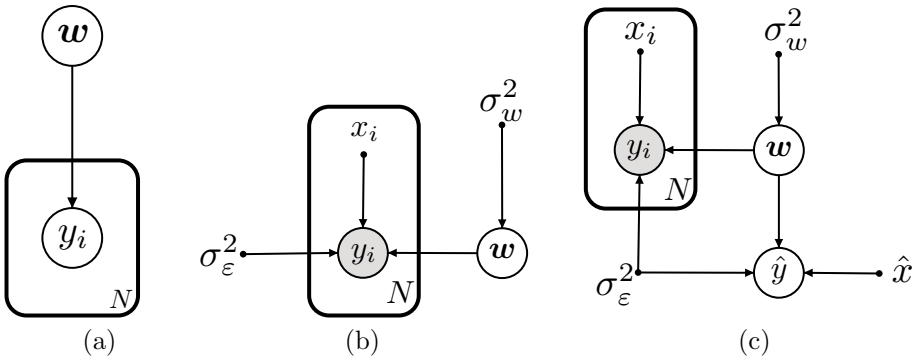

FIGURE 2. Modelo grafico dirigido para regresión polinomial usando notación de placas (o *plates*). En (a) se muestra el grafo correspondiente a (1.8). En (b) se añaden los parámetros deterministas y las variables aleatorias observadas. En (c) se añaden datos de entrada y predicciones.

Para realizar predicciones en datos nuevos $\hat{x}$, se desea encontrar la distribución de probabilidad para $\hat{y}$ condicionada a la información que ya se posee. Esta corresponde a:

$$p(\hat{y}, \boldsymbol{y}, \boldsymbol{w}|\hat{x}, \boldsymbol{x}, \sigma_w^2, \sigma_\varepsilon^2) = \left[\prod_{i=1}^{N} p(y_i|x_i, \boldsymbol{w}, \sigma_\varepsilon^2)\right] p(\boldsymbol{w}|\sigma_w^2) p(\hat{y}|\hat{x}, \boldsymbol{w}, \sigma_\varepsilon^2) \tag{1.10}$$

Finalmente, se deduce la distribución predictiva para $\hat{y}$:

$$p(\hat{y}|\hat{x}, \boldsymbol{y}, \boldsymbol{x}, \sigma_w^2, \sigma_\varepsilon^2) \propto \int p(\hat{y}, \boldsymbol{y}, \boldsymbol{w}|\hat{x}, \boldsymbol{x}, \sigma_w^2, \sigma_\varepsilon^2)\ d\boldsymbol{w} \tag{1.11}$$

El modelo gráfico dirigido que encapsula estas últimas ecuaciones se aprecia en 2(c).

1.1.3. Modelos gráficos dirigidos gaussianos. Sea $\mathcal{M}$ un modelo grafico dirigido, en el cual todas las variables son reales y sus distribuciones de probabilidad condicional son lineal-gaussianas:

$$p(x_t|\boldsymbol{x}_{pa(t)}) = \mathcal{N}(x_t|\mu_t + \boldsymbol{w}_t^T \boldsymbol{x}_{pa(t)}, \sigma_t^2) \tag{1.12}$$

La estructura de $\mathcal{M}$ permite modelar la probabilidad conjunta de las variables del modelo en la forma:

$$p(\boldsymbol{x}|\mathcal{M}) = \prod_{t=1}^{V} p(x_t|\boldsymbol{x}_{pa(t)}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) \tag{1.13}$$

Lo cual se conoce como **red bayesiana gaussiana**. Para este tipo de modelos, es posible inferir $\boldsymbol{\mu}$ y $\boldsymbol{\Sigma}$. En efecto, según (1.13):

$$\log p(\boldsymbol{x}|\mathcal{M}) = -\sum_{t=1}^{V} \frac{1}{2\sigma_t^2} \left(x_t - \sum_{s \in pa(t)} w_{ts} x_s - \mu_t \right)^2 + K \tag{1.14}$$

Donde K representa una constante independiente de de $\boldsymbol{x}$. Al ser la log-probabilidad conjunta, cuadrática en las componentes de $\boldsymbol{x}$, se obtiene que efectivamente la probabilidad conjunta es normal multivariada para $\boldsymbol{x}$ en (1.13). Para estimar la media, se observa en primera instancia:

$$x_t = \sum_{s \in pa(t)} w_{ts} \mathbb{E}[x_s] + \mu_t + \sigma_t \varepsilon_t \tag{1.15}$$

Donde $\varepsilon_t \sim \mathcal{N}(0,1)$ y $\mathbb{E}[\varepsilon_t, \varepsilon_s] = 0$, para $s \neq t$. De esto se deduce:

$$\mathbb{E}[x_t] = \sum_{s \in pa(t)} w_{ts} \mathbb{E}[x_s] + \mu_t \tag{1.16}$$

Es posible entonces, encontrar las componentes de $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}] = (\mathbb{E}[x_1], \ldots, \mathbb{E}[x_V])^T$ utilizando la estructura gráfica dirigida de $\mathcal{M}$ (y por tanto su ordenamiento topológico). Para ello, se comienza calculando $\mathbb{E}[x_1]$ para luego continuar de manera recursiva según la numeración de los nodos.

Similarmente, es posible calcular el elemento $\boldsymbol{\Sigma}_{st}$ de la matriz de covarianza, observando:

$$\begin{aligned} \operatorname{cov}(x_s, x_t) &= \mathbb{E}[(x_s - \mathbb{E}[x_s])(x_t - \mathbb{E}[x_t])] \\ &= \left[(x_s - \mathbb{E}[x_s]) \left\{ \sum_{k \in pa(x_t)} w_{tk}(x_k - \mathbb{E}[x_k]) + \sigma_t \varepsilon_t \right\} \right] \\ &= \sum_{k \in pa(x_t)} w_{tk} \operatorname{cov}[x_s, x_t] + \sigma_t^2 \boldsymbol{I}_{st} \end{aligned} \tag{1.17}$$

De donde al igual que en (1.16), se calculan los elementos de $\boldsymbol{\Sigma}$ recursivamente.

Finalmente, se puede extender el modelo inducido por (1.12) a uno donde los nodos del modelo gráfico representen variables aleatorias gaussianas multivariantes. Para esto, se reescribe la distribución de probabilidad condicional para el nodo x_t en la forma:

$$p(\boldsymbol{x}_t \mid pa(\boldsymbol{x}_t)) = \mathcal{N}\left(\boldsymbol{x}_t \,\middle|\, \sum_{s \in pa(\boldsymbol{x}_t)} \boldsymbol{W}_{ts} \boldsymbol{x}_s + \boldsymbol{\mu}_t \,, \boldsymbol{\Sigma}_t \right) \tag{1.18}$$

Donde $\boldsymbol{W}_{ts}$ es una matriz de pesos entre los vectores de cada nodo.

1.2. Independencia condicional en modelos gráficos dirigidos

Como se mencionó anteriormente, los modelos gráficos encapsulan las relaciones de independencia condicional entre las variables aleatorias del fenómeno que se modela. En esta sección se estudian las propiedades de los modelos gráficos dirigidos en cuanto a estas relaciones.

En un grafo G, se escribe $x_i \perp_G x_j | x_k$ si el nodo x_i es independiente de x_j dado x_k. Se denota por $I(G)$ al conjunto de todas las relaciones de independencia condicional codificadas en el grafo G.

Definición 1.2 (Diagrama de independencia)**.** Sea $p(\cdot)$ una distribución de probabilidad. Se dice que un grafo G es un diagrama de independencia o *I-map* para p si y solo si $I(G) \subseteq I(p)$. Donde $I(p)$ es el conjunto de todas las relaciones de independencia condicional ciertas para las variables de p.

De la definición anterior, se deduce que un grafo G es un diagrama de independencia para la distribución de probabilidad p, si este no contiene más relaciones de independencia condicional que las permitidas por p. De esta forma, toda distribución de probabilidad $p(\boldsymbol{x})$, donde $\boldsymbol{x} = (x_1, ..., x_V)^T$, posee al menos un diagrama de independencia. En efecto, si se considera un grafo G con nodos $\mathcal{V} = \{x_1, ..., x_V\}$ completamente conectados, entonces G es un diagrama de independencia para p pues no presenta aristas faltantes y por tanto se condiciona en todas las variables.

De la discusión anterior, tiene sentido hablar de un *diagrama de independencia minimal* G para p, es decir, un grafo G, tal que si G' es otro diagrama de independencia para p que cumple $G' \subseteq G$, entonces $G' = G$.

Finalmente, tal representación, permite extraer de su estructura gráfica, relaciones no triviales de independencia condicional, entre las variables de importancia. En el caso de un modelo gráfico dirigido, la noción de *separación dirigida* o *d-separación* facilita dicha tarea.

1.2.1. d-separación. Se dice que un *camino no dirigido* P esta *separado de manera dirigida* o *d-separado* por un conjunto de nodos E, si y solo si, se cumple alguna de las siguiente condiciones:

(1) P contiene una cadena, $s \to e \to t$, donde $e \in E$.
(2) P contiene una estructura $s \leftarrow e \to t$, donde $e \in E$.
(3) P contiene una estructura $s \to e \leftarrow t$, donde $e \notin E$ o e **no** es descendiente de algún elemento de E.

Se dice que un conjunto de nodos A esta d-separado de un conjunto de nodos B, dado un conjunto de nodos E, si y solo si, todo camino no dirigido desde cada nodo de A a cada nodo de B esta d-separado por E.

En un grafo acíclico dirigido G, se aprecia la siguiente propiedad:

Para comprender las propiedades anteriores, se analizan los siguientes ejemplos:

- Sea $x \to y \to z$ una cadena, tal grafo codifica la siguiente probabilidad conjunta:

$$p(x, y, z) = p(x)p(y|x)p(z|y)$$

  Usando la propiedad (1), se puede deducir que $x \perp z|y$. Esto se comprueba pues:

$$p(x, z|y) = \frac{p(x, y, z)}{p(y)} = \frac{p(x)p(y|x)p(z|y)}{p(y)} = \frac{p(x, y)p(z|y)}{p(y)} = p(x|y)p(z|y)$$

- Sea la estructura $x \leftarrow y \to z$, según la propiedad (2), $x \perp z|y$. En efecto,

$$p(x, z|y) = \frac{p(x, y, z)}{p(y)} = \frac{p(y)p(x|y)p(z|y)}{p(y)} = p(x|y)p(z|y)$$

- Sea finalmente la estructura $x \to y \leftarrow z$, en este caso $x \not\perp z|y$:

$$p(x, z|y) = \frac{p(x, y, z)}{p(y)} = \frac{p(x)p(y|x, z)p(z)}{p(y)} \neq p(x|y)p(z|y)$$

  Es por tal motivo que en la propiedad (3), se requiere en este tipo de estructuras, que no existan nodos, ni descendientes de nodos de la familia condicionante E.

En el último caso, se puede comprobar que los nodos x e y son marginalmente independientes entre si, es decir, $p(x,z) = p(x)p(z)$. Sin embargo, al condicionar ambos nodos por y, se vuelven dependientes, este efecto se denomina **paradoja de Berkson**. Finalmente, un modelo gráfico probabilístico que verifica la equivalencia (**??**) se dice que cumple la **propiedad global de Markov**.

1.2.2. Markov blankets. De la d-separación es posible concluir:

$$x_t \perp \boldsymbol{x}_{nd(t)\setminus pa(t)} | \boldsymbol{x}_{pa(t)} \tag{1.19}$$

Donde $nd(t)$ son los **no-descendientes** del nodo x_t [3]. De esta forma, es posible concluir que en 1(a) $x_4 \perp x_5 | x_1, x_2, x_3$, pues en efecto, $nd(4) \setminus pa(4) = x_5$ y $pa(4) = x_1, x_2, x_3$. La ecuación (1.19) se conoce como **propiedad dirigida local de Markov**.

En especial, dado que $pred(t) \subseteq nd(t)$ se deriva la **propiedad ordenada de Markov**, ya presentada en (1.3). Sorprendentemente, estas tres propiedades son equivalentes.

Por otra parte, para cada nodo x_t es posible extraer el conjunto de nodos que lo separan del resto del grafo, es decir, se puede para cada nodo x_t, obtener el conjunto de todas las variables aleatorias que lo vuelven condicionalmente independiente a los demás nodos del modelo. El conjunto antes descrito se denomina **Markov blanket** y se denota por $mb(t)$ este conjunto de nodos corresponde a:

$$mb(t) := ch(t) \cup pa(t) \cup copa(t) \tag{1.20}$$

Donde $ch(t)$ son los nodos hijos de x_t, de manera análoga $pa(t)$ son nodos padres y $copa(t)$ sus copadres [4]. En la figura 1(a) se tiene por ejemplo $mb(5) = \{x_6, x_1, x_3, x_4\}$. Según la propiedad global de Markov, la presencia de los nodos copadres no parece ser necesaria en primera instancia (la dependencia condicional debería recaer unicamente en los nodos padres), sin embargo, al definir $\boldsymbol{x}_{-t}$ como el conjunto de nodos distintos a x_t, es posible observar que la probabilidad conjunta adquiere la forma $p(\boldsymbol{x}) = p(x_t, \boldsymbol{x}_{-t})$. De donde, al marginalizar sobre el nodo x_t, se obtiene que $p(\boldsymbol{x}_{-t})$ contiene sólo a aquellos nodos del modelo en los que la variable x_t no aparece como argumento, ni como condicionante (dada la factorización de la probabilidad conjunta codificada en el grafo). Lo anterior implica que en $p(x_t|\boldsymbol{x}_{-t}) = p(\boldsymbol{x})/p(\boldsymbol{x}_{-t})$ solo se podrán encontrar probabilidades condicionales donde x_t sea el argumento, lo que expresa con $p(x_t|\boldsymbol{x}_{pa(t)})$, o donde sea variable condicionante, es decir, sea padre o copadre de algún otro nodo. Se deduce:

$$p(x_t|\boldsymbol{x}_{-t}) \propto p(x_t|\boldsymbol{x}_{pa(t)}) \prod_{s \in ch(t)} p(x_s|\boldsymbol{x}_{pa(s)}) \tag{1.21}$$

La expresión (1.21) se conoce como **condicional completa** del nodo x_t.

---

[3] $nd(t) = \mathcal{V} \setminus \{t \cup desc(t)\}$, donde $desc(t)$ son los descendientes del nodo x_t, es decir, aquellos nodos que provienen de un camino dirigido con origen en x_t.

[4] nodos que comparten hijos con x_t

CHAPTER 2

Métodos de inferencia aproximada

Introducción

En este capitulo se estudian métodos de inferencia aproximada, en particular se abordan algoritmos basados en Markov Chain Monte Carlo e inferencia variacional. La idea detrás de la inferencia aproximada se basa en la obtención de muestras de una distribución de la forma $\boldsymbol{x}^s \sim p(\boldsymbol{x}|\mathcal{D})$ para calcular por ejemplo la distribución posterior predictiva de cierto modelo $p(y|\mathcal{D})$ o cierta marginal posterior $p(x_1|\mathcal{D})$. Cuando no se conocen las formas cerradas de estos objetos, es necesario por tanto aproximarlos de manera tal que se tengan garantías sobre las cantidades aproximadas que se obtienen.

En términos generales, se considera una densidad conjunta de variables latentes $\boldsymbol{z} = z_1, \dots z_m$ y observaciones $\boldsymbol{x} = x_1, \dots, x_n$ dada por,

$$p(\boldsymbol{z}, \boldsymbol{x}) = p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z}).$$

Desde la perspectiva bayesiana, las variables latentes provienen de una densidad prior $p(\boldsymbol{z})$. Esta se relaciona con las observaciones a través de $p(\boldsymbol{x}|\boldsymbol{z})$. En este contexto, el problema de inferencia equivale a condicionar los datos y calcular la posterior $p(\boldsymbol{z}|\boldsymbol{x})$. Los método explorados a continuación, abordan tal problema a través de la aproximación de la probabilidad posterior, ya sea de manera numérica o a través de distribuciones base.

2.1. Inferencia Markov Chain Monte Carlo

En esta sección se estudian métodos de Monte Carlo basados en cadenas de Markov. Este tipo de método permite obtener representaciones numéricas (muestras) de distribuciones incluso de alta dimensionalidad.

2.2. Cadenas de Markov

Una cadena de Markov es una sucesión de variables aleatorias $x_1, \dots, x_n$ que verifican $x_{t+1} \perp x_1, \dots, x_{t-1} \mid x_t$. Es decir, la distribución condicional de x_{t+1}, que puede ser interpretada como un estado futuro, depende únicamente del estado presente x_t. Si la probabilidad de transición es independiente de t, la cadena se denomina Markov homogénea. Tal tipo de cadenas se define especificando las probabilidades de transición de un estado a otro. Para el caso continuo, esto se traduce en:

$$K(x, y) = p\left(x_{t+1} = y | x_t = x\right) \tag{2.1}$$

En el caso discreto, se obtiene una matriz de transición $K_{x,y}$. Dado cierto espacio de estado, una cadena de Markov homogénea en el tiempo posee una distribución

estacionaria π si:

$$\pi(y) = \int \pi(x)K(x,y)dx \tag{2.2}$$

Una cadena de MArkov se dice irreductible, si puede transitar desde cualquier estado x de un espacio de estado discreto a cualquier otro estado y en un numero finito de pasos, es decir, si existe t tal que $K^n_{xy} > 0$. Si una cadena con distribución estacionaria es irreductible, la distribución estacionaria es unica y la cadena se dice recurrente positiva. Una cadena recurrente positiva, aperiodica con distribución estacionaria π, cumple para toda distribución inicial λ sobre sus estados:

$$\lim_{n\to\infty} \|\lambda K^n - \pi\| = 0 \tag{2.3}$$

donde K es el operado (o matriz) de transición. Para cadenas de Markov irreductibles con una única distribución estacionaria π, según la ley de los grandes números se debe cumplir que el valor esperado de una función $g(x)$ sobre π verifica:

$$\mathbb{E}_\pi[g(x)] = \int g(x)\pi(x)dx = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} g\left(x_i\right) \tag{2.4}$$

Esta propiedad permite estimar calcular aproximaciones de cantidades especificas de interés a partir de una cadena de Markov. Los algoritmos que utilizan tal resultado se conoce como métodos MCMC.

Una cadena con distribución estacionara π se dice reversible si el par (x_t, x_{t+1}) tiene la misma distribución conjunta que (x_{t+1}, x_t). En terminos del operador de transición K, esto se traduce como

$$\pi\left(x_t\right) K\left(x_t, x_{t+1}\right) = \pi\left(x_{t+1}\right) K\left(x_{t+1}, x_t\right) \tag{2.5}$$

Una cadena de Markov no necesariamente debe ser reversible para poseer una distribución estacionaria. Sin embargo, la reversibilidad de una cadena, garantiza la existencia de una distribución estacionaria. Esta es la razon por la cual se busca que en la mayoria de los algotimos MCMC se cumpla esta propiedad (*balance detallado*).

2.3. Algorimo Metropolis Hastings (MH)

El algoritmo canónico MCMC se conoce como Metropolis-Hastings. Su formulación pasa por obtener muestras de una distribución $p(x)$ en un espacio de estados E donde $x \in E$. Se construye un kernel de transición $k(x, y)$ para pasar del estado x al y mediante dos procesos: en primer lugar, se especifica una distribución $q(y|x)$ propuesta (proposal). Para luego en segunda instancia obtener muestras de $q(y|x)$ con un coeficiente de aceptación $\alpha(x, y) = \min\left[1, \frac{p(y)q(x|y)}{p(x)q(y|x)}\right]$. De esta forma, el kernel de transición queda definido como $K(x, y) = q(y|x)\alpha(x, y)$. El algoritmo se describe a continuación:

El kernel de transición correspondiente al algoritmo MH es reversible y por tanto cumple con la propiedad de balance detallado, $p(x)K(x, y) = p(y)K(y, x)$.

Existen múltiples formas de construir distribuciones proposal q, según la elección de esta, se construyen distintas versiones del algoritmo MH.

2.3.1. MH caminata aleatoria (RWMH). Seleccionando $q(y|x) = q(y-x)$, ocurre que la direccóon y distancia de un nuevo punto con respecto al punto actual, es independiente del punto actual. Las distribuciones que cumplen esta condición de manera inmediata son $\mathcal{N}(x, \sigma^2)$ y $U(x - \sigma, x + \sigma)$

Algorithm 1: Algoritmo Metropolis Hastings

Input: Punto inicial x_1, $p(x)$, kernel de transición $q(y|x)$
Output: Vector de N puntos $x_1, \ldots, x_N$
for *t = 1, . . . , N-1* **do**
  Obtener una muestra y de $q(y|x_t)$
  Obtener una muestra de una v.a uniforme U
  if $U < \frac{p(y)q(x_t|y)}{p(x_t)q(y|x_t)}$ **then** $x_{t+1} = y$
  else $x_{t+1} = x_t$
end

2.3.2. Muestreo de independencia. Seleccionando $q(y|x) = q(y)$, es decir, el nuevo estado es independiente del estado actual. el coeficiente de aceptación en este caso corresponde a $\min\left\{1, \frac{p(y)q(x)}{p(x)q(y)}\right\}$. En general se busca que $q(x)$ sea similar a $p(x)$ pero con colas más pesadas.

2.3.3. Metropolis simétrico. Seleccionando $q(y|x) = q(x|y)$, esta opción simplica la probabilidad de aceptación a $\min\{1, f(y)/f(x)\}$.

2.4. Muestreo de Gibbs

El muestreo de Gibbs, supone una distribución $p(x)$ de la cual se desea obtener muestras, donde $x \in \mathbb{R}^d$. En este método, el kernel de transición $K(x, y)$ es separado en múltiples pasos. En cada paso, se actualiza el valor de una coordenada basada en la densidad condicional con respecto a la demás coordenadas. Esto se observa en el siguiente algoritmo:

Algorithm 2: Algoritmo de Muestreo de Gibss

Input: Punto inicial x_1, y distribución $p(x)$
Output: Vector de N puntos $x_1, \ldots, x_N$
for *t = 1, . . . , N-1* **do**
  Obtener una muestra $x_1^{t+1} \sim p(x^{t+1}|\boldsymbol{x}_{-1}^t)$
  Obtener una muestra $x_2^{t+1} \sim p(x^{t+1}|\boldsymbol{x}_{-2}^t)$
  $\vdots$
  Obtener una muestra $x_d^{t+1} \sim p(x^{t+1}|\boldsymbol{x}_{-d}^t)$
end

El kernel de transformación se puede escribir como:

$$K_{1\to d}\left(x_{t+1}|x_t\right) = \prod_{i=1}^{d} p\left(x_{t+1}^i|x_{t+1}^1, \ldots x_{t+1}^{i-1}, x_t^{i+1}, \ldots, x_{t+1}^d\right) \tag{2.6}$$

De esta relación, se puede observar que

$$p\left(x_t\right) K_{1\to d}\left(x_{t+1}|x_t\right) = p\left(x_{t+1}\right) K_{d\to 1}\left(x_t|x_{t+1}\right) \tag{2.7}$$

De donde al integrar en ambos lados

$$\int p\left(x_t\right) K_{1\to d}\left(x_{t+1}|x_t\right) dx = p(y) \tag{2.8}$$

Luego, p es la distribución estacionaria de la cadena de Markov formada por el kernel de transición $K(xt+1|x_t)$. El método de muestreo de Gibbs con el kernel propuesto no es reversible. Otra caracterisitca de este método, es que puede verse como un caso especial del metodo MH con coeficiente de aceptacion $\min\left(1, \frac{p(y)q(x|y)}{p(x)q(y|x)}\right) = 1$, de esto, al observar $\boldsymbol{y}_{-i} = \boldsymbol{x}_{-i}$ se tiene:

$$\begin{aligned} p(\boldsymbol{y})q(\boldsymbol{x}|\boldsymbol{y}) &= p\left(y_i|\boldsymbol{y}_{-i}\right) p\left(\boldsymbol{y}_{-i}\right) p\left(x_i|\boldsymbol{y}_{-i}\right) = p\left(y_i|\boldsymbol{x}_{-i}\right) p\left(\boldsymbol{x}_{-i}\right) f\left(\boldsymbol{x}_i|\boldsymbol{x}_{-i}\right) \\ &= p\left(x_i|\boldsymbol{x}_{-i}\right) p\left(\boldsymbol{x}_{-i}\right) p\left(y_i|\boldsymbol{x}_{-i}\right) = p(\boldsymbol{x})q(\boldsymbol{y}|\boldsymbol{x}) \end{aligned} \tag{2.9}$$

Finalmente, cabe destacar que la falencia del Muestreo Gibbs recae en que en cada paso, se deben obtener muestras de la distribución conjunta de las demás variables, este paso puede ser costoso desde el punto de vista computacional. Una solución a este problema consiste en aplicar un muestreo de esta probabilidad condicional aplicando MH, esto da lugar al algoritmo MWG (Metropolis within Gibbs).

2.5. Convergencia en métodos MCMC

Si bien la formulación de los algoritmos MCMC estudiados hasta el momento garantizan convergencia a la distribución que se desea muestrear, no se especifica a priori el número de iteraciones necesarias para alcanzar esta convergencia. Para abordar este problema se estudian esquemas de diagnostico capaces de detectar fallas en la convergencia en este tipo de algoritmos.

2.5.1. Tamaño efectivo de muestra. Sea $p(\boldsymbol{x})$ la densidad de la variable aleatoria $\boldsymbol{x}$ con desviación estándar σ_x, si se obtienen n muestras independientes, el error estándar Monte Carlo se aproxima a $\sigma/\sqrt{n}$. De esta forma, para medir la media de una cantidad con un error cercano al 3%,, con respecto a la incertidumbre σ_x, es necesario obtener alrededor de $n = 1000$ muestras.

Debido a la correlación existente en los puntos de una cadena MCMC, el calculo anterior no se verifica (las muestras no son independientes). No obstante, es posible medir que tan correlacionados están estos punto entre si, por medio del calculo de la autocorrelación, esta se define por:

$$\rho_{xx}(t) = \frac{\mathbb{E}\left[\left(x_i - \overline{x}\right)\left(x_{i+t} - \overline{x}\right)\right]}{\mathbb{E}[\left(x_i - \overline{x}\right)^2]} \tag{2.10}$$

la cual se aplica sobre dos puntos separados por una distancia fija (o lag) t. Es de esperar, que la autocorrelación disminuya con respecto a la distancia t con que se comparan los puntos, de hecho el valor de la autocorrelación decae de manera proporcional a $\exp()-t/\tau x)$ donde tau_x se conoce como la *longitud de correlación.* En este contexto, se define la *correlación integrada* como $\tau_{int,x} = \sum_t \rho_{xx}(t)$. La variancia de la media $\overline{x}$ de $\boldsymbol{x}$, para una muestra de tamaño n se puede escribir en función de la correlación integrada obteniéndose:

$$\mathrm{Var}(\overline{x}) = \left(2\ \tau_{\mathrm{int,x}}\right) \frac{\mathbb{E}[\left(x_i - \overline{x}\right)^2]}{n} \tag{2.11}$$

De esta forma, para muestras correlacionadas, la varianza pasa a ser 2 $\tau_{int,x}$ veces mayor con respecto a la variancia sobre muestras independientes. Usando

$\tau_{int,x}$ es posible medir el *numero efectivo* de muestras independientes en una cadena correlacionada calculado $n/(2\,\tau_{int,x})$ para luego usar esta cantidad para decidir si la cantidad de muestras es suficiente, en el caso de la media se busca como heurística un numero efectivo de muestras mayor a 1000.

2.5.2. Varianza entre cadenas. Sean M cadenas consistentes de $2N$ iteraciones, de las cuales se utilizan las últimas N muestras. Dado un modelo probabilístico un parámetro de interés θ, sea $[\theta_{mt}\}_{t=1}^{N}$ la cadena m-ésima para $m = 1, \dots M$. En este contexto, sean $\hat{\theta}_m$ y $\hat{\sigma}_m^2$ la media muestral posterior y la variancia de la cadena m-ésima, sea además, la media posterior muestral total $\hat{\boldsymbol{\theta}} = (\mathbf{1}/M)\sum_{m=1}^{M}\hat{\theta}_m$. La variancia entre cadenas (B) e intra cadena (W) estan dadas por

$$B = \frac{N}{M-1}\sum_{m=1}^{M}\left(\hat{\theta}_m - \hat{\theta}\right)^2 \quad \text{y} \quad W = \frac{1}{M}\sum_{m=1}^{M}\hat{\sigma}_m^2 \tag{2.12}$$

El estimador insesgado $\hat{V}$ para la variancia posterior marginal de θ se obtiene bajo ciertas condiciones de estacionalidad [**16**] por medio de:

$$\widehat{V} = \frac{N-1}{N}W + \frac{M+1}{MN}B \tag{2.13}$$

El factor potencial de reducción de escala (PSRF) se define como el coeficiente entre $\widehat{V}$ y W. Si las M cadenas convergen a la distribución posterior, entonces el factor PSRF debe ser cercano a 1. Este factor estima la disminución potencial para la variabilidad entre cadenas B con respecto a la variabilidad intra cadenas W.

Utilizando una formulación similar [**17**], es posible obtener una corrección a este índice:

$$R_c = \sqrt{\frac{\hat{d}+3}{\hat{d}+1}\frac{\widehat{V}}{W}} \tag{2.14}$$

donde $\hat{d}$ representa los grados de libertad estimados de una distribución t de Student. Si R_c crece, a medida que las cadenas son iteradas una mayor cantidad de veces, se espera que tanto B disminuya o W crezca. Según se sugiere en [**16**] si $R_c < 1.2$ para todos los parámetros de interés, se puede considerar que se ha alcanzado convergencia.

2.6. Monte Carlo Hamiltoniano

Una ventaja de MCMC sobre otro métodos de Monte Carlo, es su rendimiento en muestreo de probabilidades en altas dimensiones. Sin embargo, en presencia de dimensionalidad extremadamente alta, los algoritmo tradicionales MCMC comienzan a presentar problemas. La formulación HMC (Monte Carlo Hamiltoniano) introduce una variable auxiliar de *momentum* u para para cada variable x. La distribución log posterior $\pi(x)$ se considera como el potencial de energia $U(x) = -\ln\pi(x)$ donde el momentum define la energía cinética $K(u)$. De esta forma se define el *Hamiltoniano* $H(x,u) = U(x) + K(u)$, donde $K(u) = u^2/2$. La distribución a explorar corresponde a

$$p(x,u) = \exp[-H(x,u)] = \exp\left[-\ln \pi(x) - \frac{1}{2}u^2\right] \tag{2.15}$$

Siguiendo las ecuaciones de la dinámica hamiltoniana [**18**], se encuentra un nuevo punto/posiciónx' que finalmente es aceptado o rechazado en base al algoritmo MH. El uso de la dinámica hamiltoniana permite explorar área en posiciones más lejanas a la posición actual.

Cabe destacar que si bien HMC aborda el problema de muestreo en altas dimensiones. para su implementación se requiere el gradiente de la densidad buscada, además de dos parámetros extra que deben ser obtenidos por el usuario.

2.7. Inferencia variacional

En esta sección, se estudia un método de aproximación de distribucioness conocido como inferencia variacional (**VI** por sus siglas en ingles). En general, la aproximación por inferencia variacional tiende a ser más rápida y fácil de escalar a datos de gran tamaño en comparación al muestreo por **MCMC**. Por tal motivo, se ha hecho presente en análisis de documentos a gran escala y problemas de neurociencia computacional.

A diferencia de los métodos ya estudiados, la idea principal detrás de la inferencia variacional consiste en usar optimización sobre una familia $\mathcal{Q}$ de densidades aproximadas sobre las variables latentes $\boldsymbol{z} = z_1, \dots z_m$. Basándose en esto, se busca al miembro (distribución) de esa familia que minimice cierta noción de distancia o disimilitud $\boldsymbol{D}(\cdot,\cdot)$ [1] en el espacio de distribuciones, con respecto a la densidad posterior ideal,

$$q^*(\boldsymbol{z}) = \arg\min_{q(\boldsymbol{z}) \in \mathcal{Q}} \boldsymbol{D}(q(\boldsymbol{z}), p(\boldsymbol{z}|\boldsymbol{x})). \tag{2.16}$$

Finalmente, se utiliza la distribución aproximada $q^*(\cdot)$ para aproximar la posterior buscada. De esta forma, el problema de inferencia pasa a ser un problema de optimización, donde la complejidad se ve regulada por la estructura $\mathcal{Q}$. Por tanto, se busca modelar $\mathcal{Q}$ aumentando su expresividad, de manera tal, que se garantice una buena aproximación de $p(\boldsymbol{z}|\boldsymbol{x})$, pero a la vez, se busca disminuir su complejidad en cuanto al gasto computacional.

2.8. Descripción del método

La idea clave tras el proceso de aproximación por inferencia variacional, pasa por resolver su problema de optimización subyacente. Desde este punto de vista, $\mathcal{Q}$ se puede escoger como una familia de funciones parametrizadas, por tanto, el problema de encontrar la función optima $q^*(\cdot)$ se transforma en encontrar los parámetros que la caracterizan. Para definir esta optimalidad, en la aproximación, se utiliza la divergencia de *Kullback Leibler* (KL), dada por:

$$\mathtt{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) = \mathbb{E}\left[\log q(\boldsymbol{z})\right] - \mathbb{E}\left[\log p(\boldsymbol{z}|\boldsymbol{x})\right], \tag{2.17}$$

La densidad variacional obtenida de esta forma, sirve para representar la densidad condicional exacta.

En el contexto inicial, si $\boldsymbol{x} = x_1, \dots, x_n$ es un conjunto de variables observadas y $\boldsymbol{z} = z_1, \dots, z_m$ variables latentes, con densidad conjunta $p(\boldsymbol{z}, \boldsymbol{x})$. El problema de inferencia, cosiste en calcular la densidad condicional de las variables latentes dadas las observaciones, $p(\boldsymbol{z}|\boldsymbol{x})$, esta se puede escribir como:

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{z}, \boldsymbol{x})}{p(\boldsymbol{x})}. \tag{2.18}$$

Donde el denominador contiene la densidad marginal de las observaciones, también llamada la *evidencia*. Al marginalizar sobre las variables latentes, se obtiene

[1] No se requiere estrictamente que $\boldsymbol{D}(\cdot,\cdot)$ sea una métrica.

$$p(\boldsymbol{x}) = \int p(\boldsymbol{z}, \boldsymbol{x}) d\boldsymbol{z}. \tag{2.19}$$

La dificultad de este proceso, recae en el calculo de esta integral, pues por lo general no se tiene su forma cerrada o se requiere de recursos computacionales prohibitivos para obtenerla. A modo de ejemplo se estudia el siguiente modelo:

2.8.1. GMM Bayesiano I. Se considera un **GMM** (modelo de mezcla de Gaussianas) de C componentes, univariado y de varianza unitaria. Cada una de sus componentes corresponden a distribuciones Gaussianas con medias $\boldsymbol{\mu} = \{\mu_1, \ldots, u_C\}$.

Los parámetros correspondientes a las medias se obtienen independientemente según su distribución prior $p(\mu_k)$, que se supondrá $\mathcal{N}(0, \sigma^2)$, donde la varianza prior σ^2 corresponde a un hiperparámetro.

Para generar una observación x_i del modelo, es necesario generar una asignación de cluster, esta se denota por c_i e indica de qué cluster latente x_i proviene. Posteriormente, el valor de c_i se obtiene de una distribución categórica sobre $\{1, \ldots, C\}$, siendo c_i un vector indicador de dimensión C. Finalmente, se asgina a x_i un valor obtenido de $\mathcal{N}(c_i^\intercal \boldsymbol{\mu}, 1)$. Esto se resume según el siguiente esquema:

$$\begin{aligned} \mu_k &\sim \mathcal{N}(0, \sigma^2), & k &= 1, \ldots, C, \\ c_i &\sim \mathrm{Cat}(1/k, \ldots, 1/k), & i &= 1, \ldots, n, \\ x_i | c_i, \boldsymbol{\mu} &\sim \mathcal{N}(c_i{}^\intercal \boldsymbol{\mu}, 1) & i &= 1, \ldots, n. \end{aligned} \tag{2.20}$$

Para una muestra de tamaño n, la densidad conjunta de variables latentes y observadas es

$$p(\boldsymbol{\mu}, \boldsymbol{c}, \boldsymbol{x}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \boldsymbol{\mu}). \tag{2.21}$$

En este modelo, las variables latentes corresponden a $\boldsymbol{z} = \{\boldsymbol{\mu}, \boldsymbol{c}\}$. De este modo, la evidencia se expresa por:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} \sum_{c_i} p(c_i) p(x_i | c_i, \boldsymbol{\mu}) d\boldsymbol{\mu}. \tag{2.22}$$

El integrando en la ecuación (2.22) no se puede reduir a un producto de integrales unidimensionales sobre los μ_k. Más aún, la complejidad del tiempo de evaluar numéricamente esta integra C-dimensional es $\mathcal{O}(k^n)$.

Si por otra parte, se distribuye el producto sobre la suma en (2.22), es posible escribir la evidencia como:

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{c}} p(\boldsymbol{c}) \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(x_i | c_i, \boldsymbol{\mu}) d\boldsymbol{\mu}. \tag{2.23}$$

Donde debido a la conjugación entre las distribuciones (prior gaussiana), cada integral individual pasa a ser computable. Sin embargo, se deben calcular k^n integrales

de este tipo: una para cada configuración de las asignaciones de cluster. El calculo de la evidencia permanece exponencial en C y por tanto intratable.

2.9. Cota inferior para la evidencia

Para abordar problemas como el anterior desde el punto de vista de inferencia variacional, se especifica una familia $\mathcal{Q}$ de densidades sobre las variables latentes. Cada $q(\boldsymbol{z}) \in \mathcal{Q}$ es una aproximación candidata densidad condicional exacta.

Para encontrar tal candidato, se minimiza la divergencia $\mathtt{KL}$ con respecto a la probabilidad condicional exacta, esto se traduce en resolver el siguiente problema:

$$q^*(\boldsymbol{z}) = \arg\min_{q(\boldsymbol{z}) \in \mathcal{Q}} \mathtt{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})). \tag{2.24}$$

Esto pues, se requiere calcular:

$$\mathtt{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) = \mathbb{E}\left[\log q(\boldsymbol{z})\right] - \mathbb{E}\left[\log p(\boldsymbol{z}|\boldsymbol{x})\right], \tag{2.25}$$

Que corresponde a

$$\mathtt{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) = \mathbb{E}\left[\log q(\boldsymbol{z})\right] - \mathbb{E}\left[\log p(\boldsymbol{z}, \boldsymbol{x})\right] + \log p(\boldsymbol{x}). \tag{2.26}$$

Es decir, se requiere calcular la evidencia $\log p(\boldsymbol{x})$, que como se comprobó anteriormente, es por lo general difícil de calcular. Debido a que no se puede calcular la divergencia $\mathtt{KL}$, se optimiza una función objetivo equivalente (hasta la suma una constante):

$$\mathtt{ELBO}(q) = \mathbb{E}\left[\log p(\boldsymbol{z}, \boldsymbol{x})\right] - \mathbb{E}\left[\log q(\boldsymbol{z})\right]. \tag{2.27}$$

Esta función se denomina cota inferior para la evidencia (**ELBO** por sus siglas en ingles). De su definición, se observa que $\mathtt{ELBO}(q) = \log p(\boldsymbol{x}) - \mathtt{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x}))$. Cabe destacar, que $p(\boldsymbol{x})$, es una constante con respecto a $q(\boldsymbol{z})$.

Según lo anterior, maximizar la cota $\mathtt{ELBO}$ es equivalente a minimizar la divergencia $\mathtt{KL}$. Para entender el problema de inferencia variacional, se estudian las propiedades de este funcional. Para ello, se reescribe de la siguiente forma:

$$\begin{aligned}\mathtt{ELBO}(q) &= \mathbb{E}\left[\log p(\boldsymbol{z})\right] + \mathbb{E}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] - \mathbb{E}\left[\log q(\boldsymbol{z})\right] \\ &= \mathbb{E}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] - \mathtt{KL}(q(\boldsymbol{z}||p(\boldsymbol{z}))).\end{aligned}$$

Es decir, la cota $\mathtt{ELBO}$ corresponde a la resta entre log verosimilitud y la divergencia $\mathtt{KL}$ entre la distribución prior $p(\boldsymbol{z})$ y $q(\boldsymbol{z})$.

El primer término de la expresión anterior la verosimilitud esperada por tanto, al maximizar la cota $\mathtt{ELBO}$ se están buscando densidades que distribuyan su masa en aquellos sectores donde las variables latentes explican mejor los datos observados. Por su parte, el segundo término es la divergencia negativa entre la densidad variacional y la prior, es decir, esta parte de la igualdad fomenta densidades cercanas a la prior. Se concluye que al maximizar la cota $\mathtt{ELBO}$ se busca ajuste a los datos, al mismo tiempo que se regulariza según la distribución prior.

Otra propiedad de la cota $\mathtt{ELBO}$ es que además acota inferiormente la log evidencia, es decir, $\log p(\boldsymbol{x}) \geq \mathtt{ELBO}(q)$ para todo $q(\boldsymbol{z})$. Para observar esto, se puede obtener de las ecuaciones (2.26) y (2.27) la siguiente expresión:

$$\log p(\boldsymbol{x}) = \mathtt{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) + \mathtt{ELBO}(q) \tag{2.28}$$

El resultado buscado se obtiene al observar que $\texttt{KL}(\cdot) \geq 0$.

2.10. Familia variacional mean-field

En cuanto a la familia $\mathcal{Q}$ sobre la cual se busca aproximar, es necesario que su complejidad de sus complejidad permita resolver el problema de optimización subyacente. En esta contexto, se estudia la familia variacional *mean-field*, la cual se define de manera tal que las variables latentes son mutuamente independientes, cada una gobernada por un parametro distinto en la densidad variacional. Por tanto, si $q \in \mathcal{Q}$ entonces $q(\cdot)$ se podrá expresar como:

$$q(\boldsymbol{z}) = \prod_{j=i}^{m} q_j(z_j). \tag{2.29}$$

De esta forma, latente z_j se rige por su propia componente variacional $q_j(\cdot)$.

En principio, no existe restricción para seleccionar el factor $q_j(\cdot)$, no obstante, su elección se relaciona directamente con el tipo de variable (continua/discreta) latente que se modela. Finalmente, los parámetros que caracterizan estos factores, definen al elemento de la familia $\mathcal{Q}$. Se continua con el ejemplo presentado anteriormente.

2.10.1. GMM Bayesiano II. Al modelo GMM bayesiano del ejemplo anterior, se agrega una familia variacional mean-field de la forma:

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

En este caso, el factor $q(\mu_k; m_k, s_k^2)$ se modela como una distribución gaussiana en la componente k-esima con media m_k y varianza es s_k^2. El factor $q(c_i; \varphi_i)$ corresponde a una distribución para la i-ésima asignación cluster y por tanto, sus probabilidades de asignación corresponden a un vector C-dimensional denotado por φ_i. De esta manera, se posee una familia paramétrica de distribuciones para los factores de $q(\cdot)$ y se puede abordar la maximización de la cota `ELBO` mediante la ecuación (2.21), usando la familia mean-field de la ecuación (2.30). El problema de optimización variacional correspondiente maximiza la cota `ELBO` con respecto a los parámetros variacionales, es decir, los parámetros gaussianos para cada componente y los parámetros categóricos para cada asignación de cluster.

2.11. Inferencia variacional mean-field de coordenadas ascendentes

El método descrito anteriormente, permite por medio de la cota `ELBO` y una familia mean-field adecuada, establecer el problema de inferencia variacional y obtener su formulación como un problema de optimización. En esta sección, se describe el algoritmo de **inferencia variacional mean-field de coordenadas ascendentes** (`CAVI` por sus siglas en ingles), el cual permite resolver tal problema de optimización.El algoritmo `CAVI` optimiza iterativamente cada factor de la densidad variacional mean-field, mientras mantiene los demás fijos.

2.11.1. Formulación. Para la j-ésima variable latente z_j, se utiliza su *condicional completa*, es decir, su densidad condicional dadas las demás variables latentes y las observaciones $p(z_j|\boldsymbol{z}_{-j}, \boldsymbol{x})$. Manteniendo fijos los factores variacionales $q_\ell(z_\ell), \ell \neq j$, se obtiene que el factor óptimo $q_j^*(z_j)$ verifica,

$$q_j^*(z_j) \propto \exp\left\{\mathbb{E}_{-j}\left[\log p(\boldsymbol{z}_j|z_{-j}, \boldsymbol{x})\right]\right\}. \tag{2.31}$$

El valor esperado en la ecuación (2.31) se calcula con respecto la densidad variacional de las variables $\boldsymbol{z}_{-j}$ para todo $j = 1, \ldots, m$, esta a su vez se expresa como $\prod_{\ell \neq j} q_\ell(z_\ell)$, de esta forma, se pueden incluir las variables $\boldsymbol{z}_{-j}$ al calculo de la probabilidad condicional y excluir del calculo del valor esperado $\mathbb{E}_j$. Por tal motivo, se verifica la siguiente relación de proporcionalidad (sobre la conjunta):

$$q_j^*(z_j) \propto \exp\left\{\mathbb{E}_{-j}\left[\log p(\boldsymbol{z}_j, z_{-j}, \boldsymbol{x})\right]\right\}. \tag{2.32}$$

Debido a que todas las variables latentes son consideradas independientes (hipotesis mean-field) los valores esperados del lado derecho no involucran al j-ésimo factor variacional. Por tanto la ecuación anterior, permite actualizar los factores (coordenadas) de manera iterativa. A continuación se presenta este algoritmo:

Algorithm 3: `CAVI`

Input: Modelo $p(\boldsymbol{x}, \boldsymbol{z})$ con datos $\boldsymbol{x}$
Output: Densidad variacional $\prod_{j=1}^m q_j(z_j)$

Inicializar: Factores variacionels $q_j(z_j)$
while `ELBO` *no converge* **do**
  for $j = 1, \ldots, m$ **do**
    Calcular $q_j^*(z_j) \propto \exp\left\{\mathbb{E}_{-j}\left[\log p(\boldsymbol{z}_j|z_{-j}, \boldsymbol{x})\right]\right\}$
  end
end
return $q(\boldsymbol{z})$

2.11.2. Actualización de coordenadas. .

El proposito de esta sección es obtener la actualización de coordenadas en la ecuación (2.32). Para ello se reescribe la cota `ELBO` en

$$\texttt{ELBO}(q_j) = \mathbb{E}_j\left[\mathbb{E}_{-j}\left[\log p(z_j, \boldsymbol{z}_{-j}, \boldsymbol{x})\right]\right] - \mathbb{E}_j\left[\log q_j(z_j)\right] + \tag{2.33}$$

De esta forma, el primer término de `ELBO` se reescribe como una esperanza iterada. El segundo término, por su parte, se descompone, usando la hipótesis de independencia mean-field y conservando solo el término que depende de $q_j(z_j)$.

Hasta una constante añadida, la función objetivo en la ecuación (2.33) es igual a la divergencia `KL` negativa entre $q_j(z_j)$ y $q_j^*(z_j)$ de la ecuación (2.32). Por lo tanto, se maximiza el `ELBO` con respecto a q_j cuando fijamos $q_j(z_j) = q_j^*(z_j)$.

2.11.3. GMM Bayesiano III. Se continua explorando el modelo GMM, para ello se consideran C componentes y n observaciones (datos) $x_1, \ldots, x_n$. Las variables latentes son C parámetros $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_C)^T$ y n asignaciones de clase $\boldsymbol{c} = c_1, \ldots, c_n$. Se supone nuevamente una varianza unitaria y una distribución prior uniforme sobre los componentes de la mezcla.

La densidad conjunta de las variables latentes y observadas se encuentra en la ecuación (2.21). La familia variacional está en la ecuación (2.30). En este caso hay

dos tipos de parámetros variacionales: los parámetros categóricos φ_i para aproximar la asignación de clúster posterior del i-ésimo punto de datos y los parámetros gaussianos m_k y s_k^2 para aproximar la posterior de la k-ésima componente de mezcla.

Usando la familia mean-field en la definición de la cota `ELBO`, se obtiene una función de los parámetros variacionales $\boldsymbol{m}$, $\boldsymbol{s}^2$ y $\boldsymbol{\varphi}$:

$$\begin{aligned}\texttt{ELBO}(\boldsymbol{m}, \boldsymbol{s}^2, \boldsymbol{\varphi}) = &\sum_{k=1}^{C} \mathbb{E}\left[\log p(\mu_k); m_k, s_k^2\right] \\ &+ \sum_{k=1}^{C} (\mathbb{E}\left[\log p(c_i); \varphi_i\right] + \mathbb{E}\left[\log p(x_i|c_i, \boldsymbol{\mu}); \varphi_i, \boldsymbol{m}, \boldsymbol{s}^2\right]) \\ &- \sum_{k=1}^{C} \mathbb{E}\left[\log q(c_i; \varphi_i)\right] - \sum_{k=1}^{C} \mathbb{E}\left[\log q(\mu_k; m_k, s_k^2\right].\end{aligned} \tag{2.34}$$

En cada término, se hace explícita la dependencia de los parámetros variacionales. Más aún, cada esperanza puede ser calculada en forma cerrada.

Debido a que el algoritmo `CAVI` actualiza cada parámetro variacional a su vez, es posible obtener una regla de actualización para cada factor de forma cerrada.

2.11.4. Densidad variacional para las asignaciones de cluster. Se derivamos la actualización variacional para la asignación de cluster c_i. Para ello se usa la ecuación (2.32):

$$q^*(c_i; \varphi_i) \propto \exp\left\{\log p(c_i) + \mathbb{E}[\log p(x_i|c_i, \boldsymbol{\mu}); \boldsymbol{m}, \boldsymbol{s}^2]\right\}. \tag{2.35}$$

Los términos en el exponente son los componentes de la densidad conjunta que dependen de c_i, la esperanza en el segundo término se calcula sobre $\boldsymbol{\mu}$. El primer término de la ecuación (2.35) corresponde a la log prior de c_i, que al considerarla uniforme pasa a ser $\log p(c_i) = -\log k$.

El segundo término es la log esperanza de la c_i-ésima densidad gaussiana. Al ser c_i es un vector indicador, se obtiene:

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

Usando este resultado, y eliminando los términos constantes con respecto a c_i, se calcula la probabilidad de la log esperanza,

$$\begin{aligned}\mathbb{E}\left[\log p(x_i|c_i, \boldsymbol{\mu})\right] &= \sum_k c_{ik} \mathbb{E}\left[\log p(x_i|\mu_k); m_k, s_k^2\right] \\ &= \sum_k c_{ik} \mathbb{E}\left[-(x_i - \mu_k)^2/2; m_k, s_k^2\right] + \text{const.} \\ &= \sum_k c_{ik} \left(\mathbb{E}\left[\mu_k; m_k, s_k^2\right] x_i - \mathbb{E}\left[\mu_k^2; m_k, s_k^2\right]/2\right) + \text{const}\end{aligned} \tag{2.36}$$

Este cálculo requiere de $\mathbb{E}[\mu_k]$ y $\mathbb{E}[\mu_k^2]$ para cada componente, los cuales son calculables a través de la distribución variacional gaussiana para la k-ésima componente.

Por lo tanto, la actualización variacional para la i-ésima asignación de cluster, depende de únicamente de los parámetros variacionales y verifica:

$$\varphi_{ik} \propto \exp\left\{\mathbb{E}\left[\mu_k; m_k, s_k^2\right] x_i - \mathbb{E}\left[\mu_k^2; m_k, s_k^2\right]/2\right\} \tag{2.37}$$

2.11.5. Densidad variacional para las medias. Se continua con el cálculo de la densidad variacional $q(\mu_k; m_k, s_k^2)$ para la k-ésima componente. Nuevamente, se utiliza la ecuación (2.32) y se escribe la densidad conjunta hasta una constante de normalización:

$$q(\mu_k) \propto \exp\left\{\log p(\mu_k) + \sum_{i=1}^{n} \mathbb{E}\left[\log p(x_i|c_i, \boldsymbol{\mu}); \varphi_i, \boldsymbol{m}_{-k}, \boldsymbol{s}_{-k}^2\right]\right\}. \tag{2.38}$$

Posteriormente, se calcula la log coordenada-optimal $q(\mu_k)$. En este caso, dado nuevamente que c_i es un vector indicador, se aprecia que $\varphi_{ik} = \mathbb{E}[c_{ik}; \varphi_i]$, luego

$$\begin{aligned}
\log q(\mu_k) &= \log p(\mu_k) + \sum_i \mathbb{E}\left[\log p(x_i|c_i, \boldsymbol{\mu}); \varphi_i, \boldsymbol{m}_{-k}, \boldsymbol{s}_{-k}^2\right] + \text{const.} \\
&= \log p(\mu_k) + \sum_i \mathbb{E}\left[c_{ik} \log p(x_i|\mu_k); \varphi_i\right] + \text{const.} \\
&= -\mu_k^2/2\sigma^2 + \sum_i \mathbb{E}\left[c_{ik}; \varphi_i\right] \log p(x_i|\mu_k) + \text{const.} \\
&= -\mu_k^2/2\sigma^2 + \sum_i \varphi_{ik}\left(-(x_i - \mu_k)^2/2\right) + \text{const.} \\
&= -\mu_k^2/2\sigma^2 + \sum_i \varphi_{ik} x_i \mu_k - \varphi_{ik}\mu_k^2/2 + \text{const.} \\
&= \left(\sum_i \varphi_{ik} x_i\right) \mu_k - \left(1/2\sigma^2 + \sum_i \varphi_{ik}/2\right) \mu_k^2 + \text{const.}
\end{aligned} \tag{2.39}$$

El cálculo anterior muestra que la densidad variacional optima por coordenada de μ_k corresponde a una familia exponencial con estadísticos suficientes $\mu_k, \mu 2_k$ y parámetros $\left\{\sum_{i=1}^n \varphi_{ik} x_i - 1/2\sigma^2 - \sum_{i=1}^n \varphi_{ik}/2\right\}$, es decir, corresponde a una gaussiana. En términos de la media y la varianza variacionales, las reglas de actualización para $q_(\mu k)$ son

$$m_k = \frac{\sum_i \varphi_{ik} x_i}{1/\sigma^2 + \sum_i \varphi_{ik}}, \quad s_k^2 = \frac{1}{1/\sigma^2 + \sum_i \varphi_{ik}} \tag{2.40}$$

Estas reglas se relacionan fuertemente con la densisdad condicional completa de la para la k-ésima componente del modelo. La condicional completa es una gaussiana posterior dados los datos asignados a la k-ésima componente. La actualización variacional corresponde a una condicional completa ponderado, donde cada dato se pondera por su probabilidad variacional de ser asignado a la componente k.

2.11.6. `CAVI` para GMM. El algoritmo 4 muestra el algoritmo `CAVI` aplicado al GMM estudiado. En este, se combinan las actualizaciones variacionales en la ecuación (2.35) y la ecuación (2.40). El algoritmo requiere calcular el `ELBO` de la ecuación (2.34). Usando el `ELBO` para rastrear el progreso del algoritmo y evaluar cuándo ha convergido. Una vez se tiene una densidad variacional ajustada, esta se puede usar como reemplazo para la posterior. Por ejemplo, es posible obtener una

descomposición posterior de los datos. Asignando puntos a su cluster más probable $\hat{c}_i = \arg\max_k \varphi_{ik}$ y se estiman las medias de cluster con sus medias variacionales m_k.

Algorithm 4: `CAVI` para GMM

Input: Datos $x_1, \ldots, x_n$, numero de componentes C, varianza prior para las medias de las componentes σ^2

Output: Densidades variacionales $q(\mu_k; m_k, s_k^2)$ y $q(c_i, \varphi_i)$

Inicializar: $\boldsymbol{m} = m_1, \ldots m_k$, $\boldsymbol{s}^2 = s_1 \ldots s_k$, $\boldsymbol{\varphi} = \varphi_1, \ldots, \varphi_n$

while `ELBO` *no converge* **do**

  for $i = 1, \ldots, n$ **do**

    Calcular $\varphi_{ij} \propto \exp\left\{\mathbb{E}\left[\mu_k; m_k, s_k^2\right] x_i - \mathbb{E}\left[\mu_k^2; m_k, s_k^2\right]/2\right\}$

  end

  for $k = 1, \ldots, C$ **do**

    $m_k \leftarrow \frac{\sum_i \varphi_{ik} x_i}{1/\sigma^2 + \sum_i \varphi_{ik}}$

    $s_k^2 \leftarrow \frac{1}{1/\sigma^2 + \sum_i \varphi_{ik}}$

  end

  Calcular `ELBO`$(\boldsymbol{m}, \boldsymbol{s}^2, \boldsymbol{\varphi})$

end

return $q(\boldsymbol{m}, \boldsymbol{s}^2, \boldsymbol{\varphi})$

También es posible usar la densidad variable ajustada para aproximar la densidad predictiva en nuevos datos:

$$p(x_{\text{nex}}|x_{1:n}) \approx \frac{1}{K}\sum_{k=1}^{C} p(x_{\text{new}}|m_k), \tag{2.41}$$

donde $p(x_{\text{new}}|m_k)$ es gaussiana con media m_k y varianza unitaria.

2.12. Inferencia variacional con familias exponenciales

En el modelo GMM expuesto anteriormente, la distribución condicional completa pertenece a la clase de distribuciones conocida como *familia exponencial*, este tipo de distribuciones (normal, gamma, chi-cuadrado, beta, ...) han sido ampliamente estudiadas y poseen propiedades interesantes en cuanto a conjugación prior-posterior, entre otras. Los modelos cuyas distribuciones condicionales completas pertenecen a esta familia de distribuciones presentan grandes ventajas en cuando a al aproximación por medio de inferncia variacional. En esta sección se estudian tales propiedades.

2.12.1. Condicionales completas en familias exponenciales. Se considera el modelo $p(\boldsymbol{z}, \boldsymbol{x})$ y se agrega la suposición de que cada condicional completa pertence a la familia de distribuciones exponencial, es decir:

$$p(z_j|\boldsymbol{Z}_{-j}, \boldsymbol{x}) = h(z_j)\exp\left\{\eta_j(\boldsymbol{z}_j, \boldsymbol{x})^\intercal z_j - a(\eta_j(\boldsymbol{z}_{-j}, \boldsymbol{x}))\right\}, \tag{2.42}$$

Donde z_j es su propio estadístico suficiente, $h(\cdot)$ es una medida base, y $a(\cdot)$ es el log normalizador. Debido a que esta es una densidad condicional, el parámetro $\eta_j(\boldsymbol{z}_{-j}, \boldsymbol{x})$ es una función del conjunto condicionante.

En este contexto, la inferencia variacional por medio de dsitribuciones mean-field pasa por ajustar $q(\boldsymbol{z}) = \prod_j q_j(z_j)$. Al suponer pertenencia a la familia exponencial, se simplifica la actualización de coordenadas de la ecuación (2.31), en efecto:

$$\begin{aligned} q(z_j) &\propto \exp\left\{\mathbb{E}\left[\log p(z_j|\boldsymbol{z}_{-j}, \boldsymbol{x})\right]\right\} \\ &= \exp\left\{\log h(z_j) + \mathbb{E}\left[\eta_j(\boldsymbol{z}_{-j}, \boldsymbol{x})\right]^\intercal z_j - \mathbb{E}\left[a(\eta_j(\boldsymbol{z}_{-j}, \boldsymbol{x}))\right]\right\} \\ &\propto h(z_j)\exp\left\{\mathbb{E}\left[\eta_j(\boldsymbol{z}_{-j}, \boldsymbol{x})\right]^\intercal z_j\right\} \end{aligned} \tag{2.43}$$

En esta actualización se observa la forma paramétrica de los factores variacionales óptimos. Más aún, cada uno está en la misma familia exponencial que su correspondiente condicional completo. Su parámetro tiene la misma dimensión y tiene la misma medida base $h(\cdot)$ y log normalizador $a(\cdot)$.

Habiendo establecido sus formas paramétricas, se denota por ν_j el parámetro variacional para el factor variacional j-ésimo. Cuando se actualiza cada factor, se establece su parámetro igual al parámetro esperado de la condicional completa:

$$\nu_j = \mathbb{E}\left[\eta_j(\boldsymbol{z}_{-j}, \boldsymbol{x})\right]. \tag{2.44}$$

Esta transformación permite simplificar los calculos necesarios para implementar `CAVI`.

2.12.2. Conjugación condicional y modelos bayesianos. Un caso especial de modelos de familias exponenciales son los *modelos condicionalmente conjugados* con variables locales y globales. Este tipo de modelos son comunes en aprendizaje automático, donde las variables globales corresponden a "parámetros" y las variables locales a variables latentes.

En tal contexto, sea β un vector de *variables latentes globales* y $\boldsymbol{z}$ un vector de *variables locales latentes*. La probabilidad conjunta en este caso es:

$$p(\beta, \boldsymbol{z}, \boldsymbol{x}) = p(\beta)\prod_{i=1}^{n} p(z_i, x_i|\beta). \tag{2.45}$$

En la perspectiva del modelo GMM ya explorado, las variables globales corresponden a las componentes de mezcla y la i-ésima variable local es la asignación de cluster para el dato x_i.

En el modelo de la ecuación (2.45) se supondrá, que los términos que la componen hacen que cada condicional completa pertenesca a la familia exponencial. Para ello, se supondrá que la densidad conjunta de cada par (x_i, z_i), condicional a β, tiene una forma de familia exponencial,

$$p(z_i, x_i|\beta) = h(z_i, x_i)\exp\left\{\beta^\intercal t(z_i, x_i) - a(\beta)\right\}, \tag{2.46}$$

donde $t(\cdot, \cdot)$ es el estadístico suficiente. Posteriormente, se escoge una prior sobre las variables globales de manera que sea su conjugada,

$$p(\beta) = h(\beta)\exp\left\{\alpha^\intercal\left[\beta, -a(\beta)\right] - a(\alpha)\right\}. \tag{2.47}$$

Esta prior tiene el parámetro $\alpha = [\alpha_1, \alpha_2]^\intercal$, un vector de columna, y estadísticos suficientes que concatenan la variable global y su log normalizador en la densidad de las variables locales.

Con la prior conjugada, la condicional completa de las variables globales está en la misma familia. Su parámetro natural es

$$\hat{\alpha} = \left[\alpha_1 + \sum_{i=1}^{n} t(z_i, x_i), \alpha_2 + n\right]^\intercal . \tag{2.48}$$

En cuanto al condicional completo de la variable local z_i, se observa que dado β y x_i, la variable local z_i es condicionalmente independiente de las otras variables locales z_{-i} y otros datos x_{-i}. Esto se deduce de la forma de la densidad conjunta en la ecuación 2.45). Así

$$p(z_i|x_i, \beta, \boldsymbol{z}_{-i}, \boldsymbol{x}_{-i}) = p(z_i|x_i, \beta). \tag{2.49}$$

Si se supone además que esta densidad está en una familia exponencial,

$$p(z_i|x_i, \beta) = h(z_i)\exp\left\{\eta(\beta, x_i)^\intercal z_i - a(\eta(\beta, x_i))\right\}. \tag{2.50}$$

2.13. Inferencia variacional en modelos condicionalmente conjugados

A continuación, se describe `CAVI` para esta clase general de modelos. Para ello, sea $q(\beta|\lambda)$ la aproximación posterior variacional en β, se denomina además por λ al parámetro variacional global. El cual posee la misma densidad de familia exponencial que la prior. De forma similar, sea $q(z_i|\varphi_i)$ la distribución variacional posterior en cada variable local, en este caso, cada z_i rije por un parámetro local variacional φ_i. Este posee la misma densidad de familia exponencial que el condicional local completo. El algotitmo `CAVI` itera entre la actualización de cada parámetro variacional local y la actualización del parámetro variacional global. La actualización variacional local es

$$\varphi_i = \mathbb{E}_\lambda\left[\eta(\beta, x_i)\right]. \tag{2.51}$$

Esta es una aplicación de la Ecuación (2.44), donde se toma la esperanza del parámetro natural de la condicional completa en la ecuación (2.49). Para la actualización variacional global se aplica la misma técnica:

$$\lambda = [\alpha_1 + \sum_{i=1}^{n} \mathbb{E}_{\varphi_i}[t(z_i, x_i)], \alpha_2 + n]^\intercal. \tag{2.52}$$

Aquí se calcula la esperanza del parámetro natural en la Ecuación (2.48). El algoritmo `CAVI` optimiza la cota `ELBO` al iterar entre las actualizaciones locales de cada parámetro local y las actualizaciones globales de los parámetros globales. Para evaluar la convergencia, se puede calcular la cota `ELBO` en cada iteración hasta una constante que no depende de los parámetros variacionales,

$$\texttt{ELBO} = \left(\alpha_1 + \sum_{i=1}^{n} \mathbb{E}_{\varphi_i}[t(z_i, x_i)]\right)^\intercal \mathbb{E}_\lambda[\beta] - (\alpha_2 + n)\mathbb{E}_\lambda[a(\beta)] - \mathbb{E}[\log q(\beta, \boldsymbol{z})]. \tag{2.53}$$

Esta corresponde a la cota `ELBO` en la ecuación (2.27) aplicada en (2.45) con la correspondiente densidad variacional mean-field. El último término es

$$\mathbb{E}[\log q(\beta, \boldsymbol{z})] = \lambda^\intercal \mathbb{E}_\lambda[t(\beta)] - a(\lambda) + \sum_{i=1}^{n} \varphi_i^\intercal \mathbb{E}_{\varphi_i}[z_i] a(\varphi_i). \tag{2.54}$$

`CAVI` para GMM (Algoritmo 4) corresponde a una instancia de este método.

2.14. Inferencia variacional estocástica

En relación al escalamiento a conjuntos de datos masivos, la mayoría de los algoritmos de inferencia posterior, incluido `CAVI` sufren de problemas de escalamiento. La razón de esto, es que la estructura de ascenso de coordenadas del algoritmo requiere iterar a través de todo el conjunto de datos en cada iteración. Como alternativa a esto, surge la optimizción basada en gradiente, esta perspectiva es la clave para ampliar la inferencia variacional s su variante estocástica (`SVI`). Esta se centra en la optimización de los parámetros variacionales globales λ en modelos condicionalmente conjugados.

El esquema de calculo se basa en mantener una estimación de los parámetros variables globales, para repetidamente sub-samplear un dato del conjunto completo y luego, usar los parámetros globales actuales para calcular los parámetros locales óptimos para el dato muestreado, finalmente, se ajustan los parámetros globales actuales de una manera apropiada.

2.14.1. Gradiente natural de `ELBO`. En la optimización basada en gradiente, el termino *gradiente natural* toma en cuenta la estructura geométrica de los parámetros de probabilidad. Específicamente, los gradientes naturales deforman el espacio de parámetros de una manera sensible, de modo que mover la misma distancia en diferentes direcciones equivale a un cambio igual en la divergencia `KL` simetrizada. El gradiente Euclidiano habitual no posee esta propiedad.

En familias exponenciales, se encuentra el gradiente natural con respecto al parámetro al premultiplicar el gradiente usual por la covarianza inversa del estadístico suficiente, $a''(\lambda)^{-1}$. Esta es la métrica Riemanniana inversa y la matriz de información Fisher inversa.

Los modelos conjugados condicionalmente poseen gradientes naturales simples para la cota `ELBO`. Para gradientes con respecto al parámetro global λ se conoce la formula del gradiente euclidiano de `ELBO`:

$$\nabla_\lambda \texttt{ELBO} = a''(\lambda)(\mathbb{E}_\varphi[\hat{\alpha}] - \lambda), \tag{2.55}$$

donde $\mathbb{E}_\varphi[\hat{\alpha}]$ está en la ecuación (2.52). Premultiplicar por la información de Fisher inversa da el gradiente natural $g(\lambda)$,

$$g(\lambda) = \mathbb{E}_\varphi[\hat{\alpha}] - \lambda. \tag{2.56}$$

La cual es la diferencia entre las actualizaciones de coordenadas $E_\varphi[\hat{\alpha}]$ y los parámetros variacionales λ en los que se evalua el gradiente.

Es posible usar este gradiente natural en un algoritmo de optimización basad en gradiente. Donde en cada iteración, se actualizan los parámetros globales,

$$\lambda_t = \lambda_{t-1} + \epsilon_t g(\lambda_{t-1}), \tag{2.57}$$

donde ϵ_t corresponde a un tamaño de paso. Al sustituir la ecuación (2.56) en el segundo término se obtiene una estructura especial,

$$\lambda_t = (1 - \epsilon_t)\lambda_{t-1} + \epsilon_t \mathbb{E}_\varphi[\hat{\alpha}]. \tag{2.58}$$

Esto no requiere de cálculos adicionales a los de las actualizaciones de ascenso de coordenadas. En cada iteración, primero se calcula la actualización de coordenadas. Luego se ajusta la estimación actual para que sea una combinación ponderada de la actualización y el parámetro variacional actual.

2.14.2. Optimización estocástica de `ELBO`. Aunque es fácil de calcular, usar el gradiente natural tiene el mismo costo que la actualización de coordenadas en la ecuación (2.52), pues requiere sumar en todo el conjunto de datos y calcular los parámetros variables locales óptimos para cada uno de ellos.

La inferencia variacional estocástica resuelve este problema utilizando el gradiente natural en un algoritmo de optimización estocástico. Los algoritmos de optimización estocástica siguen gradientes ruidosos pero de bajo costo computacional para alcanzar el óptimo de una función objetivo.

El objetivo es construir un gradiente natural, ruidoso calculado a bajo costo computacional. Para ello, se expande el gradiente natural en la ecuación (2.56) usando (2.48):

$$g(\lambda) = \alpha + \left[\sum_{i=1}^{n} \mathbb{E}_{\varphi_i^*}[t(z_i, x_i)], n\right]^\intercal - \lambda, \tag{2.59}$$

Se construye un gradiente natural ruidoso al muestrear un índice de los datos y luego reescalar el segundo término,

$$t \sim \text{Unif}(1, \ldots, n) \tag{2.60}$$

$$\hat{g}(\lambda) = \alpha + n[\mathbb{E}_{\varphi_t^*}[t(z_t, x_t)], 1]^\intercal - \lambda. \tag{2.61}$$

El gradiente natural ruidoso $\hat{g}(\lambda)$ es insesgado: $E_t[\hat{g}(\lambda)] = g(\lambda)$. Y es computacionalmente eficiente, solo involucra un único dato muestreado y un conjunto de parámetros locales optimizados.

Finalmente, se establece una sucesión de tamaño de paso. La cual debe cumplir las condiciones:

$$\sum_t \epsilon_t = \infty \quad ; \quad \sum_t \epsilon_t^2 < \infty \tag{2.62}$$

Muchas sucesiones cumplen estas condiciones, por ejemplo $\epsilon_t = t^{-\kappa}$ para $\kappa \in (0.5, 1]$. `SVI` no requiere una nueva derivación más allá de lo que se necesita para `CAVI`. Cualquier implementación de `CAVI` puede escalarse inmediatamente a un algoritmo estocástico.

CHAPTER 3

Aprendizaje con Kernels

3.1. Introducción

Un kernel es una función simétrica y definida positiva $k(\cdot,\cdot)$ que puede ser entendida como una medida de similitud entre los argumentos que opera. En el siguiente capítulo, se definen estos objetos matemáticos de manera formal, se investigan sus características y se derivan algunos métodos del aprendizaje de máquinas que toman ventaja sus propiedades.

3.2. Terminología y propiedades

El término **kernel** proviene del estudio de operadores integrales en el campo del análisis funcional. En tal contexto se les identifica como aquellas funciones k que determinan un operador T_k a través de:

$$(T_k f)(x) = \int_{\mathcal{X}} k(x,x')f(x')dx \tag{3.1}$$

En concordancia con la perspectiva que se desea abarcar, se denotará como kernel a toda función $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ en el espacio de características [1]. Dentro de tal clase de funciones, son de importancia a aquellas capaces de "generalizar" el concepto de *porducto interno*, el **Teorema de Mercer** permite identificar tal subconjunto.

Teorema 3.1 (Teorema de Mercer)**.** Sea $(\mathcal{X},\mu)$ un espacio de medida finita y $k \in L_\infty(\mathcal{X}^2)$ una función real y simétrica, tal que el operador integral:

$$\begin{gathered} T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X}) \\ (T_k f)(x) := \int_{\mathcal{X}} k(x,x')f(x')d\mu(x') \end{gathered} \tag{3.2}$$

es semidefinido positivo, es decir, para toda $f \in L_2(\mathcal{X})$ se cumple:

$$\int_{\mathcal{X}^2} k(x,x')f(x)f(x')d\mu(x)d\mu(x') \geq 0 \tag{3.3}$$

Si $\psi_j \in L_2(\mathcal{X})$ son las funciones propias ortonormales de T_f asociadas a los valores propios $\lambda_j > 0$, ordenados de manera decreciente. Entonces,

(1) $(\lambda_j)_j \in l_1$

[1] El condominio del kernel k no tiene por que restringirse a los reales, para ciertas aplicaciones puede ser conviene utilizar los complejos. En general las propiedades de interés se preservan en ambos cuerpos, por lo que por simplicidad se considera solo el caso real en esta monografía.

(2) $k(x,x') = \sum_{j=1}^{N_{\mathcal{H}}} \lambda_j \psi_j(x)\psi_j(x')$ se cumple para casi todos los elementos $x, x' \in \mathcal{X}$. Además $N_{\mathcal{H}} \in \mathbb{N}$ o $N_{\mathcal{H}} = \infty$, en este último caso, la serie respectiva converge absoluta y uniformemente para casi todos los elementos $x, x' \in \mathcal{X}$.

La segunda implicación del teorema anterior, permite construir una aplicación $\Phi : \mathcal{X} \to l_2^{N_{\mathcal{H}}}$ tal que $x \mapsto (\sqrt{\lambda_j}\psi_j(x))_{j=1,\ldots,N_{\mathcal{H}}}$, es decir, a cada elemento de $\mathcal{X}$ se le asocia una transformación por medio de las funciones propias asociadas a $k(\cdot,\cdot)$ al espacio $l_2^{N_{\mathcal{H}}}$. Este último espacio esta provisto de producto interno, por lo que es posible expresar $k(x,x') = \langle \Phi(x), \Phi(x') \rangle_{l_2^{N_{\mathcal{H}}}}$ para casi todo $x \in \mathcal{X}$. Esto permite interpretar a Φ como una aplicación a un espacio de *características*, más aún, en este espacio $k(\cdot,\cdot)$ actúa como un producto interno, esto se concreta en el siguiente teorema.

Teorema 3.2 (Aplicación kernel de Mercer)**.** Si k es un kernel que cumple las condiciones del teorema (3.1), se puede construir una aplicación Φ a un espacio donde k se comporta como un producto interno:

$$k(x,x') = \langle \Phi(x), \Phi(x') \rangle$$

para casi todo $x, x' \in \mathcal{X}$. Más aún, para todo $\varepsilon > 0$, existe una aplicación Φ_n a un espacio $n-$dimensional con producto interno tal que

$$|k(x,x') - \langle \Phi_n(x), \Phi_n(x') \rangle| < \varepsilon$$

para casi todo $x, x' \in \mathcal{X}$.

La convergencia uniforme en el teorema anterior implica que para cualquier precisión $\varepsilon > 0$, debe existir un $n \in \mathbb{N}$ tal que k puede ser aproximado como un producto interno en $\mathbb{R}^n$. Lo anterior se observa al notar que para casi todo $x, x' \in \mathcal{X}$, se tiene $|k(x,x') - \langle \Phi^n(x), \Phi^n(x') \rangle| < \varepsilon$, donde $\Phi^n(x) : x \mapsto (\sqrt{\lambda_1}\psi_1(x), \ldots, \sqrt{\lambda_n}\psi_n(x))$. En tal contexto, se puede interpretar al espacio de características como un espacio finito dimensional dentro de cierta precisión ε. Esta característica es una parte esencial en el aprendizaje con kernels y dará lugar a la técnica conocida como *truco del kernel*.

La condición de simetría para $k(\cdot,\cdot)$, necesaria en el teorema (3.1), permite dotar de esta propiedad al producto interno definido en (3.2). Por otra parte, es necesario caracterizar la positividad de $k(\cdot,\cdot)$ de manera tal que se pueda contextualizar dentro del aprendizaje de máquinas, para ello se utiliza la noción de matriz de **Gram**.

Definición 3.3 (Matriz de Gram)**.** Dado un un kernel $k(\cdot,\cdot)$ y un conjunto de puntos $\mathcal{X}_n = \{x_i \mid i = 1, ..., n\}$, se define la matriz de Gram como la matriz K que verifica $K_{i,j} = k(x_i, x_j)$.

Esta noción, permite definir la positividad de un kernel $k(\cdot,\cdot)$ en función de las propiedades que sus matrices de Gram poseen, en este sentido, el concepto de **matriz semidefinida** juega un rol fundamental.

Definición 3.4 (matriz semidefinida positiva)**.** Una matriz real K de n x n se dice semidefinida positiva (denotado SDP) si para todo vector $\mathrm{v} \in \mathbb{R}^n$ se cumple que $Q(\mathrm{v}) = \mathrm{v}^\top K \mathrm{v} \geq 0$. Si además se cumple que $Q(\mathrm{v}) = 0 \Leftrightarrow \mathrm{v} = 0$, entonces se dice que la matriz K es definida positiva(denotada DP).

Dentro de las propiedades de este tipo de matrices se encuentra la siguiente proposición:

Proposición 1. Una matriz simétrica es semidefinida positiva si y solo si todos sus valores propios son no negativos.

Finalmente, se podrá caracterizar la positividad de un kernel $k(\cdot,\cdot)$ mediante la siguiente proposición:

Proposición 2. Un kernel k es semidefinido positivo si y sólo si toda matriz de Gram K generada con k es semidefinida positiva. En ese caso, se dice que k es una función de covarianza.

En particular, esto significa que cualquier función simétrica y semidefinida positiva, acepta una descomposición por medio del teorema de Mercer y por tanto induce una transformación Φ a un espacio de características (posiblemente de dimensión infinita, $N_{\mathcal{H}} = \infty$) donde actúa como un producto interno. Si por otra parte, se posee una transformación $\Phi : \mathcal{X} \to \mathcal{H}$, donde $\mathcal{H}$ es un espacio con producto interno, es posible construir un kernel por medio de $k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$. Este es por definición, simétrico (lo hereda del producto interno) y además para todo $c_i \in \mathbb{R}$, $x_i \in \mathcal{X}$, $i = 1, ..., m$, cumple

$$\sum_{i,j} c_i c_j k(x_i, x_j) = \left\langle \sum_i c_i \Phi(x_i), \sum_j c_j \Phi(x_j) \right\rangle = \left\| \sum_i c_i \Phi(x_i) \right\|^2 \geq 0$$

Por tanto será semidefindo positivo y finalmente un kernel que cumple las condiciones del teorema (3.1). Esto permite modelar kernels por medio de transformaciones conocidas, a la vez que permite asumir la existencia de una transformación dado un kernel.

3.3. Espacios de Hilbert con kernel reproductor - RKHS

Hasta este punto, se puede entender un kernel como una función semidefinida positiva y simétrica compatible con una representación, ya sea implícita o explicita, de un espacio inicial $\mathcal{X}$ en un espacio con producto interno (o *pre-Hilbertiano*). El propósito de esta sección corresponde a estudiar las características de este espacio con el fin de obtener herramientas para el uso de kernels en tareas de aproximación de funciones.

Definición 3.5 (Espacio de Hilbert con Kernel Reproductor)**.** Sea $\mathcal{X}$ conjunto no vacíoy $\mathcal{H}$ un espacio de Hilbert de funciones $f : \mathcal{X} \to \mathbb{R}$. Se denomina a $\mathcal{H}$ espacio de Hilbert con kernel reproductor con producto punto $\langle \cdot, \cdot \rangle$ (y por tanto la norma $\|f\| := \sqrt{\langle f, f \rangle}$) si existe una función $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ con las siguientes propiedades.

(1) $k(\cdot,\cdot)$ tiene la propiedad de reproductor
$\langle f, k(x,\cdot) \rangle = f(x)$ para todo $f \in \mathcal{H}$, en particular
se verifica: $\langle k(x,\cdot), k(x',\cdot) \rangle = k(x, x')$
(2) $k(\cdot,\cdot)$ genera $\mathcal{H}$, es decir, $\mathcal{H} = \overline{gen\{k(x,\cdot) \mid x \in \mathcal{X}\}}$ donde $\overline{X}$ denota la completación del conjunto X.

Por definición (punto 2), un RKHS es un espacio formado por combinaciones lineales de funciones de la forma $k_x(\cdot)$ para todo $x \in \mathcal{X}$ y sus funciones limite (clausura). Esto último quiere decir, que toda función $f \in \mathcal{H}$ puede ser aproximada

a través de sucesiones de combinaciones lineales de elementos de la forma $k_x(\cdot)$, que claramente están únicamente determinados por el kernel $k(\cdot,\cdot)$.

Por su parte, la propiedad de reproducción (punto 1), se puede comprender al observar que para todo $f \in \mathcal{H}$, espacio RKHS correspondiente al kernel $k(\cdot,\cdot)$, es posible escribir f como:

$$f(\cdot) = \sum_{i=1}^{\infty} \alpha_i k(x_i, \cdot) \tag{3.4}$$

Que por el teorema de Mercer verifica:

$$f(x) = \sum_{i=1}^{\infty} \alpha_i \sum_{j=1}^{N_{\mathcal{H}}} \lambda_j \psi_j(x_i)\psi_j(x) \tag{3.5}$$

Como todo funcional de la forma $k_x(\cdot) = k(x,\cdot) \in \mathcal{H}$ y este espacio posee producto interno, es posible escribir $\langle f(\cdot), k(x,\cdot)\rangle$ de la forma:

$$\langle f, k_x\rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \alpha_i \sum_{j=1}^{N_{\mathcal{H}}} \sum_{l=1}^{N_{\mathcal{H}}} \lambda_j \psi_j(x_i)\lambda_l\psi_l(x) \left\langle \psi_j, \psi_l \right\rangle_{L_2(\mathcal{X})} \tag{3.6}$$

$$= \sum_{i=1}^{\infty} \alpha_i \sum_{j=1}^{N_{\mathcal{H}}} \lambda_j(x_i)\psi_j(x_i)\psi_j(x) \tag{3.7}$$

$$= f(x) \tag{3.8}$$

Donde en la primera ecuación se utiliza la otrogonalidad $\psi_i \perp \psi_j$ en todo $i \neq j$. Posteriormente, se redefine el producto interno en $L_2(\mathcal{X})$ de manera que $\langle \psi_j, \psi_l\rangle = \delta_{ij}/\lambda_j$, finalmente se tiene que [2] :

$$\langle f, k(x,\cdot)\rangle = f(x) \tag{3.9}$$

para todo $f \in \mathcal{H}$. Es decir, en un espacio compuesto por funciones generadas por un kernel de Mercer se cumple la propiedad de reproducción, por lo tanto, si el espacio es completo (en caso contrario se considera la adherencia del conjunto) se posee un RKHS con respecto a tal kernel. Desde este punto vista, se puede también concluir que es posible construir un espacio RKHS con respecto a un espacio $\mathcal{X}$ y un kernel de Mercer, defiendo el producto interno de las funciones generadas por el kernel de manera tal que se cumpla la propiedad de reproducción. Finalmente, se consideran las funciones limite de tal conjunto para construir el RKHS correspondiente.

Por definición para cada espacio RKHS existe una función kernel, por el argumento anterior, se puede intuir la existencia de cierta relación de equivalencia entre los kernels que describen a un espacio RKHS. La respues a tal inquietud es aún más fuerte, pues establece que a cada RKHS corresponde una único kernel [**6**].

Teorema 3.6 (Moore-Aronszajn)**.** Sea $\mathcal{X}$ un conjunto, para cada función definida positiva k en $\mathcal{X} \times \mathcal{X}$ existe un único RKHS asociado a esta. Más aún, cada RKHS posee un único kernel definido positivo asociado.

Debido a esta relación de unicidad, tiene sentido definir al espacio RKHS $\mathcal{H}$ por medio de la base ortogonal conformada por las funciones propias de su kernel.

[2] $\delta_{ij} = 1$ si $i = j$, y $\delta_{ij} = 0$ en caso contrario. Este objeto matemático se conoce como *delta de Kronecker*.

Proposición 3. Sea $\mathcal{H}$ el espacio compuesto por las combinaciones lineales de las funciones propias ortogonales $\{\psi_j\}_j^{N_\mathcal{H}}$ de un kernel $k(\cdot,\cdot)$ con respecto a una medida μ. Toda $f \in \mathcal{H}$, se expresa por tanto como $f(x) = \sum_{i=1}^{N_\mathcal{H}} f_i\psi_i(x)$, si se impone la restricción, $\sum_{i=1}^{N} \frac{f_i^2}{\lambda_i} < \infty$, es posible dotar a $\mathcal{H}$ con el producto interno $\langle f, g\rangle_\mathcal{H} = \sum_{i=1}^{N} \frac{f_i g_i}{\lambda_i}$, donde λ_j es el valor propio asociado a ψ. Si $\mathcal{H}$ es completo, entonces es el RKHS asociado a $k(\cdot,\cdot)$.

De esta forma, se puede formular el siguiente teorema:

Teorema 3.7 (Teorema representador)**.** Dado un kernel k sobre $\mathcal{H}$, seconsidera el espacio de funciones:

$$\mathcal{H}_0 = \left\{ f(x) = \sum_{i=1}^{n_f} f_i k\left(x, x_i\right) | n_f \in \mathbb{N}, x_i \in \mathcal{X}, f_i \in \mathbb{R} \right\}$$

con el producto interno $\langle f, g\rangle_\mathcal{H} = \sum_{i=1}^{n_f}\sum_{j=1}^{n_g} f_i g_j k\left(x_i, x_j\right)$. Luego $\mathcal{H} = \overline{\mathcal{H}}_0$ es el RKHS asociado a $k(\cdot,\cdot)$.

Lo anterior es directo de $f(x) = k_{\overline{x}}(x) \in \mathcal{H}_0,$ donde $n_f = 1, x_1 = \overline{x}$ y $f_i = 1$, de donde se obtiene:

$$\langle f, k_{\overline{x}}\rangle_{\mathcal{H}_0} = \sum_{i=1}^{n_f} f_i k\left(\overline{x}, x_i\right) = f(\overline{x})$$

Dado que $k(\cdot,\cdot)$ es único con respecto a su RKHS.

3.4. Kernels

En la presente sección se estudian distintos kernels. Se observan sus características y se derivan modelos basados en estas.

3.4.1. Kernel RBF. El *kernel exponencial cuadrático* (SE) o *gaussiano* para $\boldsymbol{x} = (x_1, \ldots, x_D)^T \in \mathbb{R}^D$ se define por:

$$k\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \exp\left(-\frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x'}\right)^\top \Sigma^{-1}\left(\boldsymbol{x} - \boldsymbol{x'}\right)\right) \tag{3.10}$$

donde si Σ es diagonal, se reduce a

$$k\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \exp\left(-\frac{1}{2}\sum_{j=1}^{D}\frac{1}{\sigma_j^2}\left(x_j - x_j'\right)^2\right) \tag{3.11}$$

En este tipo de kernels, se puede interpretar σ_j como el **ancho de banda característico** de la componente (característica) j-ésima. En tal contexto, si $\sigma_j{=}\infty$, los valores de tal dimensión pasan a ser ignorados. Por lo tanto, se pueden identificar las características o componentes con menor influencia examinando la magnitud de su ancho de banda correspondiente. Este método se conoce como **Kernel ARD**.

Por otra parte, si Σ es esférica, se obtiene el kernel isotrópico

$$k\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x'}\|^2}{2\sigma^2}\right) \tag{3.12}$$

donde σ Se conoce como **ancho de banda**. Las funciones del tipo (3.12), se conocen como **funciones de base radial**, y se caracterizan por depender unicamente de la diferencia entre los puntos que opera, es decir, $k(\boldsymbol{x}, \boldsymbol{x'}) = k(\boldsymbol{x} - \boldsymbol{x'})$.

Por este motivo el kernel de la ecuación (3.12) se conoce como kernel RBF (Radial Basis Function).

3.4.2. Kernels lineales. Como se estudió en la sección pasada, la obtención de la aplicación de características $\Phi(\cdot)$ asociada a un kernel requería de una descomposición espectral. Sin embargo, el proceso de obtener un kernel desde a partir de una aplicación de características requiere únicamente de:

$$k\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x'}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x'}) \rangle$$

El caso $\phi = I_d$, es decir $\phi(\boldsymbol{x}) = \boldsymbol{x}$ para todo x, se obtiene el **kernel lineal**, este corresponde al producto interno del espacio donde se opera. En el caso $\mathcal{X} = \mathbb{R}^D$ este se define de manera natural por:

$$k\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \boldsymbol{x}^\top \boldsymbol{x'}$$

3.4.3. Matern Kernels. Los **Matern kernel** , corresponden funciones del tipo,

$$k(r) = \frac{2^{(1-v)}}{\Gamma(v)} \left(\frac{\sqrt{2v}r}{l} \right)^v K_v \left(\frac{\sqrt{2v}r}{l} \right)$$

Donde $r = \|\boldsymbol{x} - \boldsymbol{x'}\|$, $v > 0$, $l > 0$ y K_v corresponde a una función de Bessel modificada. Cuando $v \to \infty$ este kernel se aproxima al kernel SE. Por otra parte, si $v = \frac{1}{2}$ el kernel se simplifica a

$$k(r) = exp(-r/l)$$

3.5. Kernels derivados de modelos probabilísticos generativos

En presencia de un modelo generativo probabilístico de vectores representados por $p(\boldsymbol{x}|\theta)$. Es posible utilizar la estructura de tal modelo para construir funciones kernel.

3.5.1. Kernels producto de probabilidad. En el contexto anterior, se define un kernel por medio de la siguiente identidad

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \int p(\boldsymbol{x}|\boldsymbol{x_i})^\rho p(\boldsymbol{x}|\boldsymbol{x_j})^\rho dx \tag{3.13}$$

donde $\rho > 0$ y $p(\boldsymbol{x}|\boldsymbol{x_i})$ se aproxima por $p(\boldsymbol{x}|\widehat{\theta}(\boldsymbol{x_i}))$, donde $\widehat{\theta}(\boldsymbol{x_i}))$ es un parámetro estimado obtenido usando el dato $\boldsymbol{x_i}$. Este kernel es llamado **kernel producto de probabilidad** [**19**].

El modelo ajustado será usado para ver cuan similar son los objetos que se operan. En particular, si se ajusta el modelo a $\boldsymbol{x_i}$ se desea que este sea capaz de indicar cuando otro dato $\boldsymbol{x_j}$ es similar. Si a modo de ejemplo, se supone $p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{N}(\mu, \sigma^2 \boldsymbol{I})$ donde σ^2 es fijo. Con $\rho = 1$ y $\widehat{\mu}(\boldsymbol{x_i}) = \boldsymbol{x_i}$ y $\widehat{\mu}(\boldsymbol{x_j}) = \boldsymbol{x_j}$, el kernel

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \frac{1}{(4\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{(4\pi\sigma^2)} \|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2 \right) \tag{3.14}$$

el cual corresponde a kernel de base radial. Se puede calcular la ecuación 3.13 por una variedad de modelos generativos, incluyendo aquellos con variables latentes.

3.5.2. Kernels Fisher. Una manera más eficiente para usar modelos generativos en definición kernels es usar **Kernel Fisher** [**8**] definido por

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{g}(\boldsymbol{x})^\top \boldsymbol{F}^{-1} \boldsymbol{g}(\boldsymbol{x'})$$

donde $\boldsymbol{g}$ es el gradiente de la log verosimilitud o **vector de puntaje**, evaluado en el estimador MLE $\widehat{\theta}$:

$$\boldsymbol{g}(\boldsymbol{x}) \triangleq \nabla_\theta \log p(\boldsymbol{x}|\theta)|_{\widehat{\theta}}$$

$\boldsymbol{F}$ corresponde a la matriz de información de Fisher, la cual es corresponde esencialmente a la matriz Hessiana:

$$\boldsymbol{F} = \nabla\nabla \log p(\boldsymbol{x}|\theta)|_{\widehat{\theta}}$$

Debido a que el estimador $\widehat{\theta}$ se obitne en función de todos los datos, se tiene que similitud de $\boldsymbol{x}$ y $\boldsymbol{x'}$ es por tanto calculada usando todos los datos.

Si $\boldsymbol{g}(\boldsymbol{x})$ es la dirección en la cual los datos $\boldsymbol{x}$ condicionan el valor de $\widehat{\theta}$) al momento de maximizar la verosimilitud. Se desea considerar como similares a los puntos $\boldsymbol{x}$ y $\boldsymbol{x'}$ si las direcciones para el gradiente que condicionan la verosimilitud en θ son similares entre si.

3.6. Kernels en modelos lineales generalizados

Con las herramientas estudiadas, es posible definir una clase de modelos basados en kernel para tareas de clasificación y regresión.

3.6.1. Máquinas de kernel. Se define una **Máquina de kernel** como un modelo lineal generalizado (GLM) donde el vector de características tiene la forma:

$$\boldsymbol{\phi}(\boldsymbol{x}) = [k(\boldsymbol{x}, \mu_1), ..., k(\boldsymbol{x}, \mu_K)] \tag{3.15}$$

donde $\mu_k \in \mathcal{X}$ son un conjunto de K **centroides**. Si k es un kernel RBF, este modelo se denota por una red RBF. El vector de características presente en la ecuación 3.15 se denota como **vector de características kernelisado**. En este enfoque, no se requiere que el kernel sea Mercer.

Se puede usar el vector de características kernelizado entrenar un modelo de regresión logística definiendo $p(y|\boldsymbol{x}, \theta) = Ber(\boldsymbol{w}^\top \phi(\boldsymbol{x}))$. Esta formulación permite definir bordes de descición no lineales de manera sencilla a traves de la elección del kernel.

De la misma forma, es posible usar el vector de características anterior para entrenar un modelo de regresión lineal al definir $p(y|\boldsymbol{x}, \theta) = \mathcal{N}(\boldsymbol{w}^\top \phi(\boldsymbol{x}), \sigma^2)$.

3.6.2. LIVMs, RVMs y Maquinas de vectores sparse. Al formular modelos basados en Maquinas de kernel, no se especifica la manera optima para la obtención de centroides μ_k. Un enfoque consiste en encontrar clusters en los datos y luego asignarlos como prototipos. Sim embargo, no se puede garantizar que a través de esta búsqueda se encuentren representaciones convenientes para la predicción. A esto se suma la elección del número de clusters, que tampoco esta especificada. Otro enfoque es considerar hacer cada elemento $\boldsymbol{x_i}$ como un prototipo, de esta forma se obtiene

$$\boldsymbol{\phi}(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), ..., k(\boldsymbol{x}, \boldsymbol{x}_N)]$$

De esta forma, se tienen tantos parámetros con datos. Si bien esto supone un problema desde el punto de vista computacional, es posible subsanarlo por medio del uso de distribuciones prior $\boldsymbol{w}$ que induzcan dispersión (sparsity) en el regresor final, de esta forma, no solo se resuelve el problema referente a la alta cantidad de parámetros, sino que también la dispersión inducida en el vector de características permite escoger aquellos puntos más relevantes, permitiendo seleccionar un subconjunto de los datos para predecir. Los modelos que hace uso de este principio se denominan **Maquinas de vectores sparse**.

En tal sentido, la elección más natural al momento de inducir dispersión es el uso de regularización l_1 [**9**]. A la maquina de vectores resultante del uso de este regularizador para inducir dispersión se le denominará **LIVM** (Maquina de vectores regularizada l_1) . Análogamente, el uso de un regularizador l_2 da lugar a **L2VM** (máquina de vectores l_2-regularizada), en este último modelo no se induce dispersión (debido al tipo de regularización).

Es también posible implementar el método ARD para inducir dispersión, el método resultante se denomina **Máquina de vectores de relevancia** o **RVM** [**10**].

Otra manera maquina de vector sparse los lo modelos conocidos como **máquinas de vectores de soporte** (SVM)

3.6.3. Maquinas de vectores de soporte (SVM). Este método se desarrolló para resolver problemas de clasificación y aproximación estática de funciones [**14**] [**15**], tanto para problemas de regresión como para reconocimiento de patrones. La formulación de esta metodología, adaptada a la estimación no lineal de funciones, comienza considerando una regresión de la forma:

$$f(x) = w^T \varphi(x) + b \tag{3.16}$$

En este contexto, la aplicación $\varphi : \mathbb{R}^n \to \mathbb{R}^{N_{\mathcal{H}}}$ corresponde la transformación de caracterisitcas sobre los datos. El parametro $w \in \mathbb{R}^{N_{\mathcal{H}}}$ corresponde a al vector de pesos asociado a la construcción del modelo y cuya dimensión depende directamente de φ, pudiendo ser infinito dimensional.

Se busca estimar minimizando el riesgo empírico $\mathbf{R}_{emp}$ definido por:

$$\mathbf{R}_{emp} = \frac{1}{N} \sum_{i=1}^{N} \left| y_i - w^T \varphi(x_i) - b \right|_{\varepsilon} \tag{3.17}$$

Donde se emplea la función de perdida ε-insensible de Vapnik, definida por:

$$|y - f(x)|_{\varepsilon} = \begin{cases} 0, & \text{si } |y - f(x)| \leq \varepsilon \\ |y - f(x)| - \varepsilon, & \text{en caso contratrio} \end{cases} \tag{3.18}$$

Una primera aproximación a un modelo de entrenamiento, viene dada por el siguiente problema de optimización:

$$\left[\begin{array}{l} \boxed{P} : \min\limits_{w,b} J_P(w) = \dfrac{1}{2} w^T w \\ \quad \text{s.a } y_i - w^T \varphi(x_i) - b \leq \varepsilon, \ i = 1, \ldots, N \\ \quad\quad w^T \varphi(x_i) + b - y_i \leq \varepsilon, \ i = 1, \ldots, N \end{array} \right] \tag{3.19}$$

Dicha formulación corresponde al caso en que todos los datos de entrenamiento se encuentran dentro una banda de ancho ε. Como es de esperar, elegir una precisión muy alta puede dejar puntos de entrenamiento fuera de dicha banda, lo cual hace que el problema (3.19) se vuelva infactible. Para solucionar tal inconveniente y obtener un modelo más flexible, se agregan variables de holgura ξ_i , ξ_i^* para $i = 1, \ldots, N$, de esta manera se permiten ciertos puntos fuera de la precisión preestablecida, al agregar dichas variables, se obtiene el siguiente problema:

$$\left[\begin{array}{l} \boxed{P}: \min\limits_{w,b,\xi,\xi^*} J_P(w,\xi,\xi^*) = \dfrac{1}{2}w^T w + c\sum\limits_{i=1}^{N}(\xi_i + \xi_i^*) \\ \quad \text{s.a } y_i - w^T\varphi(x_i) - b \leq \varepsilon + \xi_i,\ i = 1,\ldots,N \\ \qquad w^T\varphi(x_i) + b - y_i \leq \varepsilon + \xi_i^*,\ i = 1,\ldots,N \\ \qquad \xi_i, \xi_i^* \geq 0,\ k = 1,\ldots,N \end{array}\right] \tag{3.20}$$

Donde la constante $c > 0$ determina la proporción de *desviaciones* toleradas a partir de cierta precisión ε. El Lagrangiano de este problema es:

$$\begin{aligned} &\mathcal{L}(w,b,\xi,\xi^*;\alpha,\alpha^*,\eta,\eta^*) = \\ &\frac{1}{2}w^T w + c\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i - y_i + w^T\varphi(x_i) + b) \\ &-\sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* + y_i - w^T\varphi(x_i) + b) - \sum_{i=1}^{N}(\eta_i\xi_i + \eta_i^*\xi_i^*) \end{aligned} \tag{3.21}$$

Donde los factores $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0$ son los multiplicadores de Lagrange correspondientes. El punto estacionario de (3.21) se caracteriza de la siguiente manera:

$$\max_{\alpha,\alpha^*,\eta,\eta^*} \min_{w,b,\xi,\xi^*} \mathcal{L}(w,b,\xi,\xi^*;\alpha,\alpha^*,\eta,\eta^*) \tag{3.22}$$

De las condiciones de optimalidad se observa:

$$\left[\begin{array}{l} \dfrac{\partial\mathcal{L}}{\partial w} = 0 \rightarrow w = \sum\limits_{i=1}^{N}(\alpha_i - \alpha_i^*)\varphi(x_i) \\ \dfrac{\partial\mathcal{L}}{\partial b} = 0 \rightarrow \sum\limits_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ \dfrac{\partial\mathcal{L}}{\partial\xi_i} = 0 \rightarrow c - \alpha_i - \eta_i = 0 \\ \dfrac{\partial\mathcal{L}}{\partial\xi_i^*} = 0 \rightarrow c - \alpha_i^* - \eta_i^* = 0 \end{array}\right] \tag{3.23}$$

De lo cual se obtiene el siguiente problema dual:

$$\left[\begin{array}{l} \boxed{D}: \max\limits_{\alpha,\alpha^*} J_D(\alpha,\alpha^*) = -\frac{1}{2}\sum\limits_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\varphi(x_i)^T\varphi(x_j) \\ \qquad\qquad -\varepsilon\sum\limits_{i=1}^{N}(\alpha_i+\alpha_i^*)+\sum\limits_{i=1}^{N}y_i(\alpha_i-\alpha_i^*) \\ \text{s.a } \sum\limits_{i=1}^{N}(\alpha_i-\alpha_i^*)=0 \\ \qquad \alpha_i,\alpha_i^*\in[0,c],\ i=1,...,N \end{array}\right] \tag{3.24}$$

Finalmente la representación dual del modelo de regresión es:

$$f(x)=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)\varphi(x_i)^T\varphi(x)+b \tag{3.25}$$

Los datos de entrada x_i tales que la diferencia $\alpha_i-\alpha_i^*$ es no nula, son los llamados *vectores de soporte*. La estructura de (3.25) permite hacer la regresión sólo sobre estos elementos, por lo que este método induce o dispersión en la estimación (3.25).

Esta formulación responde a un esquema de regresión no lineal donde de induce dispersión por medio de restricciones, la formulación presentada presenta la gran desventaja de ser incalculable cuando la dimensión del espacio de características $N_{\mathcal{H}}$ es infinita. Usando los resultados obtenidos anteriormente, es posible reformular por medio de la herramienta conocida como el **truco del kernel** [**12**].

El truco del kernel consiste en reemplazar el calculo de productos internos por una función kernel haciendo uso del resultado (2). En el caso de la formulación SVM, es posible reemplazar $\varphi(x_i)^T\varphi(x_j)$ por una función kernel de la forma $k(x_i,x_j)=\varphi(x_i)^T\varphi(x_j)$. La estructura de este kernel no se conoce a piori, esto pues la definición de $\varphi(\cdot)$ fue netamente teórica. Sin embargo, se sabe que su producto interno se puede representar por un kernel. La idea detrás de este método consiste en seleccionar un kernel de manera tal que las funciones base de su RKHS subyacente permitan una aproximación lo suficientemente flexible y regular.

Al aplicar el truco del kernel a los modelos de entrenamiento y regresión obtenidos anteriormente para SVM, se obtiene la versión *kernelizada*:

$$\left[\begin{array}{l} \boxed{D}: \max\limits_{\alpha,\alpha^*} J_D(\alpha,\alpha^*) = -\frac{1}{2}\sum\limits_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(x_i,x_j) \\ \qquad\qquad -\varepsilon\sum\limits_{i=1}^{N}(\alpha_i+\alpha_i^*)+\sum\limits_{i=1}^{N}y_i(\alpha_i-\alpha_i^*) \\ \text{s.a } \sum\limits_{i=1}^{N}(\alpha_i-\alpha_i^*)=0 \\ \qquad \alpha_i,\alpha_i^*\in[0,c],\ i=1,...,N \end{array}\right] \tag{3.26}$$

Con su respectiva formula de regresión dual:

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) k(x_i, x) + b \tag{3.27}$$

3.6.4. LS-SVM. Una variante del método SVM es su versión basada en minimos cuadrados, la cual se formula como una alternativa más eficiente[**13**].

Su formulación es similar a la expuesta en 3.20, considerando la formula habitual de regresión presentada en (3.16):

$$\left[\begin{array}{l} \boxed{P} : \min\limits_{w,b,e} J_P(w,e) = \dfrac{1}{2} w^T w + \gamma \dfrac{1}{2} \sum\limits_{i=1}^{N} e_i^2 \\ \\ \text{s.a } y_i = w^T \varphi(x_i) + b + e_i \ , \ i = p, \ldots, N \end{array}\right] \tag{3.28}$$

En esta nueva formulación, los términos de error e_i juegan un papel similar al de las variables de holgura ξ_i, ξ_i^* en (3.20), con excepción de que la función de perdida que se aplica a estas variables es ahora cuadrática. Se comienza entonces por establecer el Lagrangiano:

$$\mathcal{L}(w, b, e; \alpha) = J_P(w, e) - \sum_{i=1}^{N} \alpha_i (w^T \varphi(x_i) + b - e_i - y_i) \tag{3.29}$$

Donde los coeficientes α_i son los multiplicadores de Lagrange. Las condiciones de optimalidad obtenidas son las siguientes:

$$\left[\begin{array}{ll} \dfrac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \sum\limits_{i=1}^{N} \alpha_i \varphi(x_i) & \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum\limits_{i=1}^{N} \alpha_i = 0 & \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i & i = 1, ..., N \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i - y_i = 0 & i = 1, ..., N \end{array}\right. \tag{3.30}$$

Al despejar w, e_i y reemplazar en la ultima igualdad se obtiene:

$$\sum_{i=1}^{N} \alpha_i \varphi(x_i)^T \varphi(x_j) + b + \frac{\alpha_j}{\gamma} = y_j \ j = 1, ..., N \tag{3.31}$$

Usando el truco del kernel en la forma:

$$\mathbf{K}_{ij} = k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \ j = 1, ..., N \tag{3.32}$$

Reescribiendo (3.31) y haciendo uso de la notación $\mathbf{y} = [y_1, ..., y_N]^T$, $\mathbf{1}_N = [1, ..., 1]^T$, $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^T$, la condición $\sum_{i=1}^{N} \alpha_i = 0$ obtenida en (3.30) y el resultado (3.32), se elabora el siguiente problema:

$$\left[\begin{array}{l} \boxed{D}: \text{Resolver para } \boldsymbol{\alpha},\ b \\ \qquad \left[\begin{array}{c|c} 0 & \mathbf{1}_N^T \\ \hline \mathbf{1}_N & \mathbf{K}+\gamma^{-1}\mathbf{I} \end{array}\right] \left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{y} \end{array}\right] \end{array}\right] \tag{3.33}$$

Haciendo nuevamente uso de las condiciones obtenidas en (3.30), se puede verificar que el estimador obtenido es bastante similar en estructura al expuesto en (3.27), siendo este:

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i) + b \tag{3.34}$$

El proceso de entrenamiento a través de (3.33), se ve facilitado numéricamente, ya que se basa en la solución de un sistema de ecuaciones lineales, el cual presenta una solución única cuando es de rango completo y que además, en su forma dual, no depende de la dimensión del espacio de características. Estas ventajas, sin embargo, no son gratuitas, pues como es posible observar en la obtención de las condiciones de primer orden del Lagrangiano (3.30), la ecuación $\alpha_i = \gamma e_i$ anula la característica de *dispersión* que posee la formulación tradicional, esto se debe a que en este nuevo escenario, se dificulta la existencia de α_i exactamente iguales a 0, dando el carácter de *vector de soporte* a todos los elementos presentes en el conjunto de entrenamiento, cada uno con cierto factor α_i de influencia en el modelo.

3.6.5. Clasificación kernel knn. En un clasificador 1-NN se necesita calcular la distancia Euclidiana de un vector test a todos los puntos en entrenamiento, para luego encontrar al mas cercano, y asignar su etiqueta. Esto puede ser kernelizado al observar:

$$\|x_i x_{i'}\|_2^2 = \langle x_i, x_i \rangle + \langle x_{i'}, x_{i'} \rangle - 2 \langle x_i, x_{i'} \rangle \tag{3.35}$$

Esta reformulación permite la aplicación del clasificador knn una clase más amplia de problemas.

3.6.6. Clustering de K-medoides Kernelizado. El clustering K-medias al, igual que knn, usa la distancia euclidiana para medir similitud.

Para kernelizar este algoritmo, se reemplaza el algoritmo K-medias con el algoritmo **K-medoides**. El cual es similar a K-medias, pero en vez de representar cada centroide del cluster por la media de todos los vectores de datos asignados a este cluster, se hace que cada centroide sea uno de los vectores de datos por si mismo. En este caso se busca resolver:

$$m_k = \underset{i:z_i=k}{\arg\min} \sum_{i':z_{i'}=k} d(i, i')$$

donde

$$d(i, i') \triangleq \|\boldsymbol{x_i} - \boldsymbol{x_i'}\|_2^2$$

Esto toma $O(n_k^2)$ trabajo por cluster, mientras K-medias toma $O(n_k D)$ para actualizar cada cluster. En el algoritmo (5) se aprecia la estrucutra del método.

Al calular el medoide más cercano a la clase, el problema se transforma en **clasificación medoide mas cercano**[**11**] Este algoritmo puede ser kernelizado usando Ecuación (3.35) para reemplazar la distancia $d(i, i')$.

Algorithm 5: Algorimo kernel K-medoids

Inicializar: $m_1, \ldots, m_K$ subconjunto aleatorio de tamaño K en $\{1, \ldots, N\}$;
repeat
  $z_i = \arg\min_k d(i, m_k)$ para $i = 0, \ldots, N$
  $m_k \leftarrow \arg\min_{i:z_i=k} \sum_{i':z_{i'}=k} d(i, i')$
until convergencia

3.6.7. Kernel PCA. PCA requiere encontrar vectores propios de la matriz de covarianza muestral $\boldsymbol{S} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x_i}\boldsymbol{x_i}^\top = (1/N)\boldsymbol{X}^\top\boldsymbol{X}$. Sin embargo, se puede también calcular PCA, encontrando los vectores propios de la matriz de producto interno $\boldsymbol{X}\boldsymbol{X}^\top$, esto permitirá derivar el método **kernel PCA**[**7**].

Primero, sea $\boldsymbol{U}$ la matriz ortogonal conteniendo los vectores propios de $\boldsymbol{X}\boldsymbol{X}^\top$ con los correspondientes valores propios en Λ. Por definición, se tiene $(\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{U} = \boldsymbol{U}\Lambda$. Premultiplicando por $\boldsymbol{X}^\top$:

$$(\boldsymbol{X}^\top\boldsymbol{X})(\boldsymbol{X}^\top\boldsymbol{U}) = (\boldsymbol{X}^\top\boldsymbol{U})\Lambda$$

de donde se ve que los vectores propios de $(\boldsymbol{X}^\top\boldsymbol{X})$ son $\boldsymbol{V} = \boldsymbol{X}^\top\boldsymbol{U}$ con los valores propios dados por Λ. Sin embargo, tales vectores propios no se encuentran normalizados. Dado $\|v_j\|^2 = \boldsymbol{u_j}^\top\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{u_j} = \lambda_j\boldsymbol{u_j}^\top\boldsymbol{u_j} = \lambda_j$. por lo que es posible obtener un vector propio normalizado por medio de $\boldsymbol{V_{pca}} = \boldsymbol{X}^\top\boldsymbol{U}\Lambda^{-1/2}$.

Sea ahora $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^\top$ la matriz de Gram del kernel $k(\cdot,\cdot)$. Sea Φ la matriz diseño correspondiente a la transformación de características inducida por $k(\cdot,\cdot)$. $\boldsymbol{S_\phi} = \frac{1}{N}\sum_i \boldsymbol{\phi_i}\boldsymbol{\phi_i}^\top$ la correspondiente matriz de covarianza en el espacio de características. Los vectores propios son dados por $\boldsymbol{V_{kpca}} = \boldsymbol{\Phi}^\top\boldsymbol{U}\Lambda^{-1/2}$ donde $\boldsymbol{U}$ y Λ contienen los vectores propios y valores propios de $\boldsymbol{K}$. Claramente no es posible calcular $\boldsymbol{V_{kpca}}$,dado que $\boldsymbol{\Phi_i}$ es potencialmente de dimensión infinita. Sin embargo, se puede calcular la proyección de un vector $\boldsymbol{x}$ sobre el espacio característica como sigue.

$$\phi_*^T\mathbf{V}_{kpca} = \phi_*^T\boldsymbol{\Phi}\mathbf{U}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \mathbf{k}_*^T\mathbf{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}$$

donde $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_N)]$.

Hasta este punto, se asume que los datos proyectados tienen media cero, lo cual en general no es el caso. Para sortear tal detalle, se define el vector de características centrado como $\tilde{\boldsymbol{\phi}}_i = \phi(\mathbf{x}_i) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\phi}(\mathbf{x}_j)$. La matriz Gram de vectores característicos centrados esta dada por:

$$\begin{aligned}\tilde{K}_{ij} &= \tilde{\phi}_i^T \tilde{\phi}_j \\ &= \phi_i^T \phi_j - \frac{1}{N}\sum_{k=1}^{N} \phi_i^T \phi_k - \frac{1}{N}\sum_{k=1}^{N} \phi_j^T \phi_k + \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{M} \phi_k^T \phi_l\end{aligned}$$

De donde

$$\tilde{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{N}\sum_{k=1}^{N} k(\boldsymbol{x}_i, \boldsymbol{x}_k) - \frac{1}{N}\sum_{k=1}^{N} k(\boldsymbol{x}_j, \boldsymbol{x}_k) + \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{M} k(\boldsymbol{x}_k, \boldsymbol{x}_l)$$

Lo cual puede ser expresado en notación matricial como:

$$\tilde{\mathbf{K}} = \mathbf{HKH}$$

donde $\mathbf{H} \triangleq \mathbf{I} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ es la **matriz centrada**. El proceso seguido anteriormente, se resume en el algoritmo 6.

Algorithm 6: kernel PCA

Input: $\boldsymbol{K}$ de tamaño $N \times N$, $\boldsymbol{K}_*$ de tamaño $N_* \times N$, numero de dimenciones latentes L

$\boldsymbol{O} = \mathbf{1}_N\mathbf{1}_N^T/N$
$\tilde{\mathbf{K}} = \boldsymbol{K} - \boldsymbol{OK} - \boldsymbol{KO} + \boldsymbol{OKO}$
$[\boldsymbol{U}, \boldsymbol{\Lambda}] = \text{eig}\tilde{\mathbf{K}}$
for $i = 1, \ldots, N$ **do**
  $\boldsymbol{v}_i = \boldsymbol{u}_i/\sqrt{\lambda_i}$
end
$\boldsymbol{O}_* = \mathbf{1}_{N_*}\mathbf{1}_N^T/N$
$\tilde{\mathbf{K}} = \boldsymbol{K}_* - \boldsymbol{O}_*\boldsymbol{K}_* - \boldsymbol{K}_*\boldsymbol{O}_* + \boldsymbol{O}_*\boldsymbol{K}_*\boldsymbol{O}_*$
$Z = \tilde{\mathbf{K}}\boldsymbol{V}(:, 1:L)$;

3.7. Kernels para construir modelos generativos

Otra case kernels, conocidos como kernels de suavizado, permiten caracterizar estimaciones no paramétricas de desidades. Lo cual puede ser utilizado para generar estimaciones de densidades de manera no supervisada, así como también para crear modelos generativos tanto para regresión como para clasificación.

3.7.1. Kernel de suavizado. Un **kernel suavizado** es una función de un argumento que satisface las siguientes propiedades:

$$\int k(x)dx = 1, \int xk(x)dx = 0, \int x^2k(x)dx > 0$$

Un ejemplo sencillo es el **kernel gaussiano**:

$$k(x) \triangleq \frac{1}{(2\pi)^{\frac{1}{2}}} e^{-x^2/2}$$

En este caso, se puede controlar el ancho de banda del kernel introduciendo un parámetro h:

$$k_h(x) \triangleq \frac{1}{h} k\left(\frac{x}{h}\right)$$

Esto puede ser generalizado a un kernel RBF:

$$k_h(\mathbf{x}) = k_h(\|\mathbf{x}\|)$$

No obstante, en el caso del kernel gaussiano, el kernel de suavizado pasa a ser:

$$k_h(\mathbf{x}) = \frac{1}{h^D (2\pi)^{D/2}} \prod_{j=1}^{D} \exp\left(-\frac{1}{2h^2} x_j^2\right)$$

Si bien los kernels de la familia gausiana poseen propiedades que permiten simplicar el gasto computacional, su soporte no es acotado, por lo que en modelos reales carece de expresividad. Una alternatica a esto es el **kernel Epanechnikov**, el cual posee soporte compacto y esta definido por:

$$k(x) \triangleq \frac{3}{4}\left(1 - x^2\right) \mathbb{I}(|x| \leq 1)$$

Desafortunadamente, el kernel Epanechnikov no es diferenciable en los bordes del soporte. Una alternativa es el **kernel tri-cubo**, defino como sigue:

$$k(x) \triangleq \frac{70}{81}\left(1 - |x|^3\right)^3 \mathbb{I}(|x| \leq 1)$$

Este tiene soporte compacto y tiene dos derivadas continuas en los bordes del soporte. Por otra parte, se puede obtener un kernel a partir de la distribución uniforme:

$$k(x) \triangleq \mathbb{I}(|x| \leq 1)$$

Denominado **kernel boxcar**.

3.7.2. kernel density estimation (KDE). El modelo GMM estudiado anteriormente corresponde a un estimador de densidad paramétrico para datos en $\mathbb{R}^D$. Este requiere especificar el número de componentes K y ubicaciones $\boldsymbol{\mu_k}$ de los cluster. Una alternativa para la estimación de $\boldsymbol{\mu_k}$ es ubicar un cluster central por punto de datos, tal que $\boldsymbol{\mu}_i = \mathbf{x}_i$. En este caso el modelo se convierte:

$$p(\mathbf{x}|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma^2 \mathbf{I})$$

Lo cual se puede generalizar escribiendo:

$$\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} k_h\left(\mathbf{x} - \mathbf{x}_i\right)$$

Modelo conocido como estimador **Parzen window density** , o **estimador de densidad de kernel (KDE)** , y corresponde a una formulación no paramétrica. La ventaja sobre modelos paramétricos es que no se requiere el modelo ajustado y no existe la necesidad de elegir K. La desventaja es que el modelo toma mucho almacenamiento de memoria y gran tiempo para evaluar.

3.7.3. De KDE a KNN. Se puede usar KDE para definir las clases de densidades condicionales en un clasificador generativo. Esto proporciona una derivación alternativa del clasificador KNN. En KDE con un kernel boxcar se fijara el ancho de banda y se contará cuantos puntos caen dentro del hyper cubo centrado sobre cada dato. Si se modela el ancho de banda h como un parámetro de modelo, se permite que el ancho de banda o volumen sea distinto para cada dato. Específicamente, se aumentará el volumen alrededor de $\boldsymbol{x}$ hasta encontrar K datos, sin importar su clase. Sea el volumen resultante V con $V(x)$ y $N_c(x)$ ejemplos de clase c en tal volumen. Luego es posible estimar la densidad condicional la de clase de como sigue:

$$p(\mathbf{x}|y=c,\mathcal{D}) = \frac{N_c(\mathbf{x})}{N_c V(\mathbf{x})}$$

Donde N_c es el número total de ejemplos en la clase c en aquel conjunto de datos. La clase a priori puede ser modelada como

$$p(y=c|\mathcal{D}) = \frac{N_c}{N}$$

Por lo tanto, la posterior de clase esta dada por:

$$p(y=c|\mathbf{x},\mathcal{D}) = \frac{\frac{N_c(\mathbf{x})}{N_c V(\mathbf{x})}\frac{N_c}{N}}{\sum_{c'}\frac{N_{c'}(\mathbf{x})}{N_{c'}V(\mathbf{x})}\frac{N_{c'}}{N}} = \frac{N_c(\mathbf{x})}{\sum_{c'} N_{c'}(\mathbf{x})} = \frac{N_c(\mathbf{x})}{K}$$

Donde se usa el hecho que $\sum_c N_c(\mathbf{x}) = K$, dado que se elige un total de K puntos alrededor de cada punto.

3.7.4. Regresión por kernel. Es posible usar KDE para regresión. El objetivo es calcular la esperanza condicionada

$$f(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}] = \int y p(y|\mathbf{x})dy = \frac{\int y p(\mathbf{x},y)dy}{\int p(\mathbf{x},y)dy}$$

Se puede usar KDE para aproximar la densidad conjunta $p(\mathbf{x},y)$ como sigue

$$p(\mathbf{x},y) \approx \frac{1}{N}\sum_{i=1}^{N} k_h\left(\mathbf{x}-\mathbf{x}_i\right) k_h\left(y-y_i\right)$$

Por lo tanto

$$\begin{aligned} f(\mathbf{x}) &= \frac{\frac{1}{N}\sum_{i=1}^{N} k_h\left(\mathbf{x}-\mathbf{x}_i\right)\int y k_h\left(y-y_i\right)dy}{\frac{1}{N}\sum_{i=1}^{N} k_h\left(\mathbf{x}-\mathbf{x}_i\right)\int k_h\left(y-y_i\right)dy} \\ &= \frac{\sum_{i=1}^{N} k_h\left(\mathbf{x}-\mathbf{x}_i\right) y_i}{\sum_{i=1}^{N} k_h\left(\mathbf{x}-\mathbf{x}_i\right)} \end{aligned}$$

Lo cual se obtiene al observar que la integral $\int k_h\left(y-y_i\right)dy = 1$. Ademàs del hecho que $\int y k_h\left(y-y_i\right)dy = y_i$. Esto sigue definiendo $x = y - y_i$ y usando la propiedad de media igual a cero de kernel suavizado:

$$\int \left(x+y_i\right) k_h(x)dx = \int x k_h(x)dx + y_i\int k_h(x)dx = 0 + y_i = y_i$$

Se puede reescribir el resultado anterior de la siguiente manera:

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}) y_i$$

$$w_i(\mathbf{x}) \triangleq \frac{k_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{i'=1}^{N} k_h(\mathbf{x} - \mathbf{x}_{i'})}$$

La prediccón es por tanto una suma ponderada de los resultados de los datos de entrenamiento, donde los pesos dependen de que tan similar es $\mathbf{x}$ de los puntos de entrenamientos almacenados. Este método es llamado **regresión kernel**, **kernel suavizado** o el modelo **Nadaraya-Watson**.

3.7.5. Regresión localmente ponderada. Si se define $k_h(\mathbf{x} - \mathbf{x}_i) = k(\mathbf{x}, \mathbf{x}_i)$ se puede reescribir la predicción hecha mediante regresiòn kernel como sigue

$$\hat{f}(\mathbf{x}_*) = \sum_{i=1}^{N} y_i \frac{k(\mathbf{x}_*, \mathbf{x}_i)}{\sum_{i'=1}^{N} k(\mathbf{x}_*, \mathbf{x}_{i'})}$$

Si en este caso $k(\mathbf{x}, \mathbf{x}_i)$ fuera un kernel suavizado, se podrìa ignorar el termino normalización, en tal caso:

$$\hat{f}(\mathbf{x}_*) = \sum_{i=1}^{N} y_i k(\mathbf{x}_*, \mathbf{x}_i)$$

Este modelo consiste esencialmente en una función fija localmente constante. Este modelo se puede mejorar fijando un modelo de regresión lineal para cada punto $\mathbf{X}_*$ resolviendo:

$$\min_{\beta(\mathbf{x}_*)} \sum_{i=1}^{N} k(\mathbf{x}_*, \mathbf{x}_i) \left[y_i - \boldsymbol{\beta}(\mathbf{x}_*)^T \boldsymbol{\phi}(\mathbf{x}_i) \right]^2$$

donde $\phi(\mathbf{x}) = [1, \mathbf{x}]$. Esto es llamado **regresión ponderada localmente**. Un ejemplo de tal metodo es **LOESS** o **LOWES** que significa para "locally-weighted scatterplot smoothing". Se puede calcular el parámetro $\boldsymbol{\beta}(\mathbf{x}_*)$ para cada caso de test resolviendo el siguiente problema de mínimos cuadrados ponderados:

$$\boldsymbol{\beta}(\mathbf{x}_*) = \left(\boldsymbol{\Phi}^T \mathbf{D}(\mathbf{x}_*) \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^T \mathbf{D}(\mathbf{x}_*) \mathbf{y}$$

donde Φ es una matriz de diseño de $N \times (D+1)$ y $\mathbf{D} = \text{diag}(k(\mathbf{x}_*, \mathbf{x}_i))$.La predicción correspondiente tiene la forma

$$\hat{f}(\mathbf{x}_*) = \phi(x_*)^T \beta(\mathbf{x}_*) = \left(\boldsymbol{\Phi}^T \mathbf{D}(\mathbf{x}_*) \Phi \right)^{-1} \boldsymbol{\Phi}^T \mathbf{D}(\mathbf{x}_*) \mathbf{y} = \sum_{i=1}^{N} w_i(\mathbf{x}_*) y_i$$

El termino $w_i(\mathbf{x}_*)$ combina el kernel local suavizado con el efecto de la regresión lineal, es llamado el **kernel equivalente** .

CHAPTER 4

Discusión

En este reporte se busco compilar ciertos tópicos de interés pertenecientes al aprendizaje automático. En especifico se abordaron temas referentes a modelos gráficos probabilísticos, haciendo hincapié en modelos gráficos dirigidos. En este contexto, el paso natural es expandir dicha sección para tratar temas relacionados a modelos gráficos probabilísticos no dirigidos y grafos factor. En segunda instancia se abordaron métodos de inferencia aproximada basados tanto en MCMC como en inferencia variacional con enfoque en el algoritmo CAVI. En cuanto a la tercera y ultima sección, se revisaron las bases y propiedades referentes al los métodos basados en kernel.

Para futuras versiones de esta monografía, se propone desarrollar teoría relacionada a modelos bayesianos en aprendizaje automático.

Trabajar en este proyecto me permitió conocer nuevas áreas y perspectivas en aprendizaje automático. Doy las gracias a Felipe Tobar por su iniciativa y por permitirme desarrollar mis habilidades en este campo.

APPENDIX A

Anexos

A.0.1. Tópicos de Análisis.

A.0.1.1. *Normas.* Una norma sobre $\mathbb{R}^n$ es una funcion $f(x) : \mathbb{R}^n \to \mathbb{R}$ que satisface las siguientes propiedades:

(1) $f(x) \geq 0$	$x \in \mathcal{R}^n$ ($f(x) = 0$ si y solo si $x = 0$)
(2) $f(x+y) \leq f(x) + f(y)$	$x, y \in \mathcal{R}^n$
(3) $f(\alpha x) =\mid \alpha \mid f(x)$	$\alpha \in \mathcal{R}, x \in \mathcal{R}^n$

Una norma $f(\cdot)$ cumpliendo lo anterior, se denota por $f(x) = ||x||$. Ejemplos de normas son:

$$\begin{aligned} ||x||_p &= (\mid x_1^p \mid + ... + \mid x_n^p \mid)^{1/p},\ p \geq 1 \\ ||x||_1 &=\mid x_1 \mid + ... + \mid x_n \mid \\ ||x||_2 &= (x^T x)^{1/2} \\ ||x||_\infty &= max_{1 \leq i \leq n} \mid x_i \mid \end{aligned}$$

En la práctica, si no se especifica la norma $||x||$ se asume por la norma-2 $||x||_2$. Se puede adem"as definir la noción de angulo θ entre los vectores x e y a través de la norma y el producto entre ellos:

$$cos(\theta) = \frac{x^T y}{||x||_2 ||y||_2}$$

A.0.1.2. *Cauchy Schwarz.* Sea E un espacio vectorial real y sea $\langle \cdot, \cdot \rangle : E \to \mathbb{R}$ una función positiva, simétrica y bilinear, es decir:

(1) $\langle x, x \rangle \geq 0$
(2) $\langle x, y \rangle = \langle y, x \rangle$
(3) $\langle \alpha x + y, z \rangle = \alpha \langle x, z \rangle + \langle y, z \rangle$

Para todo $x, y, z \in E$, $\alpha \in \mathbb{R}$. Se quiere demostrar que $\langle \cdot, \cdot \rangle$ cumple la desigualdad de Cauchy-Schwarz, es decir:

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \cdot \langle y, y \rangle \ x, y \in E$$

Proof. Sean $x, y \in E$ y $\alpha \in \mathbb{R}$, debido a la positividad de $\langle \cdot, \cdot \rangle$ se cumple:

$$0 \leq \langle \alpha x + y, \alpha x + y \rangle$$

Lo cual, aplicando la bilinearidad y simetría de $\langle \cdot, \cdot \rangle$, se puede reescribir en la forma:

$$0 \leq \alpha^2 \langle x, x \rangle + 2\alpha \langle x, y \rangle + \langle y, y \rangle$$

Si $\langle x, x \rangle = 0$ o $\langle y, y \rangle = 0$, la desigualdad se cumple de manera trivial. Por otra parte, nuevamente debido a la positividad de $\langle \cdot, \cdot \rangle$, si $\langle x, x \rangle > 0$, tomar $\alpha = -\frac{\langle x,y \rangle}{\langle x,x \rangle}$ reduce la expresión anterior a:

$$0 \leq -\frac{|\langle x, y \rangle|^2}{\langle x, x \rangle} + \langle y, y \rangle$$

De la arbitrariedad de $x, y \in E$ se concluye la afirmación buscada. □

A.0.2. Tópicos de Teoría de la Medida.

Definición A.1 (medida)**.** Sea X un conjunto y $\mathcal{C} \subseteq \mathcal{P}(X)$ tal que $\varphi \in \mathcal{C}$. Una función:

$$\mu : \mathcal{C} \to \overline{\mathbb{R}}_+$$

Se dice que es una **medida** sobre $\mathcal{C}$ si satisface:

(1) $\mu(\varphi) = 0$
(2) Si $(A_n)_{n \in \mathbb{N}} \subseteq \mathcal{C}$ es una sucesión de conjuntos disjuntos tales que $\bigcup_{n \in \mathbb{N}} A_n \in \mathcal{C}$, entonces:

$$\mu(\bigcup_{n \in \mathbb{N}} A_n) = \sum_{n \in \mathbb{N}} \mu(A_n)$$

(esta propiedad se llama σ-aditividad).

Definición A.2 (σ-álgebra)**.** $\mathcal{T} \subseteq \mathcal{P}(X)$ se dice que es una σ-álgebra si:

(1) $X \in \mathcal{T}$
(2) $A, B \in \mathcal{T} \Rightarrow A \setminus B \in \mathcal{T}$
(3) $(A_n)_{n \in \mathbb{N}} \subseteq \mathcal{T} \Rightarrow \bigcup_{n \in \mathbb{N}} (A_n) \in \mathcal{T}$

Si tenemos un conjunto X, una σ-álgebra $\mathcal{T} \subseteq \mathcal{P}(X)$, y una medida μ sobre $\mathcal{T}$, lo identificaremos con la tupla:

$$(X, \mathcal{T}, \mu)$$

Definición A.3. (Medida σ-finita) Sea $\mathcal{C} \subseteq \mathcal{P}(X)$ tal que $\phi \in \mathcal{C}$. Una medida $\mu : \mathcal{C} \to \overline{\mathbb{R}}_+$ se dirá:

(1) *σ-finita* si:

$$\exists (A_n)_{n \in \mathbb{N}} \subseteq \mathcal{C} \text{ tal que } X = \bigcup_{n \in \mathbb{N}} A_n \text{ y } \mu(A_n) < +\infty, \ \forall n \in \mathbb{N}$$

(2) *finita* si:

$$\mu(A_n) < +\infty,\ \forall A \in \mathcal{C}$$

A.0.3. Funciones Simples.

Definición A.4. (Función Simple) Sea $(X, \mathcal{T})$ un espacio medible. Una función:

$$f : X \to \mathbb{R}$$

se dirá simple si:

$$\exists (A_i)_{i=1}^n \subseteq \mathcal{T},\ \exists (a_i)_{i=1}^n \subseteq \mathbb{R}\ tal\ que\ f(x) = \sum_{i=1}^n a_i 1_{A_i}(x)$$

Denotaremos por $\xi = \xi(X, \mathcal{T}, \mathbb{R})$ al conjunto de todas las funciones simples a valores reales. Análogamente denotaremos por $\xi_+ = \xi(X, \mathcal{T}, \mathbb{R}_+)$ al conjunto de funciones simples a valores en $\mathbb{R}_+$.
Notemos que si f es una función simple, entonces existe $(A_i)_{i=1}^n$ partición medible de X tal que:

$$f(x) = \sum_{i=1}^n a_i 1_{A_i}(x)$$

A.0.4. Definición de la integral.

Definición A.5. (Integral) Sea $(X, \mathcal{T}, \mu)$ un espacio de medida. Sea $f \in \xi_+$, digamos $f(x) = \sum_{i=1}^n a_i 1_{A_i}(x)$, con $(A_i)_{i=1}^n$ partición medible de X. Se define la **integral** de f como:

$$\int f d\mu := \sum_{i=1}^n a_i \mu(A_i) \in \overline{\mathbb{R}}_+$$

A.0.5. Completitud de un espacio de medida.

Definición A.6. (Conjunto Despreciable) Sea $(X, \mathcal{T}, \mu)$ un espacio de medida. Un conjunto $N \in \mathcal{P}(X)$ se dice que es un **despreciable** si:

$$\exists A \in \mathcal{T}\ tal\ que\ N \subseteq A\ y\ \mu(A) = 0$$

La σ-álgebra $\mathcal{T}$ se dirá completa si contiene a todos los despreciables.

Definición A.7. Si $(X, \mathcal{T}, \mu)$ un espacio de medida y $p : X \to \{V, F\}$ una proposición (V=verdadero, F=falso). Se dirá que la proposición p **se satisface en casi todas partes**(lo cual se abrevia ctp ó μ-ctp) si el siguiente conjunto:

$$\{x \in X / p(x) = F\}$$

es un despreciable para $(X, \mathcal{T}, \mu)$.

A.0.6. Probabilidad.

Sea $(\Omega, \mathcal{B}, \mathbb{P})$ espacio de probabilidades, donde :

- Ω : espacio muestral
- $\mathcal{B}$ sigma álgebra

- $\mathbb{P}$: ley de probabilidades

A continuación se definen algunos conceptos de probabilidad de tal manera de explicar teoremas y proposicones que seutilizarón a lo largo del texto.

Definición A.8. Probabilidad Condicional: $\mathbb{P}(X|Y) = \frac{\mathbb{P}(X,Y)}{\mathbb{P}(Y)}$

Definición A.9. independencia: $X \perp Y \iff \mathbb{P}(X,Y) = \mathbb{P}(X)\mathbb{P}(Y)$

Definición A.10. Independencia Condicional: $X \perp Y|E \iff \mathbb{P}(X,Y|E) = \mathbb{P}(X|E)\mathbb{P}(Y|E)$

A partir de esto, se pueden formular los siguientes teoremas:

Proposición 4. Teorema de Bayes: Se define como $\mathbb{P}(X|E) = \frac{\mathbb{P}(E|X)\mathbb{P}(X)}{\mathbb{P}(E)}$, donde $\mathbb{P}(X|E)$ es la distribucion a posteriori de X dado E. $\mathbb{P}(E|X)$ es la verosimilitud de los datos. $\mathbb{P}(X)$ es la probabilidad a priori de X y $\mathbb{P}(E)$ es la probabilidad de los datos.

Proposición 5. Regla de la Cadena: Sean $X_1, \ldots, X_N$ eventos medibles, entonces la probabilidad conjunta puede expresarse como:
$\mathbb{P}(X_1, \ldots, X_N) = \mathbb{P}(X_1) \prod_{i=2}^{N} \mathbb{P}(X_i|X_1, \ldots, X_{i-1})$

Proposición 6. Regla de la Marginación: Sea $\{E_i\}_{i\in\mathbb{N}}$ una partición de Ω , tales que $\bigcup_{i\in\mathbb{N}} E_i = \Omega$ entonces:

$$\mathbb{P}(X) = \sum_{i\in\mathbb{N}} \mathbb{P}(X|E_i)\mathbb{P}(E_i)$$

Proposición 7. Dos eventos X e Y son independientes si y solo si la distribución a priori y a posteriori son iguales, es decir, $X \perp Y \iff \mathbb{P}(X|Y) = \mathbb{P}(X)$

Proposición 8. Dos eventos X e Y son independientes condicionados al evento E si y solo si, la distribución a posteriori de X dado Y, E es igual a la distribución a posteriori de X dado solo E, es decir $X \perp Y|E \iff \mathbb{P}(X|Y,E) = \mathbb{P}(X|E)$

Definición A.11. Una variable aleatoria x es una función de Ω a valores en $\mathbb{R}$, es decir, $x(\omega) \in \mathbb{R} \operatorname{con} \omega \in \Omega$

Definición A.12. Dada una variable aleatoria x, se define su esperanza y varianza de la siguiente manera:

$$\mu_x = \mathbb{E}[x] = \int_\Omega x(\omega) d\mathbb{P}(\omega)$$
$$\sigma_x^2 = \mathbb{V}[x] = \int_\Omega (x(\omega) - \mu_x)^2 \, d\mathbb{P}(\omega)$$

Definición A.13. Dada una variable aleatoria x, se dice que x sigue una distribución de probabilidades $F(c)$ si se cumple que $\mathbb{P}(\omega : x(\omega) \leq c) = F(c)$ .Se dice que x tiene función de densidad $p(x)$ si se cumple que:

$$F(c) = \int_{-\infty}^{c} p(x)dx = \int_{\omega:x(\omega)\leq c} d\mathbb{P}(\omega)$$

Se conoce como el soporte de la distribución F a la imagen de la variable aleatoria $im(x) = x(\Omega) \subseteq \mathbb{R}$.
Si el soporte de F es numerable entonces se dirá que la variable aleatoria sigue una

distribucion discreta, en caso contrario, se dirá que es una distribución continua. Para el primer caso, se utilizan sumatorias:

$$F(c) = \sum_{i \leq c} p_i$$
$$p(x) = \sum_{i \leq c} p_i \delta_i(x)$$

Donde δ_i corresponde a la función delta dirac.

Proposición 9. Dada una variable aleatoria x , su esperanza y varianza son:

$$\mu_x = \mathbb{E}[x] = \int_{\mathbb{R}} xp(x)dx$$
$$\sigma_x^2 = \mathbb{V}[x] = \int_{\mathbb{R}} (x - \mu_x)^2 p(x)dx$$

A.0.7. Tópicos de Optimización.

A.0.7.1. *Problema de Optimización con Restricciones, Lagrangiano.* Sea el problema de optimización con restricciones:

$$\begin{aligned} min\ & f(x)\ x \in \mathbb{R}^n \\ s.a\ & c_i(x) = 0,\ i \in \mathcal{C}_E \\ & c_i(x) \geq 0,\ i \in \mathcal{C}_I \end{aligned}$$

donde $f(x)$ es la función objetivo, c_i con $i = 1, 2, ..., p$ son las funciones de restricción, $\mathcal{C}_E$ es el conjunto de indices de las restricciones de igualdad en el problema y $\mathcal{C}_I$ es el conjunto de restricciones de desigualdad. Cualquier punto que satisface todas las restricciones es llamado un punto factible y el conjunto de todos aquellos puntos es referido como la región factible. Se definen las restricciones activas en un punto x' por el conjunto de indices $\mathcal{A}(x') = \{i : c_i(x') = 0\}$ tales que cualquier restricción es activa en x' si x' esta en el borde o en la región factible. En este contexto se define el Lagrangiano por:

$$\mathcal{L}(x, \lambda) = f(x) - \sum_i \lambda_i c_i(x)$$

A.0.7.2. *Condiciones KKT.* En un problema de optimización con restricciones se tiene como condición necesaria de primer orden lo siguiente:

Si x^* es un mínimo local del problema de optimización con restricciones y se mantiene cierta regularidad alrededor de x^*, entonces existen multiplicadores de Lagrange λ^* tales que x^*, λ^* satisfacen el siguiente sistema:

$$\begin{aligned}\nabla_x \mathcal{L}(x,\lambda) &= 0\\ c_i(x) &= 0 \ i \in \mathcal{C}_E\\ c_i(x) &\geq 0 \ i \in \mathcal{C}_I\\ \lambda_i &\geq 0 \ i \in \mathcal{C}_I\\ \lambda_i c_i(x) &= 0 \ \forall \ i\end{aligned}$$

El cual consiste en las llamadas condiciones de Karush-Kuhn-Tucker (KKT). Un punto x^* que satisface las condiciones se refiere como punto KKT. La última conidición $\lambda_i c_i(x) = 0$ es llamada *condición de complementariedad* y afirma que λ_i y c_i no pueden ser ambos distintos de cero, o equivalentemente que las restricciones inactivas tienen un multiplicador igual a cero. Si no existe i tal que $\lambda_i^* = c_i^*(x) = 0$ entonces se mantiene una complementariedad estricta. En cambio si $\lambda_i^* = 0$ se denomina fuertemente activa , $c_i^* > 0$ débilmente activa $\lambda_i^* = c_i^* = 0$ e inactiva si $\lambda_i^* = 0$, $c_i^* > 0$.

Las Condiciones de segundo orden pueden ser expresadas en términos de la matriz Hessiana con respecto a x del Lagrangiano $\nabla_x^2 \mathcal{L}(x^*, \lambda^*) = \nabla^2 f(x^*) - \sum_i \lambda_i^* \nabla^2 c_i(x^*)$.

A.0.7.3. *Problema de Programación Cuadrática.* Un problema de programación cuadrática es de la forma:

$$\begin{aligned}Minimizar \ q(x) &= \frac{1}{2} x^T G x + g^T x\\ sujeto \ a \ a_i^T x &= b_i, \ i \in \mathcal{C}_E\\ a_i^T x &\geq b_i, \ i \in \mathcal{C}_I\end{aligned}$$

Si la matriz Hessiana G es semidefinida positiva, la solucion x^* es global.Si G es definida positiva la solución x^* es global y única. Cuando la Hessiana es indefinida puede existir otra solución local que la solución global.

Bibliography

[1] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

[2] Bishop, C. M. (2006). Pattern Recognition. Machine Learning.

[3] Hastie, T., R. Tibshirani, and J. Fried- man (2001). The Elements of Statis- tical Learning. Springer.

[4] Schoelkopf, B. and A. Smola (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

[5] Koller, D. and N. Friedman (2009). Probabilistic Graphical Models: Principles and Techniques. MIT Press.

[6] Aronszajn, N. (1950). Theory of reproducing kernels. Transactions of the American mathematical society, 68(3), 337-404.

[7] Schoelkopf, B., A. Smola, and K.-R.Mueller (1998). Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299 – 1319.

[8] Jaakkola, T. and D. Haussler (1998). Exploiting generative models in discriminative classifiers. In NIPS, pp.487–493

[9] Krishnapuram, B., L. Carin, M. Figueiredo, and A. Hartemink (2005). Learning sparse bayesian classifiers: multi-class formulation, fast algorithms, and generalization bounds. IEEE Transaction on Pattern Analysis and Machine Intelligence.

[10] Tipping, M. (2001). Sparse bayesian learning and the relevance vector machine. J. of Machine Learning Research 1, 211–244.

[11] Hastie, T., R. Tibshirani, and J. Friedman (2009). The Elements of Statistical Learning. Springer. 2nd edition

[12] Hofmann, T., Scholkopf, B., Smola, A. J. (2008). Kernel methods in machine learning. The annals of statistics, 1171-1220

[13] Suykens, J. A., Van Gestel, T., De Brabanter, J. (2002). "Least squares support vector machines". World Scientific.

[14] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[15] Steinwart, I.,Christmann, A. (2008). Support vector machines. Springer Science and Business Media.

[16] Gelman, A., and D. B. Rubin. 1992. Inference from Iterative Simulation Using Multiple Sequences. Statistical Science 7: 457–511.

[17] Brooks, S. P., and A. Gelman. 1997. General Methods for Monitoring Convergence of Iterative Simulations. Journal of Computational and Graphical Statistics 7: 434–455.

[18] Duane, S., A. Kennedy, B. Pendleton, and D. Roweth (1987). Hybrid Monte Carlo. Physics Letters B 195(2), 216–222

[19] Jebara, T., R. Kondor, and A. Howard (2004). Probability product kernels. J. of Machine Learning Research 5, 819–844

[20] David M. Blei, Alp Kucukelbir and Jon D. McAuliffe (2017) Variational Inference: A Review for Statisticians, Journal of the American Statistical Association, 112:518, 859-877