Inteligencia Artificial

Víctor Mijangos de la Cruz

IV. Modelos probabilísticos en IA

Probabilidad en sistemas inteligentes

Incertidumbre

La **incertidumbre** se presenta continuamente en los problemas de IA, donde no hay una forma precisa de decidir.

En particular, el entorno puede determinar situaciones de incertidumbre cuando:

- Es parcialmente observable, no se tiene información completa del problema.
- No hay determinismo, es decir son procesos estocásticos.
- Existe participación de otros agentes, como adversarios, que indeterminada las circunstancias.

Modelos de incertidumbre

Los modelos basados en búsqueda y, en general, los modelos lógicos, asumen que existen **predicciones certeras** en los problemas.

La **incertidumbre** se presenta cuando no existe una solución certera de un problema, sino que se dan varias posibles soluciones.

De las primeras aproximaciones a la incertidumbre en los sistemas inteligentes son:

1. Logicista: Utiliza técnicas de lógica no-monotónica para lidiar con la incertidumbre.
2. Calculista: Utiliza métodos numéricos para aproximar soluciones a partir, por ejemplo, de lógica difusa.
3. Probabilística: Utiliza técnicas de la teoría de la probabilidad para lidiar con la incertidumbre.

Límites de la lógica

En el **razonamiento lógico**, las consecuencias de acciones se determinan como reglas formales de la forma 'Si X, entonces Y'.

Esto implica un **conocimiento de las causas y de los efectos** dentro del entrono en que trabajamos.

Si embargo, las reglas lógicas tienen limitaciones, que, principalmente, se presentan por:

- La enumeración de un sistema exhaustivo de reglas es **demasiado trabajo**, y se vuelve difícil trabajar con esas reglas (por ejemplo en el lenguaje natural).
- Hay una **ignorancia teórica**, no existe una teoría completa del dominio (por ejemplo, procesamiento de imágenes).
- A pesar de conocer una teoría, no conocemos los **casos prácticos concretos** que puedan presentarse (por ejemplo, diagnostico médico).

Similitudes y diferencias

Podemos hablar de dos paradigmas en que coinciden y se diferencian los tipos de agentes.

- **Compromisos ontológicos**: Tanto la lógica como la probabilidad que el mundo está compuesto de situaciones que pueden o no suceder dado ciertos casos particulares.
- **Compromisos epistemológicos**: Difieren en tanto que:
  - Agente lógico asume que toda proposición o es falsa o es verdadera.
  - Un agente probabilístico tiene un grado (numérico) de **creencia**.

Una **creencia** es una concepción que el agente tiene del mundo, pero que puede cambiar en el futuro bajo ciertas circunstancias.

Probabilidad

La teoría de probabilidad se basa en los **espacios probabilísticos**, definidos a partir de los elementos $(\Omega, \mathcal{F}, P)$, donde Ω es un espacio muestral, $\mathcal{F}$ un conjunto de eventos de Ω y P una medida de probabilidad.

La **medida de probabilidad** es una función $P : \mathcal{F} \to [0, 1]$ que cumple las siguientes propiedades:

1. $P(E) \geq 0$, para todo $E \in \mathcal{F}$
2. $P(\Omega) = 1$
3. Si $E_0, E_1, ... \subseteq \mathcal{F}$, y $E_i \cap E_j \neq \emptyset$, entonces:

$$P\Big(\bigcup_{i=0}^{\infty} E_i\Big) = \sum_{i=0}^{\infty} P(E_i)$$

Probabilidad conjunta

Una de las propiedades del conjunto de eventos es que dado dos eventos $E_1, E_2 \in \mathcal{F}$, se cumple necesariamente que $E_1 \cap E_2 \in \mathcal{F}$.

Por tanto la probabilidad:

$$P(E_1, E_2) := P(E_1 \cap E_2)$$

está bien determinada y se conoce como **probabilidad conjunta**.

Si E_1 y E_2 se definen por proposiciones lógicas a y b, entonces la probabilidad conjunta puede verse como la probabilidad de la operación lógica de conjunción:

$$P(a \wedge b)$$

Probabilidad condicional

Una operación importante que traslada la idea de **causalidad** dentro del lenguaje de la probabilidad es la probabilidad condicional.

Probabilidad condicional

Dados dos eventos E_1 y E_2, la probabilidad condicional del evento E_1 dado que ha sucedido el evento E_2 se define como:

$$P(E_1|E_2) = \frac{P(E_1, E_2)}{P(E_2)}$$

La probabilidad condicional puede interpretarse como la probabilidad de que un evento se siga dado que ha sucedido un evento previo.

A partir de esta definición, podemos ver que la **probabilidad conjunta** puede reescribirse como:

$$P(E_1, E_2) = P(E_1|E_2)P(E_2)$$

Variables aleatorias

Una forma común de expresar los eventos en términos numéricos son las **variables aleatorias**; estas pueden verse como mapeos que toman elementos del espacio muestral y las llevan hacia un número real.

De esta forma, la probabilidad de un evento asociado a un número real x por una variable X se puede expresar como:

$$p(X = x)$$

Y podemos expresar la **función de distribución** como:

$$p(X \leq x) = \sum_{t=-\infty}^{x} p(X = t)$$

Agentes probabilísticos

Un agente probabilístico debe tomar decisiones en base a **preferencias** entre posibilidades, representadas por eventos.

Creencia

Una creencia (belief) puede interpretarse como información que un agente tiene sobre el mundo, la cual puede modificarse a partir de nuevo conocimiento.

Estado de creencias

Un estado de creencias refiere a la representación del conjunto de todos los posibles estados del mundo que considera un agente.

Utilidad

Las preferencias que un agente tenga sobre las creencias que cuenta pueden depender de la **utilidad** de estas.

Teoría de utilidad

En la teoría de utilidad, cada evento tiene un grado de utilidad, de tal forma que un agente preferirá aquellos estados que tengan más utilidad para éste.

El grado de utilidad dependerá del agente.

También podemos interpretar la teoría de la utilidad como el hecho de **minimizar el riesgo** que ciertos estados representan.

Teoría de la decisión

Si utilizamos la **teoría de la probabilidad** para hacer decisiones sobre la utilidad, conformamos lo que se conoce como **teoría de la decisión**:

$$\text{Teoría de la Decisión} = \text{Teroía de probabilidad} + \text{Teoría de utilidad}$$

Idea fundamental de la teoría de la decisión

Un agente es racional si y sólo si escoge las acciones que lo llevan hacia la utilidad esperada más alta, promediada sobre todas los posibles resultados de la acción.

Es decir, buscamos el principio de la **Máxima Utilidad Esperada (MUE)**.

Agente de decisión

Algorithm Agente de decisión teórico

1: **procedure** DT-AGENT(percept)
2: **estático**: belief_state, action
3: belief_state ← UPDATE(action, percept)
4: Calcular las probabilidades de las consecuencias para las acciones dadas las acciones y el estado de creencias: $p(a|action, belief\_state)$
5: action ← $\arg\max MUE(a)$
6: **return** action
7: **end procedure**

Representación probabilística

Las **proposiciones lógicas** pueden expresarse como **expresiones probabilísticas**.

Lógica	Probabilística
$a \wedge b$	$p(a, b)$
NOT a	$1 - p(a)$
$a \vee b$	$p(a) + p(b) - p(a, b)$

Los **mundos posibles** son aquellas circunstancias que pueden presentarse ante la acción de un agente. Esto mundos se asocian a **variables aleatorias** a partir de las cuales se asignan probabilidades a las posibilidades.

Modelos gráficos

Distribución conjunta total

Una forma de **estimar** probabilidades es a partir de una **distribución conjunta total**, que es la tabla de distribuciones conjuntas de las variables aleatorias en juego.

Dado dos variables aleatorias X e Y, su distribución conjunta total está dada por los valores:

$$p(X = x, Y = y)$$

A partir de esta distribución se puede obtener:

- **Distribución marginal**: $p(X = x) = \sum_y p(X = x, Y = y)$.
- **Distribución condicional**: $p(Y = y|X = x) = \frac{p(X=x,Y=y)}{\sum_y p(X=x,Y=y)}$

Ejemplo: Distribución conjunta

Considérese la siguiente tabla que contienen las probabilidades conjuntas de tres variables:

	Dolor cabeza		NOT Dolor cabeza	
	Dolor garganta	NOT Dolor garganta	Dolor garganta	NOT Dolor garganta
Gripe	0.108	0.012	0.072	0.008
NOT Gripe	0.016	0.064	0.144	0.576

Podemos calcular la probabilidad:

$$p(X = gripe|Y = dolor \cdot cabeza) \propto \big(p(X = gripe, Y = dolor \cdot cabeza, Z = dolor \cdot garganta) + p(X = gripe, Y = dolor \cdot cabeza, Z = \text{NOT } dolor \cdot garganta)\big)$$

En este caso, Z es una variable no observada u oculta.

Estimación por distribución conjunta

Del ejemplo anterior, podemos ver que para obtener las probabilidades condicionales entre varias variables se tiene:

$$p(X = x|Y = y) = \frac{\sum_z p(X = x, Y = y, Z = z)}{\sum_x \sum_z p(X = x, Y = y, Z = z)}$$

Las probabilidades $p(X = x, Y = y, Z = z)$ representan la distribución conjunta total.

El problema que surge es que si el número de variables crece, el total de cálculos que se debe hacer crece.

Este tipo de modelos requieren $O(2^n)$ probabilidades, donde n es el número de variables booleanas. En variables no booleanas tenemos $O(m^n)$.

Independencia de probabilidades

Variables independientes

Dos variables aleatorias X e Y se dice que son independientes si cumplen:

$$p(X = x, Y = y) = p(X = x)p(Y = y)$$

La **independencia condicional** puede establecerse como:

$$p(X = x, Y = y|Z = z) = p(X = x|Z = z)p(Y = y|Z = z)$$

La independencia de variables permite **factorizar** el cálculo de probabilidades dentro de un conjunto pequeño de probabilidades:

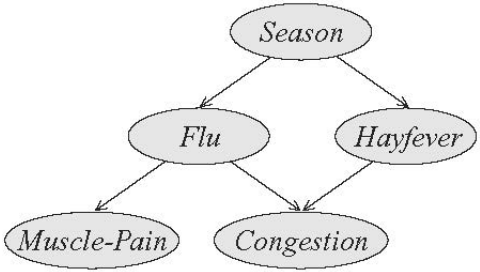
$$p(x_1, ...x_n|z) = \prod_{i=1}^{n} p(x_i|z)$$

Representación gráfica

Una forma de representar las dependencias conjuntas y condicionales de las probabilidades es a través de gráficas.

- Los nodos representan las variables aleatorias.
- Las aristas dirigidas representan las variables condicionales $x \rightarrow y$ implica $p(y|x)$.
- La probabilidad de la gráfica es:

$$p(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} p(x_i|parent(x_i))$$



Modelo gráfico

Modelo gráfico

Un modelo gráfico es un modelo probabilístico que, dada una gráfica $G = (V, E)$, indexa las variables $X_{v \in V}$ con los nodos de la gráfica.

Ventajas de los modelos gráficos son:

- Permiten una **visualización** sencilla del modelo probabilístico.
- La gráfica muestra cuando las variables son independientes.
- Permiten visualizar los cómputos necesarios para hacer estimación estadística.

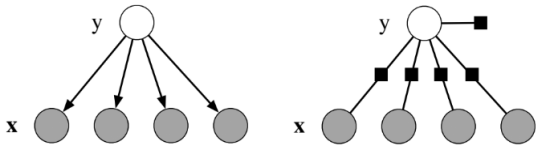
Distinciones de modelos gráficos

1. **Modelos gráficos dirigidos**: Describen una factorización basada en distribuciones condicionales locales:

$$p(\mathbf{x}) = \prod_v p(x_v | x_{\pi(v)})$$

2. **Modelos gráficos no-dirigido**: La distribución de probabilidad se factoriza de acuerdo a una colección de factores:

$$p(\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_v \psi_v(x_v)$$



Clasificación con modelos gráficos

La clasificación consiste en **predecir** una clase a partir de una descripción en rasgos de un objeto.

Sean $X_1, X_2, ..., X_d$ variables aleatorias que describen los rasgos en un vector de rasgos d-dimensional:

$$x = \begin{pmatrix} X_1 = x_1 & X_2 = x_2 & \cdots & X_d = x_d \end{pmatrix}$$

Un **problema de clasificación** buscará asignar una clase $\hat{y}$ determinada como:

$$\hat{y} = \arg\max_y p(Y = y|x)$$

Los modelos gráficos pueden usarse para los problemas de clasificación.

Modelos bayesianos

Teorema de Bayes

Teorema de Bayes

Sean X e Y dos variables aleatorias. La probabilidad $p(Y|X)$ se puede determinar como:

$$p(Y|X) = \frac{p(X|Y)p(Y)}{\sum_y p(X|Y=y)p(Y=y)}$$

Sabemos que la probabilidad condicional para cualquiera dos eventos x e y está dada por: $p(x|y) = \frac{p(x,y)}{p(y)}$ de donde se obtiene que $p(x,y) = p(x|y)p(y)$, además de la estimación de probabilidades marginales tenemos que $p(x) = \sum_y p(x,y)$, de donde:

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{\sum_y p(x,y)}$$
$$= \frac{p(x|y)p(y)}{\sum_{y'} p(x|y')p(y')}$$

Independencia y factorización

Sin x es un vector d-dimensional con d variables de entrada que son **independientes**, podemos ver que:

$$\begin{aligned} p(x|y) &= p(x_1, x_2, ..., x_d|y) \\ &= \prod_{i=1}^{d} p(x_i|y) \end{aligned}$$

Ignorando la función de partición, tenemos que:

$$\begin{aligned} p(y|x_1, x_2, ..., x_d) &\propto p(x_1, x_2, ..., x_d, y) \\ &= \prod_{i=1}^{d} p(x_i|y)p(y) \end{aligned}$$

Modelo de Bayes ingenuo

Modelo de Bayes ingenuo

El modelo de Bayes ingenuo (o Bayes naîve) es un modelo gráfico dirigido (y generativo) para clasificación que determina una clase como:

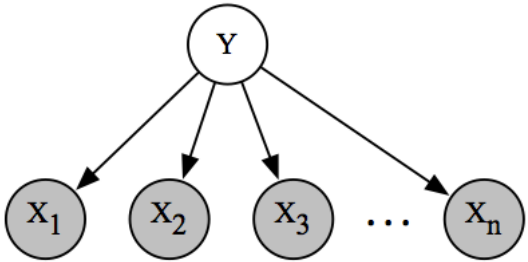
$$\arg\max_y p(x_1, x_2, ..., x_d, y) = \arg\max_y \prod_{i=1}^{d} p(x_i|y)p(y)$$

Para determinar la clasificación, estimamos una **probabilidad conjunta** de los datos de entrada con las clases. Se puede notar que:

$$\arg\max_y p(y|x_1, x_2, ..., x_d) = \arg\max_y p(x_1, x_2, ..., x_d, y)$$

Modelo de Bayes ingenuo

- El modelo de Bayes ingenueo es un **modelo gráfico dirigido**, representa las probabilidades conjuntas como una gráfica.
- Es **generativo**: asume que la clase Y genera a los datos visibles $X_1, ..., X_n$.
- En este sentido, asume que la clase Y es la **causa** y los observables $X_1, ..., X_n$ los efectos.



Ejemplo: Bayes ingenuo

Supongamos que contamos con los siguientes datos de **pacientes con gripe**:

- **Diagnóstico**: gripe; **Síntomas**: dolor de cabeza, dolor de garganta, fiebre, dolor muscular.
- **Diagnóstico**: no gripe; **Síntomas**: dolor de cabeza, fiebre, dolor estomacal.
- **Diagnóstico**: gripe; **Síntomas**: dolor de cabeza, fiebre, escurrimiento nasal.
- **Diagnóstico**: no gripe; **Síntomas**: escurrimiento nasal, dolor muscular.
- **Diagnóstico**: gripe; **Síntomas**: fiebre, escurrimiento nasal, dolor estomacal.

Nuestro modelo toma una variable clase Y binaria que predice si es gripe o no lo es.

Nuestros datos observables son 6 variables binarias X_i, $i = 1, ..., 6$, correspondientes a dolor de cabeza, dolor de garganta, fiebre, dolor muscular, dolor estomacal, escurrimiento nasal.

Ejemplo: Bayes ingenuo

Tenemos los priors: $p(gripe) = \frac{3}{5}$ y $p(no\_gripe) = \frac{2}{5}$

$\mathbf{X_i}$	gripe	no gripe
dolor de cabeza	$\frac{2}{10}$	$\frac{1}{6}$
dolor de garganta	$\frac{1}{10}$	0
fiebre	$\frac{3}{10}$	$\frac{1}{6}$
dolor muscular	$\frac{1}{10}$	$\frac{1}{6}$
dolor estomacal	$\frac{1}{10}$	$\frac{2}{6}$
escurrimiento nasal	$\frac{2}{10}$	$\frac{1}{6}$

Si recibimos los **síntomas**: dolor de cabeza, fiebre, escurrimiento nasal, dolor muscular:

$$p(gripe, dolor\_cabeza, fiebre, escurrimiento, dolor\_muscular) = \frac{2}{10}\frac{3}{10}\frac{2}{10}\frac{1}{10}\frac{3}{5} = \frac{36}{50000} \approx 0.00072$$

$$p(no\_gripe, dolor\_cabeza, fiebre, escurrimiento, dolor\_muscular) = \frac{1}{6}\frac{1}{6}\frac{1}{6}\frac{1}{6}\frac{2}{5} = \frac{2}{6480} \approx 0.0003$$

Redes bayesianas

Red bayesiana

Una red bayesiana es un modelo gráfico dirigido que representa las condiciones de un conjunto de variables aleatorias y estiman la probabilidad de la red en base a la distribución condicional de dichas variables.

Las redes bayesianas son útiles para calculo de distribuciones complejas, así como para la visualización de estos eventos.

También se pueden utilizar en aplicaciones de **clasificación**, como en el caso del modelo de Bayes ingenuo.

Elementos de una red bayesiana

Una red bayesiana es una gráfica dirigida $G = (V, E)$ tal que cumple:

- Cada nodo corresponde a una variable aleatoria (discreta o continua) $V \equiv \{X_i\}_{i=1}^{n}$.
- Las aristas dirigidas representan las relaciones condicionales entre variables, i.e., $E \equiv \{X_i \rightarrow X_j : i \neq j\}$.
- Si $X_i \perp X_j$ entonces no hay conexión entre las aristas.
- Cada nodo tiene información probabilística asociada $\theta(X_i|\pi(X_i))$ (π relación de parentesco) que depende de un número finito de parámetros.

Red bayesiana y distribución conjunta

Una red bayesiana define cada entrada de la probabilidad conjunta en base a la información probabilística:

$$p(x_1, ..., x_n) = \prod_{i=1}^{n} \theta(x_i|\pi(x_i))$$

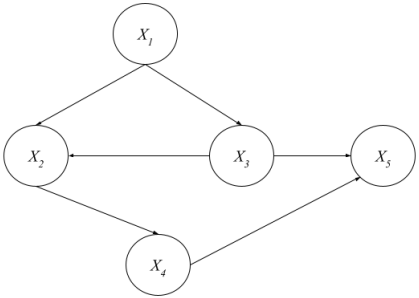
De esta forma, la red bayesiana puede **responder a cualquier consulta probabilística** al sumar sobre los valores correspondientes.

Por ejemplo, si queremos responder a la consulta $p(x_n)$ tenemos:

$$\begin{aligned} p(x_n) &= \sum_{x_1} \cdots \sum_{x_{n-1}} p(x_1, ..., x_n) \\ &= \sum_{x_1} \cdots \sum_{x_{n-1}} \prod_{i=1}^{n} \theta(x_i|\pi(x_i)) \end{aligned}$$

Representación gráfica

Podemos construir una gráfica con las variables X_1, X_2, X_3, X_4 y X_5. Las relaciones condicionales se manifiestan por las flechas:



Dada la gráfica tenemos las siguiente información para cada nodo:

- $\theta(X_1)$
- $\theta(X_2|X_1)$
- $\theta(X_3|X_1)$
- $\theta(X_4|X_2)$
- $\theta(X_5|X_3, X_4)$

Por lo que una consulta conjunta debe estimarse como:

$$p(x_1, x_2, x_3, x_4, x_5) = \theta(x_1)\theta(x_2|x_1)\theta(x_3|x_1)\theta(x_4|x_2)\theta(x_5|x_3, x_4)$$

Probabilidades condicionales

Se pude ver que la información de los nodos corresponde a dependencias condicionales, por lo que:

$$\theta(X_i|\pi(X_i)) = p(X_i|\pi(X_i))$$

Por lo que el cálculo de la probabilidad condicional se basa en la **regla de la cadena** y las dependencias condicionales:

$$\begin{aligned} p(x_1, ..., x_n) &= \prod_{i=1}^{n} p(x_i|x_{i-1}, ..., x_1) \\ &= \prod_{i=1}^{n} p(x_i|\pi(x_i)) \end{aligned}$$

Red bayesiana

Propiedad de no-descendientes

Se dice que un conjunto de variables tiene la propiedad de no-descendientes si, dado sus padres, estas variables son independientes de las variables que no son sus descendientes (en la red bayesiana).

Una red bayesiana representa adecuadamente la distribución conjunta si las variables cumplen la propiedad de no-descendientes.

Cobertura de Markov

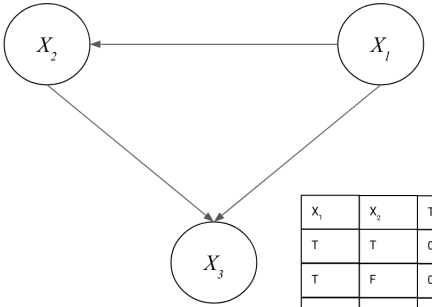
La cobertura de Markov de una variable aleatoria X_i son las otras variables que aportan información para la estimación de la distribución de esta variable.

En una red bayesiana, la cobertura de Markov son los padres, los hijos, y los padres de los hijos.

Ejemplo: Red Bayesiana

Podemos construir una red bayesiana con tres variables X_1, X_2 y X_3 binarias; indicamos las probabilidades condicionales:

X_1	T	F
T	0.5	0.5
F	0.3	0.7



T	F
0.2	0.8

X_1	X_2	T	F
T	T	0.2	0.8
T	F	0.3	0.7
F	T	0.4	0.6
F	F	0	1

Ejemplo: Red Bayesiana

Calculamos las probabilidades conjuntas:

$$\begin{aligned} p(X_1 = T, X_2 = T, X_3 = F) &= p(X_1 = T)p(X_2 = T|X_1 = T)p(X_3 = F|X_2 = T, X_1 = T) \\ &= 0.2 \cdot 0.5 \cdot 0.8 = 0.08 \end{aligned}$$

Y también podemos calcular los marginales:

$$\begin{aligned} p(X_3 = T) &= \sum_i \sum_j p(X_3 = T, X_2 = j, X_1 = i) \\ &= p(X_3 = T|X_2 = T, X_1 = T)p(X_2 = T|X_1 = T)p(X_1 = T) + \\ &\quad p(X_3 = T|X_2 = F, X_1 = T)p(X_2 = F|X_1 = T)p(X_1 = T) + \\ &\quad p(X_3 = T|X_2 = T, X_1 = F)p(X_2 = T|X_1 = F)p(X_1 = F) + \\ &\quad p(X_3 = T|X_2 = F, X_1 = F)p(X_2 = F|X_1 = F)p(X_1 = F) \\ &= 0.2 \cdot 0.5 \cdot 0.2 + 0.3 \cdot 0.5 \cdot 0.2 + 0.4 \cdot 0.3 \cdot 0.8 + 0 \cdot 0.7 \cdot 0.8 = 0.146 \end{aligned}$$

Predicción de cadenas

Supongamos que tenemos variables desconocidas $Y_1 \cdots Y_n$ y observamos una cadena de eventos $X_1 = x_1 \cdots X_n = x_n$. El problema es predecir los valores $y_1 \cdots y_n$ que mejor describan $x_1 \cdots x_n$.

Para esto, debemos observar que en lugar de calcular una probabilidad condicional, podemos obtener una probabilidad conjunta:

$$p(y_1, y_2, ..., y_n | x_1, x_2, ..., x_n) \propto p(y_1, y_2, ..., y_n, x_1, x_2, ..., x_n)$$

Podemos utilizar los conceptos de redes bayesianas para estimar esta probabilidad conjunta.

Independencia condicional

Asumiremos dos cosas sobre las variables de nuestro problema:

- $X_i \perp X_j$ para toda $i \neq j$
- Y_i depende condicionalmente únicamente de Y_{i-1} para toda i.

Bajo estos supuestos, tenemos la siguiente factorización:

$$\begin{aligned} p(y_1, y_2, ..., y_n, x_1, x_2, ..., x_n) &= p(x_1, ..., x_n | y_1, ..., y_n) p(y_1, ..., y_n) \\ &= \prod_{i=1}^{n} p(x_i|y_i) \prod_{i=2}^{n} p(y_i|y_{i-1}) p(y_1) \\ &= \prod_{i=1}^{n} p(x_i|y_i) p(y_i|y_{i-1}) \end{aligned}$$

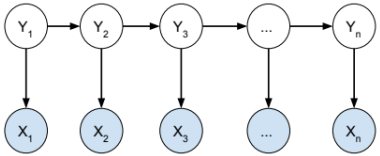
Estimación de probabilidad conjunta

Probabilidad conjunta de cadenas

Un modelo gráfico dirigido y generativo estima la probabilidad de una cadena $y_1, ..., y_n$ a partir de una cadena de entrada $x_1, ..., x_n$ (posibles vectores), a partir de maximizar la función objetivo:

$$p(y_1, y_2, ..., y_n, x_1, x_2, ..., x_n) = \prod_{i=1}^{n} p(x_i|y_i)p(y_i|y_{i-1})$$

En particular, usaremos este método de estimación de probabilidad conjunta para estimar las probabilidades de cadenas $Y_1 \cdots Y_n$.



Modelo de observaciones y emisiones

En general, a cada variable Y_i y cada variable X_j le corresponde una tabla de probabilidades condicionales de la forma $p(Y_i|Y_{i-1})$ y $p(X_j|Y_j)$.

Esto puede hacer el cálculo bastante **exhaustivo**. Para lidiar con eso se asume:

- Toda Y_i tiene la misma distribución, determina las **emisiones**.
- Toda X_j tiene la misma distribución que las otras X_i, determina las **observaciones**.

De esta forma, se requieren sólo asignar una tabla de probabilidades que se comparte entre las diferentes **transiciones en la cadena**.

Observaciones y emisiones

Para realizar el **modelo secuencial** de predicción de etiquetas contamos con dos lenguajes:

- **Símbolos de emisión** $S = \{s_1, ..., s_N\}$, los valores que pueden tomar la variable Y.
- **Observaciones** de entrada $\Sigma = \{w_1, ..., w_M\}$, los valores de X.

Asociamos las variables de salida Y con los símbolos de emisión S y las variables de entrada X con las observaciones.

Nuestro objetivo es determinar una función $\phi : \Sigma^n \to S^n$, de tal forma que una cadena de entrada se transforem en emisiones:

$$w^{(1)}...w^{(n)} \mapsto s^{(1)}...s^{(n)}$$

Factorización de probabilidades

Para estimar el modelo en base a la factorización propuesta, debemos calcular 3 tablas de probabilidades:

Probabilidades iniciales: Probabilidad de que las emisiones inicien en un símbolo s, $p(s)$ para todo $s \in S$.

Probabilidades de transición: La probabilidad de una emisión dada la emisión anterior: $p(s^{(t)}|s^{(t-1)})$.

Probabilidad de observaciones: La probabilidad de que una observación haya sido generada por una emisión en un estado t: $p(w^{(t)}|s^{(t)})$.

Podemos almacenar esta información en: un **vector de probabilidades iniciales**, una **matriz de transición** y una **matriz de probabilidades de observaciones**.

Modelos ocultos de Markov

Modelo oculto de Markov

Un modelo oculto de Markov (HMM) se define como un modelo gráfico generativo y dirigido $HMM = (S, \Sigma, A, B, \Pi)$, donde S son emisiones, Σ observaciones, $A = (a_{i,j}) = p(s_j|s_i)$ es matriz de transiciones, $B = (b_{i,j}) = p(w_j|s_i)$ es matriz de observaciones y $\Pi = (\pi_i) = p(s_i)$ es vector de iniciales.

La cadena de emisiones se predice a partir de maximizar la función:

$$\arg\max_{s_1,...,s_n} p(s_1, s_2, ..., s_n, w_1, w_2, ..., w_n) = \arg\max_{s_1,...,s_n} \prod_{i=1}^{n} p(w_i|s_i)p(s_i|s_{i-1})$$

Entrenamiento y evaluación de HMMs

Los HMMs son algoritmos de aprendizaje de máquina y como tales se componen de dos fases:

Fase de entrenamiento: Se cuenta con un conjunto de entrenamiento de la forma:

$$\mathcal{S} = \{(w^{(1)}...w^{(n)}, s^{(1)}...s^{(n)}) : w^{(1)}...w^{(n)} \in \Sigma^*, s^{(1)}...s^{(n)} \in S^*\}$$

A partir de este data set se estima el modelo oculto de Markov $HMM = (S, O, A, B, \Pi)$.
Los iniciales y la matriz de transición se estiman a partir de este conjunto de entrenamiento.

Fase de evaluación: Se predicen la cadena de emisiones óptima para cadenas de observaciones de un dataset de evaluación. Para esto se suele usar el **algoritmo de Viterbi**. Se evalúa el modelo.

Entrenamiento de HMMs

Para el entrenamiento, podemos considerar que contamos con los siguientes ejemplos:

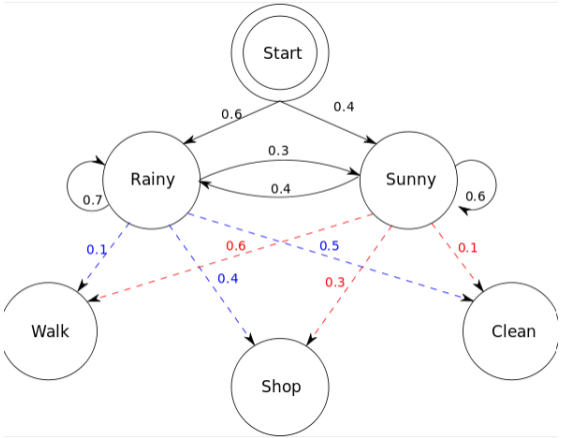
- walk shop $\mapsto$ sunny sunny
- clean shop walk $\mapsto$ rainy rainy sunny
- clean clean walk walk $\mapsto$ rainy rainy sunny sunny

En el entrenamiento se determinan los símbolos de observación y de emisión:

- **Símbolos de observación** $\Sigma = \{Clean, Walk, Shop\}$
- **Símbolos de emisión** $S = \{Rainy, Sunny\}$

A partir de estos símbolos se estiman las **probabilidades de transición** $p(s_i, s_j)$, incluyendo las probabilidades **iniciales** π_i, y las de **observación** $p(w_i|s_j)$.

Ejemplo de HMM



Probabilidades de transición:

	Rainy	Sunny
Rainy	0.7	0.4
Sunny	0.3	0.6
INIT	0.6	0.4

Probabilidades de observaciones:

	Rainy	Sunny
Walk	0.1	0.6
Shop	0.4	0.3
Clean	0.5	0.1

Cálculo de probabilidades

Si conocemos las observaciones y **las emisiones que generaron las observaciones** podemos estimar una probabilidad como dentro de un modelo bayesiano.

Por ejemplo para "walk shop" generado por "sunny sunny" tenemos:

$$\begin{aligned} p(walk, shop, sunny, sunny) &= p(sunny)p(walk|sunny)p(sunny|sunny)p(shop|sunny) \\ &= 0.4 \cdot 0.6 \cdot 0.6 \cdot 0.3 \approx 0.0432 \end{aligned}$$

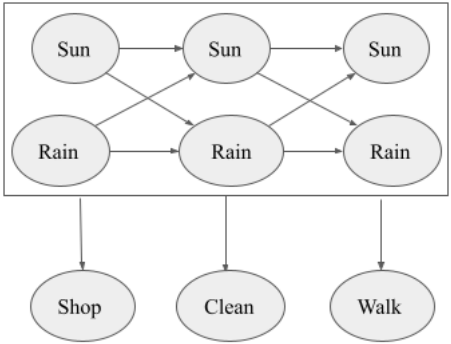
Asimismo, pueden calcularse **probabilidades marginales** (como $p(walk, shop)$).

Sin embargo, los HMMs suelen usarse como métodos de **clasificación de cadenas**; es decir, se busca determinar cuál es las emisiones que mejor describen a la cadena de observaciones.

Clasificación de cadenas

Si $|S| = N$, entonces, para una cadena de observaciones de longitud n $(w_1 ... w_n)$ se deben realizar N^n búsquedas para obtener la cadena de emisiones más probables.

Esto implica que la complejidad de un algoritmo que busque de forma arbitraria es de $\mathbf{O(N^n)}$



Algoritmos dinámicos para estimación de emisiones

Para simplificar el problema de estimar la cadena más emisiones dada una observación se propone el uso de algorimos dinámicos. En particular dos:

- **Algoritmo de avance-retroceso**: Explora los caminos posibles de emisiones en dos partes: 1) Avance, recorre caminos de atrás hacia adelante; 2) Retroceso, recorre los caminos de adelante hacia atrás. Combina estos dos casos para estimar una etiqueta adecuada.
- **Algoritmo de Viterbi:** En cada estado, considera sólo los caminos más probables precedentes, olvidándose de esos caminos que no maximicen la probabilidad.

Procedimiento de Avance

El **procedimiento de Avance** busca determinar la probabilidad de una cadena de emisiones avanzando sobre esta.

Se define una variable que almacena las probabilidades conjuntas hasta el estado t:

$$\alpha_i(t) = p(w^{(1)} w^{(2)} \ldots w^{(t)}, s_i^{(t)})$$

De tal forma que la probabilidad de una observación está determinada como:

$$p(w^{(1)} \ldots w^{(n)}) = \sum_{i=1}^{N} \alpha_i(n)$$

Algoritmo de Avance

Algorithm Procedimiento de Avance

1: **procedure** FORWARD($w^{(1)}...w^{(n)}$, HMM)
2: $\alpha_i(0) = \pi_i,\ 1 \leq i \leq N = |S|$ — Inicialización
3: **for** t **from** 0 **to** $n-1$ **do**
4: $\alpha_i(t+1) = \sum_{j=1}^{N} p(w^{(t+1)}|s_i^{(t+1)}) p(s_i^{(t+1)}|s_j^{(t)}) \alpha_j(t)$ — Inducción
5: **end for**
6: $p(w^{(1)}...w^{(n)}) \leftarrow \sum_{i=1}^{N} \alpha_i(n)$ — Terminación
7: **return** $\alpha_i(t),\ t = 1, .., n,\ i = 1, ..., N$
8: **end procedure**

Procedimiento de Retroceso

El **procedimiento de Retroceso** busca determinar la probabilidad de una cadena de emisiones retrocediendo sobre esta.

Se define una variable que almacena las probabilidades conjuntas hasta el estado t:

$$\beta_j(t) = p(w^{(t+1)} w^{(t+2)} \ldots w^{(n)} | s_j^{(t)})$$

De tal forma que la probabilidad de una observación está determinada como:

$$p(w^{(1)} \ldots w^{(n)}) = \sum_{j=1}^{N} p(s_j)\beta_j(0)$$

Algoritmo de Retroceso

Algorithm Procedimiento de Retroceso

1: **procedure** BACKWARD($w^{(1)}...w^{(n)}$, HMM)
2: $\beta_j(n+1) = 1,\ 1 \leq j \leq N = |S|$ — Inicialización
3: **for** t **from** n **to** 0 **do**
4: $\beta_j(t) = \sum_{i=1}^{N} p(w^{(t+1)}|s_i^{(t+1)}) p(s_i^{(t+1)}|s_j^{(t)}) \beta_i(t+1)$ — Inducción
5: **end for**
6: $p(w^{(1)}...w^{(n)}) \leftarrow \sum_{j=1}^{N} \pi_j \beta_j(0)$ — Terminación
7: **return** $\beta_j(t),\ t = 1, .., n,\ j = 1, ..., N$
8: **end procedure**

Procedimiento de Avance-Retroceso

Los procedimientos de avance y retroceso pueden usarse para calcular las probabilidades de cadenas de observaciones. Se pueden combinar para obtener:

$$\begin{aligned} p(w^{(1)}w^{(2)}...w^{(n)}) &= \sum_i p(w^{(1)}w^{(2)}...w^{(n)}, s_i^{(t)}) \\ &= \sum_i p(w^{(1)}...w^{(t)}, s_i^{(t)}) p(w^{(t+1)}...w^{(t)}|s_i^{(t)}) \\ &= \sum_i \alpha_i(t)\beta_i(t) \end{aligned}$$

De aquí, podemos obtener un resultado importante:

$$p(w^{(1)}w^{(2)}...w^{(n)}, s_i^{(t)}) = \alpha_i(t)\beta_i(t)$$

Predicción de etiqueta con avance retroceso

De esta forma, dada una cadena de observaciones $w^{(1)}w^{(2)}...w^{(n)}$ podemos estimar la etiqueta de la observación $t = 1, ..., n$ de la siguiente forma:

$$\begin{aligned}\hat{s}^{(t)} &= \arg\max_i p(s_i^{(t)}|w^{(1)}w^{(2)}...w^{(n)}) \\ &= \arg\max_i \frac{p(s_i^{(t)}, w^{(1)}w^{(2)}...w^{(n)})}{p(w^{(1)}w^{(2)}...w^{(n)})} \\ &= \arg\max_i \frac{\alpha_i(t)\beta_i(t)}{\sum_i \alpha_i(t)\beta_i(t)}\end{aligned}$$

Y ya que $p(s_i^{(t)}|w^{(1)}w^{(2)}...w^{(n)}) \propto p(s_i^{(t)}, w^{(1)}w^{(2)}...w^{(n)})$:

$$\hat{s}^{(t)} = \arg\max_i p(s_i^{(t)}, w^{(1)}w^{(2)}...w^{(n)})$$

Algoritmo de avance retroceso

Algorithm Procedimiento de Avance-Retroceso

procedure FORWARD-BACKWARD($w^{(1)}...w^{(n)}$, HMM)
  $\alpha_i(t) \leftarrow$ FORWARD($w^{(1)}...w^{(n)}$, HMM)
  $\beta_i(t) \leftarrow$ BACKWARD($w^{(1)}...w^{(n)}$, HMM)
  for t **from** 1 **to** n **do**
    $\hat{s}^{(t)} \leftarrow \arg\max_i \alpha_i(t)\beta_i(t)$
  end for
  return $\hat{s}^{(1)}\hat{s}^{(2)}...\hat{s}^{(n)}$
end procedure

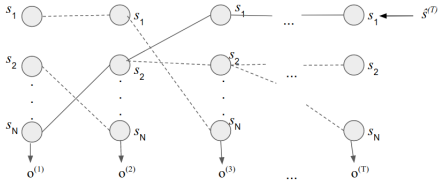
Algoritmo de Viterbi

El algoritmo de **avance-retroceso** tiene algunas **desventajas**:

- En cada iteración sólo determina el argumento que maximiza una etiqueta.
- Las etiquetas en cada estado no tienen memoria de las etiquetas tomadas en estados anteriores.

Algoritmo de Viterbi

El algoritmo de Viterbo es un algoritmo dinámico que encuentra el camino (dentro del diagrama de trellis) que maximiza la probabilidad conjunta de emisiones cuando se tiene una cadena de observaciones.



Algoritmo de Viterbo

El algoritmo de Viterbi busca **maximizar** la probabilidad de una secuencia de emisiones, observando los elementos **previos que maximizan** la probabilidad actual:

$$\max_{i_1,\dots i_{t-1}} p(s_{i_1}^{(1)},..,s_{i_{t-1}}^{(t-1)},s_j^{(t)},w^{(1)}\dots w^{(n)}) = \max_{i_1,\dots i_{t-1}} p(w^{(1)}\dots w^{(n)}|s_{i_1}^{(1)},..,s_{i_{t-1}}^{(t-1)},s_j^{(t)})$$
$$p(s_{i_1}^{(1)},..,s_{i_{t-1}}^{(t-1)},s_j^{(t)})$$
$$= \max_{i_1,\dots i_{t-1}} \prod_{t=1}^{t-1} p(w^{(t)}|s_{i_t}^{(t)})p(s_{i_t}^{(t)}|s_{i_{t-1}}^{(t-1)})$$

Este **camino máximo** hacia el símbolo s_j en el estado t se almacena en una variable:

$$\delta_j(t) = \max_{i_1,\dots i_{t-1}} p(s_{i_1}^{(1)},..,s_{i_{t-1}}^{(t-1)},s_j^{(t)},w^{(1)}\dots w^{(n)})$$

Algoritmo de Viterbi

Algorithm Algoritmo de Viterbi

procedure VITERBI($w^{(1)}...w^{(n)}$, HMM)
  $\delta_j(1) = p(w^{(1)}|s_j)\pi_j,\ 1 \leq j \leq N = |S|$
  for t **from** 1 **to** $n-1$ **do**
    $\delta_j(t+1) = \max_i p(w^{(t+1)}|s_j)p(s_j|s_i)\delta_i(t)$
    $\phi_j(t+1) = \arg\max_i p(w^{(t+1)}|s_j)p(s_j|s_i)\delta_i(t)$
  end for
  $\hat{s}^{(n)} \leftarrow \arg\max_j \delta_j(n)$
  for t **from** $n-1$ **to** 1 **do**
    $\hat{s}^t \leftarrow \phi_{\hat{s}^{(t+1)}}(t+1)$
  end for
  return $\hat{s}^{(1)}\hat{s}^{(2)}...\hat{s}^{(n)}$
end procedure

Ejemplo: Viterbi

Supóngase que se quiere etiquetar la cadena:

Shop Clean Walk

El algoritmo se basaría en los siguientes pasos:

- **Paso 0**: $\delta_{rain}(0) = 0.24$, $\delta_{sun}(0) = 0.12$
- **Paso 1**: $\delta_{rain}(1) = 0.084$ (rain), $\delta_{sun}(1) = 0.072$ (rain)
- **Paso 2**: $\delta_{rain}(2) = 0.0058$ (rain), $\delta_{sun}(2) = 0.0259$ (sun)
- **Paso 3**: $\hat{s}^3 = sun$, $\hat{s}^2 = sun$, $\hat{s}^1 = rain$

Modelos no-dirigidos

Modelos gráficos no dirigidos

A diferencia de los modelos gráficos dirigidos, los modelos no-dirigidos se basan en gráficas no-dirigidas.

Una gráfica no-dirigida $G = (V, E)$ es aquella en que si $e_{i,j} \in E$ entonces $e_{j,i} \in E$. Además, asumiremos que dada una función de peso, tenemos $\phi(e_{i,j}) = \phi(e_{j,i})$.

Modelo gráfico no-dirigido

Un modelo gráfico no-dirigido es un model gráfico representado por una gráfica no dirigida (o gráfica de factores) que factoriza la probabilidad como:

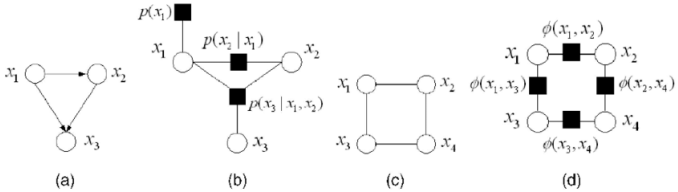
$$p(\mathbf{x}) = \frac{1}{Z} \prod_{x_i \in \pi(\mathbf{x})} \phi(x_i)$$

Donde Z es una función de partición que normaliza el producto de factores.

Gráfica de factores

Gráfica de factores

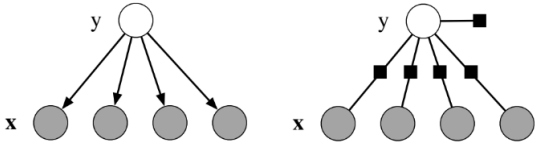
Una gráfica de factores es una gráfica no-dirigida que representa la factorización de una función de distribución. Los factores se representan como cuadros entre las conexiones de los nodos de variables aleatorias.



Clasificación con modelos no dirigidos

El problema de **clasificación** consiste en determinar una clase $Y = y$ a un elemento x, generalmente un vector $x = \begin{pmatrix} X_1 = x_1 & \cdots & X_d = x_d \end{pmatrix}$.

Una versión no-dirigida del modelo de Bayes ingenuo es:



Los factores pueden expresarse como $exp\{\theta_{y,0}\}$ para el nodo de clase y $exp\{\theta_{y,i}x_i\}$ para las conexiones entre x_i e y, $\theta_y, \theta_{y,i} \in \mathbb{R}$.

Estimación de probabilidad

Dados los factores podemos expresar la distribución probabilística como:

$$p(y|x_1, ..., x_d) = \frac{1}{Z} epx\{\theta_{y,0}\} \prod_{i=1}^{d} exp\{\theta_{y,i} x_i\}$$

$$= \frac{1}{Z} exp\Big\{ \sum_{i=1}^{d} \theta_{y,i} x_i + \theta_{y,0} \Big\}$$

Tal que $Z = \sum_y exp\Big\{ \sum_{i=1}^{d} \theta_{y,i} x_i + \theta_{y,0} \Big\}$ es la función de partición. Normaliza los factores para que se tenga una función de probabilidad.

Función logística

Los valores $\theta_{y,0}$ corresponde a los pesos por clase. Podemos expresar $\sum_{i=1}^{d} \theta_{y,i} x_i = \theta_y \cdot x$, asumiendo $\theta_y = \begin{pmatrix} \theta_{y,1} & \cdots & \theta_{y,d} \end{pmatrix}$. En un problema de clases binarias $y \in \{0, 1\}$ podemos decir que $\theta_{0,i} = 0$ para todo i:

$$\begin{aligned} p(Y = 1|x_1, ..., x_d) &= \frac{e^{\theta_1 \cdot x + \theta_{1,0}}}{e^{\theta_1 \cdot x + \theta_{1,0}} + e^0} \\ &= \frac{e^{\theta_1 \cdot x + \theta_{1,0}}}{e^{\theta_1 \cdot x + \theta_{1,0}} + 1} \\ &= \frac{1}{1 + e^{-(\theta_1 \cdot x + \theta_{1,0})}} \end{aligned}$$

Esta función se conoce como la **función logística**. Es fácil ver que $p(Y = 0|x_1, ..., x_d) = 1 - p(Y = 1|x_1, ..., x_d)$.

Función logística

Proposición

La función logística asume que el logit de la distribución es una función lineal. Esto es, que se cumple:

$$\ln \frac{p(Y=1|x)}{p(Y=0|x)} = \theta_y \cdot x + \theta_{y,0}$$

Denotemos $p := p(Y=1|x)$, de donde $p(Y=0|x) = 1-p$. Entonces tenemos:

$$\begin{aligned} p &= \frac{1}{1+e^{-(\theta_1 \cdot x + \theta_{1,0})}} \\ 1 &= p + p \cdot e^{-(\theta_1 \cdot x + \theta_{1,0})} \\ \frac{1-p}{p} &= e^{-(\theta_1 \cdot x + \theta_{1,0})} \\ -\ln \frac{1-p}{p} &= \theta_1 \cdot x + \theta_{1,0} \end{aligned}$$

Regresión logística

Regresión logística

Un modelo de regresión logística es un modelo gráfico no dirigido (discriminativo) de clasificación binaria que estima las probabilidades por medio de la función logística:

$$p(y|x_1, ..., x_d) = \frac{1}{1 + e^{-(\theta_1 \cdot x + \theta_{1,0})}}$$

Para obtener una clase se busca la clase de mayor probabilidad, se puede determinar una función de la forma:

$$\hat{y} = \begin{cases} 1 & \text{si } p(y|x) > 0.5 \\ 0 & \text{si } p(y|x) \leq 0.5 \end{cases}$$

Regresión logística

De manera general, el modelo de regresión logística puede aprender una estimación multiclase de la forma:

$$p(y|x_1, ..., x_d) = \frac{1}{Z} exp\Big\{\theta_1 \cdot x_i + \theta_{y,0}\Big\}$$

En este caso, clasificar un vector de entrada en una clase se puede realizar como:

$$\hat{y} = \arg\max_y p(y|x_1, ..., x_d)$$

El problema de la regresión consiste en obtener los valores para cada parámetro $\theta_{y,i}$. Estos se aprenden por medio de un algoritmo como el de **Gradiente descendiente**.

Aprendizaje de parámetros

Para obtener los parámetros $\theta_{y,i}$ se utiliza la regla de aprendizaje por gradiente descendiente dada como:

$$\theta_{y,i} \leftarrow \theta_{y,i} - \eta \nabla_{y,i} R(\theta)$$

Donde η es el rango de aprendizaje y $R(\theta) = -\sum_y \sum_x y \ln p(y|x)$. En particular, notamos que para la clasificación binaria tenemos:

$$\begin{aligned} -\frac{\partial}{\partial \theta_{1,i}} y \ln p(y|x) - (1-y) \ln 1 - p(y|x) &= \frac{\partial}{\partial \theta_{1,i}} y \ln 1 - e^{-(\theta_1 \cdot x + \theta_{1,0})} + (1-y) \ln 1 + e^{\theta_1 \cdot x + \theta_{1,0}} \\ &= -\Big(y - \frac{1}{1 + e^{-(\theta_1 \cdot x + \theta_{1,0})}}\Big) x_i \\ &= \Big(p(y=1|x) - y\Big) x_i \end{aligned}$$

Conditional Random Fields

En la clasificación de cadenas con modelos dirigidos, se utilizaban los modelos ocultos de Markov; de forma más general se utilizan las redes bayesianas.

Podemos definir conceptos similares para modelos gráficos no-dirigidos.

Conditional Random Field

Sea $G = (V, E)$ una gráfica cuyos nodos están indexados con variables aleatorias $\mathbb{Y} = (Y_v)$, $v \in V$. Decimos que $(\mathbf{X}, \mathbf{Y})$, $\mathbf{X}$ observaciones, es una campo condicional aleatorio (Conditional Random Field o CRF) si se cumple que:

$$p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$$

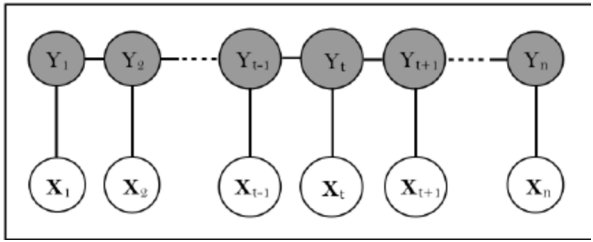
Donde $w \sim v$ indica que los nodos w y v son vecinos en G.

Linear chain CRF

Linear chain CRF

Un CRF se dice que es de cadena lineal si los vértices sobre $\mathbf{Y}$ definen una cadena; es decir, si $V = \{1, 2, ..., n\}$ y $e_{i,i+1} \in E$.

En este caso, se tiene una cadena de observaciones $X_1, ..., X_n$ y una cadena de emisiones $Y_1, ..., Y_n$ relacionados por la siguiente gráfica:



Función de rasgos

Feature function

Sea y una clase. Una función de rasgos (feature function) $f_{i,j}$ es una función que es diferente de 0 sólo para una clase y, y se define como:

$$f_{i,j}(y, x) = \delta_{i=y} x_j$$

Donde $\delta_{i=y} = 1$ si $i = y$ y 0 en otro caso.

Una notación que usaremos también es la siguiente:

$$f_{i,j}(y, x) = \delta_{i=y} \delta_{j=x}$$

De igual forma, tenemos:

$$f_{i,j}(y, y') = \delta_{i=y} \delta_{j=y'}$$

Linear chain CRF

Linear chain CRF

Sea $\mathbf{y} = y_1, ..., y_n$ una cadena y $\mathbf{x} = x_1, ..., x_n$ una cadena de observaciones. Una CRF de cadena lineal es un modelo gráfico no-dirigido que estima la probabilidad:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^{n} exp\Big\{ \sum_{i,j \in S} \theta_{i,j} f_{i,j}(y_t, y_{t-i}) + \sum_{i \in S} \sum_{o \in \Sigma} \theta_{i,o} f_{i,o}(y_t, x_t) \Big\}$$

Tal que $\theta_{i,j}, \theta_{i,o}$ refieren a los parámetros de la estimación.

En este caso, podemos ver que los factores de la gráfica están determinados como:

$$\psi_t(y_t, y_{t-1}, x_t) = exp\Big\{ \sum_{i,j \in S} \theta_{i,j} f_{i,j}(y_t, y_{t-i}) + \sum_{i \in S} \sum_{o \in \Sigma} \theta_{i,o} f_{i,o}(y_t, x_t) \Big\}$$

Clasificaicón de cadenas

De los factores arriba obtenidos, tenemos que:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^{n} \psi_t(y_t, y_{t-1}, x)$$

Para obtener la cadena de salida que mejor describa a la cadena de observaciones, entonces debemos obtener:

$$\widehat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$$

El aprendizaje de los parámetros se hará por medio del método de **gradiente descendiente**.

CRFs y HMMs

Existe una relación entre los campos condicionales aleatorios y los modelos ocultos de Markov, si en lugar de aprender los parámetros con GD los determinamos de la siguiente forma:

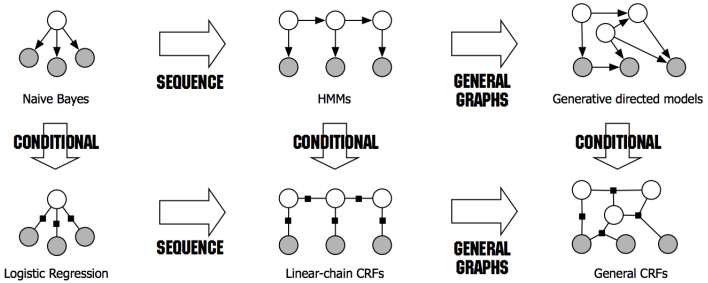
$$\theta_{i,j} = \ln p(s^{(i)}|s^{(i)})$$
$$\theta_{i,o} = \ln p(w^{(o)}|s^{(i)})$$

Siendo $w_o \in \Sigma$ símbolos de observaciones y $s_i, s_j \in S$ de emisiones. De donde obtenemos que:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^{n} exp\Big\{ \sum_{i,j \in S} \ln p(s^{(i)}|s^{(i)}) f_{i,j}(y_t, y_{t-i}) + \sum_{i \in S} \sum_{o \in \Sigma} \ln p(w^{(o)}|s^{(i)}) f_{i,o}(y_t, x_t) \Big\}$$
$$= \frac{1}{Z(x)} \prod_{t=1}^{n} exp\Big\{ \ln p(s^{(t)}|s^{(t-1)}) + \ln p(w^{(t)}|s^{(t)}) \Big\}$$
$$= \frac{1}{Z(x)} \prod_{t=1}^{n} p(s^{(t)}|s^{(t-1)}) p(w^{(t)}|s^{(t)}$$

Relaciones entre modelos dirigidos y no dirigidos

Los modelos dirigidos y no dirigidos definen familias de modelos gráficos que se pueden usar para clasificar objetos simples o cadenas de estos. Asimismo sirven para estimar probabilidades cuando las relaciones son más complejas:



Textos recomendados

Russel, S. y Norvig, P. (2021). "13. Probailistic Reasoning". *Artifitial Intelligence: A Modern Approach*, Pearson, pp. 430-478.

Joshi, P. (2017). "2. Classification and Regression Using Supervised Learning". *Artificial Intelligence with Python*, Packt, pp. 33-68.

Joshi, P. (2017). "11. Probabilistic Reasoning for Sequential Data". *Artificial Intelligence with Python*, Packt, pp. 280-204.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligen Systems*, Morgan Kaufman eds.

Sutton, C. y McCallon, A. (2011). *An Introduction to Conditional Random Fields*.