Aprendizaje generativo

Lección 09

Dr. Pablo Alvarado Moya

CE5506 Introducción al reconocimiento de patrones
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

II Semestre, 2019

Contenido

Aprendizaje discriminador

- Hasta ahora, aprendizaje basado en $p(y|\underline{\mathbf{x}};\underline{\boldsymbol{\theta}})$
  - Regresión logística: $p(y|\underline{\mathbf{x}};\underline{\boldsymbol{\theta}}) = h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}) = g(\underline{\boldsymbol{\theta}}^T\underline{\mathbf{x}})$ con $g(\cdot)$ sigmoidal
- Concepto actual ha **particionado** el **espacio de características** con un **borde de decisión**
- Clasificación se reduce a evaluar en qué lado del borde de decisión está la entrada
- Algoritmos que aprenden $p(y|\underline{\mathbf{x}})$ directamente se llaman algoritmos **discriminadores**
- Pueden aprender $h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}) \in \{0,1\}$ directamente

Aprendizaje generativo

- Otra idea: aprender $p(\underline{\mathbf{x}}|y)$ y $p(y)$
- Ejemplo: Aprendemos características de forma/textura para
  - cancer benigno
  - cancer maligno
- Para cada clase aprendemos un modelo
- Para una entrada, deben probarse todos los modelos y se selecciona el más probable
- Este enfoque se denomina **aprendizaje generativo**

Análisis gaussiano discriminador

(Primer método generativo)

Análisis gaussiano discriminador

Gaussian Discriminant Analysis

- Supongamos que $\underline{\mathbf{x}} \in \mathbb{R}^n$ son continuos
- Además supongamos que $p(\underline{\mathbf{x}}|y)$ es gaussiano

$$p(\underline{\mathbf{x}}|y) \sim \mathcal{N}(\underline{\boldsymbol{\mu}},\boldsymbol{\Sigma})$$

  - media $\underline{\boldsymbol{\mu}}$
  - matriz de covarianza $\boldsymbol{\Sigma}$

Análisis gaussiano discriminador (1)

- Supongamos que

$$
\begin{aligned}
p(y) &= \phi^y(1-\phi)^{1-y} \\
p(\underline{\mathbf{x}}|y=0) &= \frac{1}{\sqrt{(2\pi)^n|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(\underline{\mathbf{x}}-\underline{\boldsymbol{\mu}}_0)^T\mathbf{\Sigma}^{-1}(\underline{\mathbf{x}}-\underline{\boldsymbol{\mu}}_0)\right) \\
p(\underline{\mathbf{x}}|y=1) &= \frac{1}{\sqrt{(2\pi)^n|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(\underline{\mathbf{x}}-\underline{\boldsymbol{\mu}}_1)^T\mathbf{\Sigma}^{-1}(\underline{\mathbf{x}}-\underline{\boldsymbol{\mu}}_1)\right)
\end{aligned}
$$

- Buscamos entonces maximizar la verosimilitud logarítmica

$$
\ell(\phi,\underline{\boldsymbol{\mu}}_0,\underline{\boldsymbol{\mu}}_1,\mathbf{\Sigma}) = \ln \underbrace{\prod_{i=1}^{m} p(\underline{\mathbf{x}}^{(i)},y^{(i)})}_{\text{verosimilitud conjunta}} = \ln \prod_{i=1}^{m} p(\underline{\mathbf{x}}^{(i)}|y^{(i)})p(y^{(i)})
$$

Análisis gaussiano discriminador (2)

- Esta verosimilitud

$$\ell(\phi,\underline{\boldsymbol{\mu}}_0,\underline{\boldsymbol{\mu}}_1,\boldsymbol{\Sigma}) = \ln \prod_{i=1}^{m} p(\underline{\mathbf{x}}^{(i)}|y^{(i)})p(y^{(i)})$$

contrasta con la verosimilitud utilizada en regresión logística

$$\ell(\underline{\boldsymbol{\theta}}) = \ln \prod_{i=1}^{m} p(y^{(i)}|\underline{\mathbf{x}}^{(i)};\underline{\boldsymbol{\theta}})$$

Máxima verosimilitud

- Maximizando la verosimilitud anterior se obtiene:

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}$$

$$\underline{\boldsymbol{\mu}}_0 = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}\underline{\mathbf{x}}^{(i)}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}}$$

$$\underline{\boldsymbol{\mu}}_1 = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}\underline{\mathbf{x}}^{(i)}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}$$

$$\mathbf{\Sigma} = \frac{1}{m}\sum_{i=1}^{m}(\underline{\mathbf{x}}^{(i)} - \underline{\boldsymbol{\mu}}_{y^{(i)}})(\underline{\mathbf{x}}^{(i)} - \underline{\boldsymbol{\mu}}_{y^{(i)}})^T$$

Predicción

- Observemos que con la regla de Bayes, puede recalcularse:

$$p(y=1|\underline{\mathbf{x}}) = \frac{p(\underline{\mathbf{x}}|y=1)p(y)}{p(\underline{\mathbf{x}})}$$

$$p(\underline{\mathbf{x}}) = p(\underline{\mathbf{x}}|y=0)p(y=0) + p(\underline{\mathbf{x}}|y=1)p(y=1)$$

- Sin embargo, $p(\underline{\mathbf{x}})$ usualmente es innecesario pues para la predicción basta con:

$$\arg\max_y p(y|\underline{\mathbf{x}}) = \arg\max_y \frac{p(\underline{\mathbf{x}}|y)p(y)}{p(\underline{\mathbf{x}})} = \arg\max_y p(\underline{\mathbf{x}}|y)p(y)$$

- Si $p(y)$ es uniforme (i. e. $p(y=0) = p(y=1)$) entonces: $\arg\max_y p(\underline{\mathbf{x}}|y)$

Relación entre GDA y LR (1)

GDA:

- Dado el conjunto de entrenamiento $(\underline{\mathbf{x}}^{(i)}, y^{(i)})$
- Calcular con el conjunto los parámetros $\underline{\boldsymbol{\mu}}_i$, $\boldsymbol{\Sigma}$ y $p(y)$
- Para predecir probabilidad de y dado un valor de $\underline{\mathbf{x}}$:
  - Calculamos con parámetros $p(\underline{\mathbf{x}}|y=0) = \mathcal{N}(\underline{\boldsymbol{\mu}}_0, \sigma_0^2)$ y $p(\underline{\mathbf{x}}|y=1) = \mathcal{N}(\underline{\boldsymbol{\mu}}_1, \sigma_1^2)$
  - Con eso calculamos

$$p(y=1|\underline{\mathbf{x}}) = \frac{p(\underline{\mathbf{x}}|y=1)p(y=1)}{p(\underline{\mathbf{x}})}$$

donde $p(\underline{\mathbf{x}})$ se calcula con

$$p(\underline{\mathbf{x}}) = p(\underline{\mathbf{x}}|y=0)p(y=0) + p(\underline{\mathbf{x}}|y=1)p(y=1)$$

- Ver `gda_lr.m`

Relación entre GDA y LR (2)

- Si vemos a $p(y = 1|\underline{\mathbf{x}}; \phi, \underline{\boldsymbol{\mu}}_0, \underline{\boldsymbol{\mu}}_1, \boldsymbol{\Sigma})$ como función de $\underline{\mathbf{x}}$, se puede demostrar que es

$$p(y = 1|\underline{\mathbf{x}}; \phi, \underline{\boldsymbol{\mu}}_0, \underline{\boldsymbol{\mu}}_1, \boldsymbol{\Sigma}) = \frac{1}{1 + \exp(\underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}})}$$

  con $\underline{\boldsymbol{\theta}}$ dependiente de $\phi, \underline{\boldsymbol{\mu}}_0, \underline{\boldsymbol{\mu}}_1, \boldsymbol{\Sigma}$
- Esto tiene exactamente la misma forma de la regresión logística (¡algoritmo discriminador!)
- Diferencia: estructura exacta de $\underline{\boldsymbol{\theta}}$

Ventajas y desventajas de algoritmos generativos

- En GDA supusimos $\underline{\mathbf{x}}|y \sim$ gaussiano
- Eso implica que la distribución *a-posteriori* $p(y = 1|\underline{\mathbf{x}})$ es logística
- Lo contrario **no** es cierto: logístico $\not\Rightarrow$ $\underline{\mathbf{x}}|y \sim$ gaussiano
- (por ejemplo, si $\underline{\mathbf{x}}|y \sim$ Poisson también la probabilidad *a-posteriori* es logística)
- ¡Eso implica que suposición del GDA es más fuerte!
- Si la suposición es cierta, entonces GDA es mejor que la regresión logística
- Si no se sabe qué distribución tienen los datos, entonces la regresión logística es una mejor elección
- GDA funciona a veces mejor con pocos datos
- Regresión logística requiere por lo general más datos

Clasificador bayesiano ingenuo

(Segundo método generativo)

Ejemplo de motivación

- Supongamos que queremos construir un filtro de *spam* para correo-e
- No-spam y spam lo representamos con $y \in \{0,1\}$ respectivamente
- Esto es parte del área de clasificación de texto
- Supongamos que tenemos un conjunto de m correos para entrenamiento
- Necesitamos especificar las características x_i con que representaremos un correo-e

Características

- Representación usa un vector de dimensión igual al número de palabras en el diccionario
- Si el correo-e contiene la i-ésima palabra del diccionario usamos $x_i = 1$, y caso contrario $x_i = 0$
- Por ejemplo

$$\underline{\mathbf{x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \text{a} \\ \text{ababa} \\ \text{ababillarse} \\ \vdots \\ \text{compra} \\ \vdots \\ \text{zwingliano} \end{array}$$

Vocabulario

- Conjunto de palabras codificadas en el vector de características se llama **vocabulario**
- Tamaño del vocabulario igual a dimensión de $\underline{\mathbf{x}}$
- Si tenemos un vocabulario de 50 000 palabras, entonces $\underline{\mathbf{x}} \in \{0; 1\}^{50000}$
- Queremos armar un modelo generativo, así que necesitamos un modelo para $p(\underline{\mathbf{x}}|y)$
- Obviamente no es posible modelar cada $\underline{\mathbf{x}}$ explícitamente con un modelo multinomial, pues tendríamos $2^{50\,000}$ posibles configuraciones, ¡lo que implica un vector de configuración de $(2^{50\,000} - 1)$ dimensiones!

Suposición de Bayes ingenua

- Vamos a suponer entonces que los x_i en $\underline{\mathbf{x}}$ son **condicionalmente** independientes entre sí, dada y, es decir:

$$p(x_j|y) = p(x_j|y, x_i)$$

- Esto es, suponemos que si el correo es spam, la ocurrencia de una palabra i no depende de la ocurrencia de una palabra j (¡lo que en realidad es falso!)
- **¡Advertencia!** eso **no** es lo mismo que x_i y x_j sean independientes
- Esto se denomina suposición de Bayes **ingenua**
- El **clasificador de Bayes ingenuo** resulta de esta suposicón

Probabilidad conjunta condicional

$$
\begin{aligned}
p(\underline{\mathbf{x}}|y) &= p(x_1, \ldots, x_{50\,000}|y) \\
&= p(x_1|y)p(x_2|y,x_1)p(x_3|y,x_1,x_2)\cdots p(x_{50\,000}|y,x_1,\ldots,x_{49\,999}) \\
&= p(x_1|y)p(x_2|y)p(x_3|y)\cdots p(x_{50\,000}|y) \\
&= \prod_{i=1}^{n} p(x_i|y)
\end{aligned}
$$

- A pesar de que esta suposición es muy fuerte, el método funciona.
- El modelo se parametriza con $\phi_{i|y=1} = p(x_i = 1|y = 1)$, $\phi_{i|y=0} = p(x_i = 1|y = 0)$ y $\phi_y = p(y = 1)$

Máxima verosimilitud (1)

- Dado el conjunto de entrenamiento $\{(\underline{\mathbf{x}}^{(i)}, y^{(i)}); i = 1, \ldots, m\}$, la verosimilitud conjunta de los datos es

$$L\left(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}\right) = \prod_{i=1}^{m} p\left(\underline{\mathbf{x}}^{(i)}, y^{(i)}\right)$$

Máxima verosimilitud (2)

- Si se maximiza $L\left(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}\right)$ con respecto a los parámetros, se obtiene el estimado de máxima verosimilitud:

$$\phi_{j|y=1} = p(x_j = 1|y = 1) = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}$$

$$\phi_{j|y=0} = p(x_j = 1|y = 0) = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}}$$

$$\phi_y = p(y = 1) = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

- Interpretaciones "fáciles"...

Máxima verosimilitud (3)

- Con estos parámetros, para hacer la predicción en un nuevo correo $\underline{\mathbf{x}}$ solo calculamos:

$$
\begin{aligned}
&p(y=1|\underline{\mathbf{x}})\\
&\quad= \frac{p(\underline{\mathbf{x}}|y=1)p(y=1)}{p(\underline{\mathbf{x}})}\\
&\quad= \frac{\left(\prod_{i=1}^{n} p(x_i|y=1)\right)p(y=1)}{\left(\prod_{i=1}^{n} p(x_i|y=1)\right)p(y=1) + \left(\prod_{i=1}^{n} p(x_i|y=0)\right)p(y=0)}
\end{aligned}
$$

- Elegimos la clase que tenga la probabilidad *a-posteriori* mayor

Caso multinomial

- Desarrollamos el algoritmo de Bayes ingenuo para características de **entrada** x_i binarias
- Nada impide usar características $x_i \in \{1,2,\dots k_i\}$
- En ese caso modelamos $p(x_i|y)$ con una distribución multinomial en vez de Bernoulli
- En la práctica, en problemas con entradas continuas, se obtienen buenos resultados si se **discretiza** la entrada y se usa el algoritmo de Bayes ingenuo
(por ejemplo, si datos no siguen una distribución normal multivariada)

Suavizamiento de Laplace

Problema con suposición ingenua (1)

- El algoritmo ingenuo de Bayes funciona en bastantes problemas
- Un cambio simple lo mejora, especialmente para clasificación textual

Problema con suposición ingenua (2)

- Una nueva palabra k que no estuvo en el conjunto de entrenamiento tendrá:

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_k^{(i)} = 1 \land y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}} = 0$$

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_k^{(i)} = 1 \land y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}} = 0$$

Problema con suposición ingenua (3)

- Como la palabra no es ni spam ni no-spam, ¡la probabilidad de que cualquiera ocurra es cero!
- Si queremos decidir qué tipo de correo es uno que contenga la k-ésima palabra se obtiene:

$$p(y=1|\underline{\mathbf{x}}) = \frac{p(\underline{\mathbf{x}}|y=1)p(y=1)}{p(\underline{\mathbf{x}})}$$

$$= \frac{\left(\prod_{i=1}^{n} p(x_i|y=1)\right) p(y=1)}{\left(\prod_{i=1}^{n} p(x_i|y=1)\right) p(y=1) + \left(\prod_{i=1}^{n} p(x_i|y=0)\right) p(y=0)} = \frac{0}{0}$$

Suavizamiento de Laplace (1)

- Estadísticamente es mala idea suponer que la probabilidad de un evento es cero solo porque no se ha visto en el conjunto de entrenamiento.
- Para m observaciones, estimación de máxima verosimilitud es:

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{x^{(i)} = j\right\}}{m}$$

- Con esta estimación algunos ϕ_j pueden llegar a ser cero, lo que se evita con el **suavizamiento de Laplace**, que lo reemplaza con

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{x^{(i)} = j\right\} + 1}{m + k}$$

Suavizamiento de Laplace (2)

- k es el número de posibles valores que puede tomar $x^{(i)}$ (en el caso binario $k = 2$)
- Note que aún se cumple $\sum_{j=1}^{k} \phi_j = 1$ y $\phi_j \neq 0$
- El estimado de los parámetros del clasificador de Bayes ingenuo con suavizamiento de Laplace son entonces:

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \land y^{(i)} = 1\right\} + 1}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\} + 2}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \land y^{(i)} = 0\right\} + 1}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\} + 2}$$

Modelos de eventos

Modelo de eventos Bernoulli multivariado

- Hasta hora hemos supuesto un modelo multivariado de eventos Bernoulli :
  1. Se genera un correo de spam con probabilidad $p(y)$
  2. El generador de correos-e recorre todas las palabras del diccionario e incluye la i-ésima palabra con probabilidad $p(x_i = 1|y) = \phi_{i|y}$
  3. x_i es la presencia de la i-ésima palabra del vocabulario en el correo
  4. La probabilidad de un mensaje es entonces $p(y)\prod_{i=1}^{n} p(x_i|y)$
- Existe otro enfoque...

Modelo de eventos multinomial (1)

- En el modelo de eventos multinomial usamos otra notación y conjunto de características:
- x_i denota *cuál* es la i-ésima palabra en el correo
- $x_i \in \{1, \ldots, |V|\}$ con $|V|$ el tamaño del vocabulario
- Un correo de n palabras se representa con el vector $(x_1, x_2, \ldots, x_n)$ (note que el tamaño de cada correo varía)
- El modelo de generación es entonces:
  1. Se genera un correo de spam con probabilidad $p(y)$
  2. La primera palabra del diccionario x_1 se genera de una distribución multinomial $p(x_1|y)$
  3. La segunda palabra, independientemente de la primera, se genera de la misma distribución multinomial
  4. Así sucesivamente para todas las n palabras del correo-e
  5. La probabilidad de un mensaje es entonces $p(y)\prod_{i=1}^{n} p(x_i|y)$

Modelo de eventos multinomial (2)

- Los parámetros del modelo son ahora:
  1. $\phi_y = p(y)$ como antes
  2. $\phi_{k|y=1} = p(x_j = k|y = 1)$ (para cualquier j)
  3. $\phi_{k|y=0} = p(x_j = k|y = 0)$ (para cualquier j)
- Note que hemos supuesto que $p(x_j|y)$ es la misma para todo j (independencia de posición)

Máxima verosimilitud en modelo multinomial (1)

- Dado el conjunto de entrenamiento $\{(\underline{\mathbf{x}}^{(i)},y^{(i)}); i = 1,\ldots,m\}$, la verosimilitud es

$$
\begin{aligned}
&L(\phi_y,\phi_{k|y=0},\phi_{k|y=1}) \\
&\qquad = \prod_{i=1}^{m} p(\underline{\mathbf{x}}^{(i)},y^{(i)}) \\
&\qquad = \prod_{i=1}^{m} \left(\prod_{j=1}^{n_i} p(x_j^{(i)}|y;\phi_{k|y=0},\phi_{k|y=1}) \right) p(y^{(i)};\phi_y)
\end{aligned}
$$

Máxima verosimilitud en modelo multinomial (2)

- Su maximización resulta en

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\} n_i}$$

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\} n_i}$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

Máxima verosimilitud en modelo multinomial

Con suavizamiento de Laplace

Si consideramos el suavizamiento de Laplace:

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 1\right\} + 1}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\} n_i + |V|}$$

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\} n_i + |V|}$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

con $|V|$ el tamaño del diccionario.

Conclusiones

- Aunque el clasificador de Bayes ingenuo no es el "mejor", con frecuencia funciona bastante bien
- Facilidad de implementación hace que sea una de las "primeras cosas que probar".

Resumen

1. Introducción
   - Aprendizajes discriminador y generativo
2. Métodos generativos
   - Análisis gaussiano discriminador
   - Clasificador bayesiano ingenuo
     - Suavizamiento de Laplace
   - Modelos de eventos

Este documento ha sido elaborado con software libre incluyendo LATEX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, GNU-Make y Subversion en GNU/Linux