**UNIVERSIDAD CATÓLICA BOLIVIANA “SAN PABLO”**
**FACULTAD DE INGENIERÍA**
**CARRERA INGENIERÍA DE SISTEMAS**
**MAESTRÍA EN CIENCIA DE DATOS**

# Naïve Bayes

# Aprendizaje Supervisado: Naïve Bayes

- Se caracteriza por realizar predicciones probabilísticas
- Se basa en el Teorema de Bayes
- Los modelos generados por esta técnica tienen un alto rendimiento, en algunos casos comparable a los árboles de decisión.
- Cada ejemplo del training set puede incrementar o reducir la probabilidad de que la hipótesis sea correcta.

# El Teorema de Bayes

- Sea **X** una muestra de los datos (“evidencia”): La clase de la etiqueta es desconocida.
- Sea H la hipótesis que X pertenece a la clase C
- La clasificación se realiza por medio de la determinación de que P(H|**X**) (**posteriori probabilit**y), es decir, la probabilidad de **H dado X.**
- P(H) (***prior probability***), la probabilidad inicial de la hipótesis.
- P(**X**): (***prior probability of predictor***) la probabilidad inicial de la evidencia
- P(**X**|H) (**likelyhood**), la probabilidad de observar X dada la hipótesis H.

# El Teorema de Bayes (cont.)

- Dados el training set ***X****, posteriori probability de la hipótesis* H*, P(H|**X**),* sigue el teorema de Bayes.

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

- Informalmente, se puede escribir como

  posteriori = likelihood *x* prior/evidence

- Predice ***X*** que pertence a $C_i$ si y solo si la probabilidad $P(C_i|\mathbf{X})$ es la más alta entre todos los $P(C_k|X)$ para todas las *k* clases
- Dificultad: Se require conocer varias probabilidades del dataset en cuestión.

# Hacia un clasificador Naïve Bayes

- Sea *D* un training set de tuplas y sus etiquetas de clases asociadas, cada tupla es representada por una instancia *n-D* de vector $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$
- Suponga que existen *m* clases $C_1, C_2, \ldots, C_m$.
- La clasificación es derivar la máxima posteriori probability, i.e., el máximo $P(C_i|\boldsymbol{X})$
- Lo cual se deriva del Teorema de Bayes

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Siendo *P(X)* constante para todas las clases, solamente

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

necesita ser maximizada

# Derivación de un clasificador Naïve Bayes

- Suposición: los atributos son condicionalmente independientes (i.e., ninguna relación de dependencia entre atributos):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i)$$

- Si el atributo $A_k$ es categorico, $P(x_k|C_i)$ es el número de tuplas en $C_i$ que tienen valor $x_k$ para $A_k$ dividido por $|C_{i,D}|$ (Nro. de tuplas de $C_i$ en D)
- Si $A_k$ es continua, $P(x_k|C_i)$ es usualmente calculada sobre la base de la distribución Gaussian con media $\mu$ y Des. Standard $\sigma$

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

y $P(x_k|C_i)$ es

$$P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

# Training set

| age | income | student | credit_rating | buy_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31..40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31..40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31..40 | medium | no | excellent | yes |
| 31..40 | high | yes | fair | yes |
| >40 | medium | np | excellent | no |

# Un ejemplo de Naïve Bayes

- $P(C_i)$: P(buys_computer = “yes”) = 9/14 = 0.643

  P(buys_computer = “no”) = 5/14= 0.357

- Calcula $P(X|C_i)$ para cada clase

  P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222
  P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6
  P(income = “medium” | buys_computer = “yes”) = 4/9 = 0.444
  P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4
  P(student = “yes” | buys_computer = “yes) = 6/9 = 0.667
  P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2
  P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667
  P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

**$P(X|C_i)$ :** P(X|buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

**$P(X|C_i)*P(C_i)$ :** P(X|buys_computer = “yes”) * P(buys_computer = “yes”) = 0.028

P(X|buys_computer = “no”) * P(buys_computer = “no”) = 0.007

**Es por eso que X pertence a la clase (“buys_computer = yes”)**

# Evitar el problema de la probabilidad cero

- La predicción Naïve Bayesian require que cada probabilidad condicional sea diferente a cero

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Ex. Dado un dataset con 1000 tuplas, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (o Laplacian estimator)
  - *Adicione 1 a cada caso*

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - La probabilidad “corregida” es cercana a su correspondiente probabilidad no “corregida”.

# Ventajas

- Necesita menor cantidad de datos de entrenamiento
- Es simple y rápido al momento de la predicción del test dataset
- Tiene un buen rendimiento en predicciones multi-clase
- Cuando las variables son independientes puede tener un rendimiento similar al de la Regresión Logística
- Altamente escalable en número de predictores y datos
- Como es rápido, se puede utilizar en predicciones en tiempo real.
- No es sensitivo a variables irrelevantes.

# Desventajas

- Suposición: Independencia condicional de clase
- Existencia de dependencias entre variables.
- El problema de “Zero Frequency”

# Tipos de Modelos

- **Gaussian**: es utilizado en clasificación, asume que las variables siguen una distribución normal.

- **Multinomial**: Es utilizado para casos discretos. Si tenemos el problema de clasificación de texto se considera “ocurrencia de palabras en documento”, tenemos el conteo de la frecuencia de ocurrencia de palabras en documentos.

- **Bernoulli**: Este modelo es útil cuando las variables son binarias (i.e. ceros y unos). Una aplicación podría ser clasificación de texto con el modelo “Bag of Words”, donde “1” cuando una palabra ocurre en el documento y “0” cuando no ocurre en el mismo.

# Gaussian Naïve Bayes

- Cuando se cuenta con atributos del tipo continuo se utiliza Naïve Bayes tipo Gaussian.
- Una consideración de relevancia es que los atributos se distribuyan Normal (Gaussian).
- Sea $\mu_y$ la media del valor de *x* asociada con la clase *y*, y sea $\sigma_y^2$ la varianza de valores de *x* asociados con la clases *y*
- La distribución de probabilidades de $x_i$ dada la clase *y* es:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

# Bernoulli Naïve Bayes

- En este caso los atributos son independientes y tienen valores binarios en variables
- Generalmente se utiliza esta técnica para clasificación de documentos, donde la ocurrencia de los atributos es binaria, en vez de tener frecuencias de valores. Bernoulli Naive Byes se basa en:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

- Esta técnica difiere de la Multinomial Naive Bayes en que explícitamente penaliza la no ocurrencia de una variable *i* que es el indicador para la clase *y*.

# Multinomial Naïve Bayes

- La distribución es parametrizada por los vectores $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ para cada clase *y*, donde *n* es el número de variables (en clasificación de texto el tamaño del vocabulario) y $\boldsymbol{\theta_{yi}}$ **es la probabilidad de** $\boldsymbol{P(x_i \mid y)}$ de la variable *i* que aparece en el dataset y pertenece a la clase *y*.

- La variable  es estimada por medio de la version del suavizado (ing. Smoothed) de Maximun Likelyhood, i.e. conteo de frecuencia relativa.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

- Donde $N_{yi} = \sum_{x \in T} x_i$ Es el número de veces que la variable *i* aparece en el dataset T de la clase y $N_y = \sum_{i=1}^{n} N_{yi}$ es la cuenta total de todas las variables para la clase *y*.

- El suavizado $\alpha > 0$ se aplica para variables que no están presentes en el dataset, previene zero probabilidades, Cuando $\alpha = 1$ Laplace Smothing.

# Consideraciones para mejorar los modelos

- Se debiera quitar los atributos correlacionados ya que estos son tomados en cuenta dos veces en el Modelo, lo cual puede llevar a inflar la importancia de estos.

- El clasificador Naïve Bayes cuenta con limitadas opciones de afinado de parámetros como alpha=1 (Laplace smoothing), fit_prior, aprende la probabilidad prior de clase o no, etc. Se recomienda enfocarse en el pre-procesamiento de datos y en la selección de atributos.

- Se pudiera pensar en aplicar alguna técnica de combinación de clasificadores como Ensembling, Boosting, Bagging, pero no serían de ayuda. Estas técnicas se utilizan para reducir la varianza , pero Naive Bayes no tiene varianza que minimizar.

# Aplicaciones

- Predicciones en tiempo real, es una posibilidad por su rapidez
- Predicción multiclase
- Clasificación de textos
- Filtrado de Spam
- Análisis del Sentimiento
- Sistemas de Recomendaciones

# Bibliografía


- Jiawei Han, Data Mining: Concepts and Techniques, 3rd edition, 2012. Chapter 8 - 8.3

- Analytics Vidhya: ”Six easy steps to learn Naïve Bayes”, https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

- Sklearn: “Naïve Bayes”, https://en.wikipedia.org/wiki/Multinomial_distribution