Introduction to Big Data and Machine Learning Classification

Dr. Mihail

September 19, 2019

Linear models for classification

Goal

- Goal of classification: take an input vector x and assign it to one of K discrete classes $\mathcal{C}_k$ where $k = 1, \ldots, K$
- The input space is therefore divided into *decision regions* whose boundaries are called "*decision bounderies*" or "decision surfaces"
- Here, we will consider linear models where the decision boundaries are linear functions of the input vector "x" and hence are definded by $D - 1$-dimensional hyperplanes within the D-dimensional input space
- Data sets that can be separated exactly by linear decision surfaces are said to be "*linearly separable*"

Probabilistic models

- For probabilistic models, the most convenient, in the case of two-class problems is the binary representation, in which there is a single target variable $t = \{0, 1\}$
- For $K > 2$ classes, it is convenient to use $1 - of - K$ coding scheme, in which t is a vector of length K such that if the class is $\mathcal{C}_j$, then all elements of t_k are zero exacept t_j.
- For instance if we have 5 classes, the a patter from class 2 would be given by the target vector $t = (0, 1, 0, 0, 0)^T$

Using Bayes Theorem

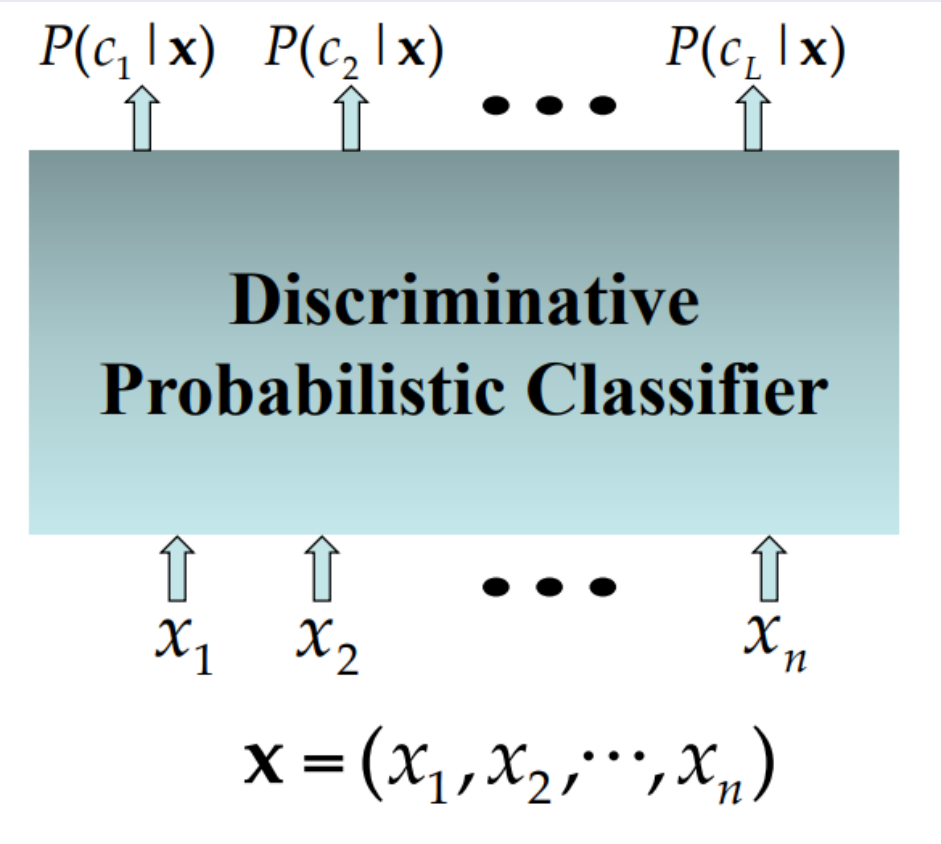
- Model posterior class conditional probability: $p(\mathcal{C}_k|x) = \frac{p(x|\mathcal{C}_k)p(\mathcal{C}_k)}{p(x)}$
- **Notice** the denominator is not a function of $\mathcal{C}$
- Prior class distribution: $p(\mathcal{C}_k)$
- Class conditional density: $p(x|\mathcal{C}_k)$

Discriminative models

Discriminative model

- $P(c|x)$
- To train a discriminative classifier, all training examples of different classes must be jointly used to build up a single discriminative classifier
- Output K probabilities for K class labels in probabilistic classifiers, while a single label is produced by non-probabilistic classifier

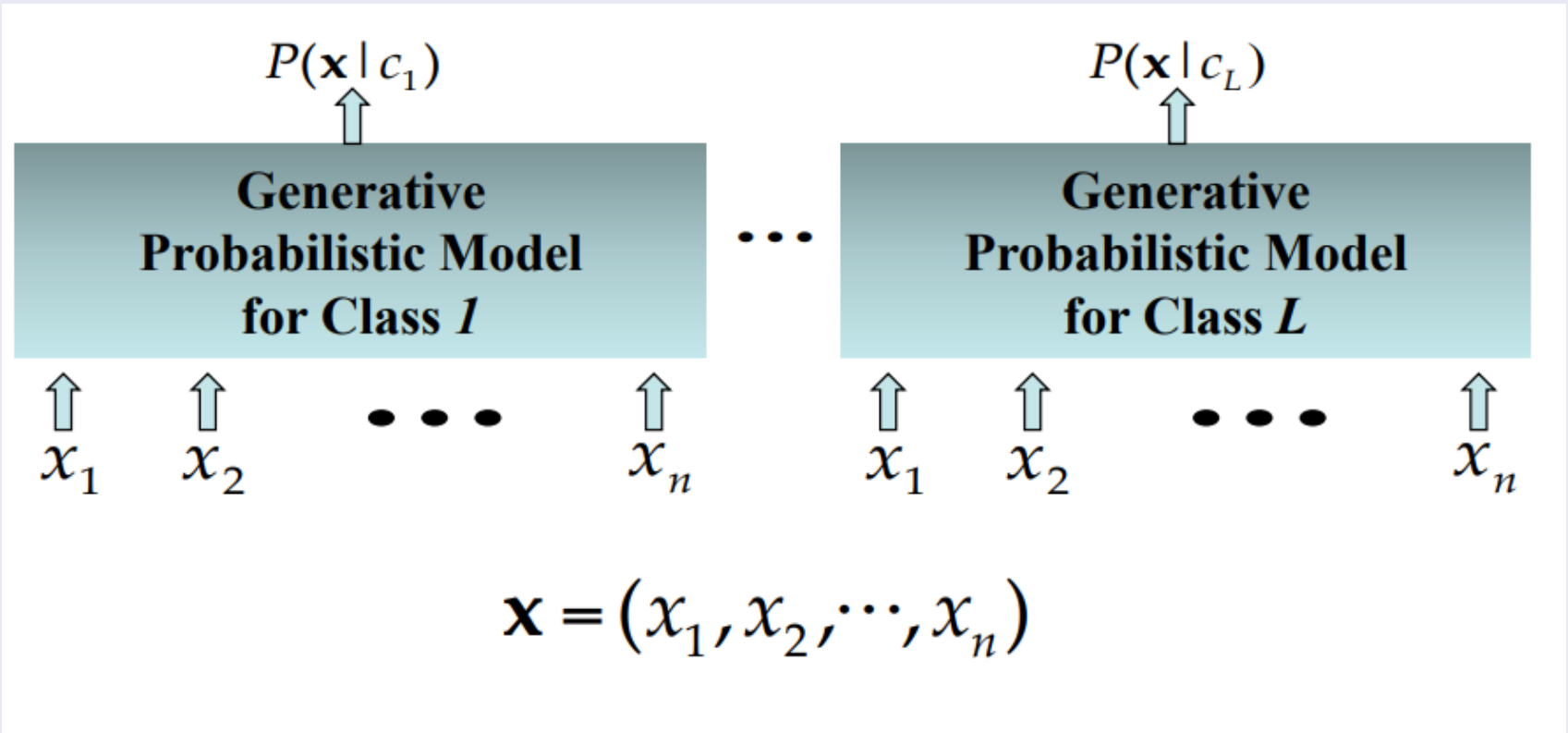
Discriminative classifier



Generative classifier

- $P(x|c)$, $c = c_1, \ldots, c_K$, $x = (x_1, \ldots, x_n)$
- K probabilistic models have to be trained **independently**
- Each is trained on only the examples of the same label
- Output K probabilities for a given input with K models
- "Generative" means that model can produce data via distribution sampling

Generative classifier



Maximum a-posteriori (MAP)

- For an input x, find the largest one from K probabilities output by a discriminative probabilistic classifier $P(c_1|x), \ldots, P(c_K|x)$
- Assign x to label c^* if $P(c^*|x)$ is the largest
- Generative classification with the MAP rule:

$$P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)} \propto P(x|c_i)P(c_i) \tag{1}$$

Naïve Bayes

Bayes classification

$$P(c|x) \propto P(x|c)P(c) = P(x_1, \ldots, x_n|c)P(c) \tag{2}$$

for $c = c_1, \ldots, c_K$

Naïve Bayes

Bayes classification

$$P(c|x) \propto P(x|c)P(c) = P(x_1, \ldots, x_n|c)P(c) \tag{2}$$

for $c = c_1, \ldots, c_K$

Problem

The joint probability $P(x_1, \ldots, x_n|c)$ is not feasible to learn.

Naïve Bayes

Bayes classification

$$P(c|x) \propto P(x|c)P(c) = P(x_1, \ldots, x_n|c)P(c) \tag{2}$$

for $c = c_1, \ldots, c_K$

Problem

The joint probability $P(x_1, \ldots, x_n|c)$ is not feasible to learn.

Solution

Assume all input features are class conditionally independent!

Bayes model

$$
\begin{aligned}
P(x_1, x_2, \ldots, x_n|c) &= P(x_1|x_2, \ldots, x_n, c)P(x_2, \ldots, x_n|c) \\
&= P(x_1|c)P(x_2, \ldots, x_n|c) \\
&= P(x_1|c)P(x_2|c)\ldots P(x_n|c)
\end{aligned} \tag{3}
$$

Algorithm

Discrete valued features

Learning phase: Given a training set S of F features and K classes,

- For each target value of $c_i (c_i = c_1, \ldots, c_K)$:
  - $\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in S
  - For every feature value x_{jk} of each feature $x_j (j = 1, \ldots, F; k = 1, \ldots N)$:
    - $\hat{P}(x_j = x_{jk}|c_i) \leftarrow$ estimate $P(x_{jk}|c_i)$ with samples in S

Output: $F \times K$ conditional probabilistic (generative) models.

Test phase: Given an unknown instance $x' = (a'_1, \ldots, a'_n)$ assign label c^* to x' if

$$[\hat{P}(a'_1|c^*) \ldots \hat{P}(a'_n|c^*)]\hat{P}(c^*) > [\hat{P}(a'_1|c_i) \ldots \hat{P}(a'_n|c_i)]\hat{P}(c_i) \qquad (4)$$

for $c_i \neq c^*, c_i = c_1, \ldots c_K$

Example

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Learning phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play=}Yes) = 9/14 \qquad P(\text{Play=}No) = 5/14$

Test phase

- Given a new instance, predict its label:
  $x' = (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong)$
- Look up tables:

P(Outlook=*Sunny* | Play=*Yes*) = 2/9	P(Outlook=*Sunny* | Play=*No*) = 3/5
P(Temperature=*Cool* | Play=*Yes*) = 3/9	P(Temperature=*Cool* | Play==*No*) = 1/5
P(Huminity=*High* | Play=*Yes*) = 3/9	P(Huminity=*High* | Play=*No*) = 4/5
P(Wind=*Strong* | Play=*Yes*) = 3/9	P(Wind=*Strong* | Play=*No*) = 3/5
P(Play=*Yes*) = 9/14	P(Play=*No*) = 5/14

- Make decision with the MAP rule:

$$P(Yes \mid \mathbf{x}') \approx [P(Sunny \mid Yes)P(Cool \mid Yes)P(High \mid Yes)P(Strong \mid Yes)]P(Play=Yes) = 0.0053$$
$$P(No \mid \mathbf{x}') \approx [P(Sunny \mid No)\ P(Cool \mid No)P(High \mid No)P(Strong \mid No)]P(Play=No) = 0.0206$$

Given the fact $P(Yes \mid \mathbf{x}') < P(No \mid \mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".