Bayes Classifier

Ravi Kothari, Ph.D.
ravi.kothari@ashoka.edu.in

"as far as laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality - Albert Einstein"

The Bayes Classifier (1/2)

The Bayes Classifier (1/2)

- Let L_{kj} be the loss when the classifier says an example comes from class j when it actually comes from class k

The Bayes Classifier (1/2)

- Let L_{kj} be the loss when the classifier says an example comes from class j when it actually comes from class k
- Average loss in assigning a pattern to class j is,

$$r_j(x) = \sum_{k=1}^{c} L_{kj} P(\omega_k|x)$$

The Bayes Classifier (1/2)

- Let L_{kj} be the loss when the classifier says an example comes from class j when it actually comes from class k
- Average loss in assigning a pattern to class j is,

$$r_j(x) = \sum_{k=1}^{c} L_{kj} P(\omega_k | x)$$

- Using Bayes rule and keeping relevant terms,

$$\begin{aligned} r_j(x) &= \sum_{k=1}^{c} L_{kj} \frac{P(\omega_k) P(x|\omega_k)}{P(x)} \\ &= \sum_{k=1}^{c} L_{kj} P(\omega_k) P(x|\omega_k) \end{aligned}$$

The Bayes Classifier (2/2)

The Bayes Classifier (2/2)

- Assign a pattern to class i if $r_i(x) < r_j(x)$

The Bayes Classifier (2/2)

- Assign a pattern to class i if $r_i(x) < r_j(x)$
- Say, $L_{kj} = 1 - \delta_{kj}$. Then,

$$
\begin{aligned}
r_j(x) &= \sum_{k=1}^{c} (1 - \delta_{kj}) P(\omega_k) P(x|\omega_k) \\
&= P(x) - P(x|\omega_j) P(\omega_j) \\
&= P(x) - d_j(x)
\end{aligned}
$$

The Bayes Classifier (2/2)

- Assign a pattern to class i if $r_i(x) < r_j(x)$
- Say, $L_{kj} = 1 - \delta_{kj}$. Then,

$$
\begin{aligned}
r_j(x) &= \sum_{k=1}^{c}(1-\delta_{kj})P(\omega_k)P(x|\omega_k) \\
&= P(x) - P(x|\omega_j)P(\omega_j) \\
&= P(x) - d_j(x)
\end{aligned}
$$

- Assign pattern to class with minimum risk, $r_j(x)$, or maximum $d_j(x) = P(x|\omega_j)P(\omega_j)$

The Bayes Classifier (2/2)

- Assign a pattern to class i if $r_i(x) < r_j(x)$
- Say, $L_{kj} = 1 - \delta_{kj}$. Then,

$$
\begin{aligned}
r_j(x) &= \sum_{k=1}^{c} (1 - \delta_{kj}) P(\omega_k) P(x|\omega_k) \\
&= P(x) - P(x|\omega_j) P(\omega_j) \\
&= P(x) - d_j(x)
\end{aligned}
$$

- Assign pattern to class with minimum risk, $r_j(x)$, or maximum $d_j(x) = P(x|\omega_j)P(\omega_j)$
- Optimum statistical classifier. Often (not always) used in parametrized form

Example # 1

Example # 1

- 2 classes; Gaussian PDF's characterized by (μ_1, σ_1) and (μ_2, σ_2) respectively

Example # 1

- 2 classes; Gaussian PDF's characterized by (μ_1, σ_1) and (μ_2, σ_2) respectively
- So,

$$d_j(x) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}} P(\omega_j)$$

Example # 1

- 2 classes; Gaussian PDF's characterized by (μ_1, σ_1) and (μ_2, σ_2) respectively
- So,

$$d_j(x) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}} P(\omega_j)$$

- On the decision boundary $d_i(x) = d_j(x)$

Example # 1

- 2 classes; Gaussian PDF's characterized by (μ_1, σ_1) and (μ_2, σ_2) respectively
- So,

$$d_j(x) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}} P(\omega_j)$$

- On the decision boundary $d_i(x) = d_j(x)$
- If $P(\omega_1) = P(\omega_2)$, the decision boundary is at x_0. If $P(\omega_1) > P(\omega_2)$, x_0 moves to the right. If $P(\omega_1) < P(\omega_2)$, x_0 moves to the left

Example # 2

[1] From R. C. Gonzalez, R. C. Woods, *Image Processing*, Prentice Hall

Example # 2

- An important event is scheduled for Saturday. On Thursday, a 60% chance of rain is predicted for Saturday. Should the social activities director postpone the event?[1]

[1] From R. C. Gonzalez, R. C. Woods, *Image Processing*, Prentice Hall

Example # 2

- An important event is scheduled for Saturday. On Thursday, a 60% chance of rain is predicted for Saturday. Should the social activities director postpone the event?[1]
- Denote "fair weather on Saturday as 1" and "rain on Saturday as 2". Let $L_{11} = -1, L_{12} = 3, L_{21} = 2, L22 = 0$

[1] From R. C. Gonzalez, R. C. Woods, *Image Processing*, Prentice Hall

Example # 2

- An important event is scheduled for Saturday. On Thursday, a 60% chance of rain is predicted for Saturday. Should the social activities director postpone the event?[1]
- Denote "fair weather on Saturday as 1" and "rain on Saturday as 2". Let $L_{11} = -1, L_{12} = 3, L_{21} = 2, L22 = 0$
- Recall that, $r_j(x) = \sum_{k=1}^{c} L_{kj} P(\omega_k|x)$

[1] From R. C. Gonzalez, R. C. Woods, *Image Processing*, Prentice Hall

Example # 2

- An important event is scheduled for Saturday. On Thursday, a 60% chance of rain is predicted for Saturday. Should the social activities director postpone the event?[1]
- Denote "fair weather on Saturday as 1" and "rain on Saturday as 2". Let $L_{11} = -1, L_{12} = 3, L_{21} = 2, L22 = 0$
- Recall that, $r_j(x) = \sum_{k=1}^{c} L_{kj} P(\omega_k|x)$

$$
\begin{aligned}
r_1(x) &= L_{11} P(\omega_1|x) + L_{21} P(\omega_2|x) \\
&= (-1)(0.4) + (2)(0.6) = 0.8 \\
r_2(x) &= L_{12} P(\omega_1|x) + L_{22} P(\omega_2|x) \\
&= (3)(0.4) + (0)(0.6) = 1.2
\end{aligned}
$$

[1] From R. C. Gonzalez, R. C. Woods, *Image Processing*, Prentice Hall

Example # 2

- An important event is scheduled for Saturday. On Thursday, a 60% chance of rain is predicted for Saturday. Should the social activities director postpone the event?[1]
- Denote "fair weather on Saturday as 1" and "rain on Saturday as 2". Let $L_{11} = -1, L_{12} = 3, L_{21} = 2, L22 = 0$
- Recall that, $r_j(x) = \sum_{k=1}^{c} L_{kj} P(\omega_k|x)$

$$
\begin{aligned}
r_1(x) &= L_{11} P(\omega_1|x) + L_{21} P(\omega_2|x) \\
&= (-1)(0.4) + (2)(0.6) = 0.8 \\
r_2(x) &= L_{12} P(\omega_1|x) + L_{22} P(\omega_2|x) \\
&= (3)(0.4) + (0)(0.6) = 1.2
\end{aligned}
$$

- So, go ahead with the program

[1] From R. C. Gonzalez, R. C. Woods, *Image Processing*, Prentice Hall

Example # 3

Example # 3

- Consider a highly simplified scenario. Let there be an conveyor belt carrying pieces of a type of fruit e.g. apple. There is a camera mounted above conveyor belt that detects rotten apples and nudges an arm to remove them from the conveyor belt. The good apples are then packed into boxes downstream

Example # 3

- Consider a highly simplified scenario. Let there be an conveyor belt carrying pieces of a type of fruit e.g. apple. There is a camera mounted above conveyor belt that detects rotten apples and nudges an arm to remove them from the conveyor belt. The good apples are then packed into boxes downstream
- Let us not worry about the intricacies of image processing including illumination etc. There is a library available which we invoke and it gives us back 2 features for each piece of fruit related to: (i) color homogenity (c), and (ii) shape conformity (s). Each feature for simplicity is normalized to line in $[0.0, 1.0]$

Example # 3

- Consider a highly simplified scenario. Let there be an conveyor belt carrying pieces of a type of fruit e.g. apple. There is a camera mounted above conveyor belt that detects rotten apples and nudges an arm to remove them from the conveyor belt. The good apples are then packed into boxes downstream
- Let us not worry about the intricacies of image processing including illumination etc. There is a library available which we invoke and it gives us back 2 features for each piece of fruit related to: (i) color homogenity (c), and (ii) shape conformity (s). Each feature for simplicity is normalized to line in $[0.0, 1.0]$
- We have many examples (each example is described in terms of the 2 features above) of a defective fruit and many examples of a good piece of fruit

Example # 3

- Consider a highly simplified scenario. Let there be an conveyor belt carrying pieces of a type of fruit e.g. apple. There is a camera mounted above conveyor belt that detects rotten apples and nudges an arm to remove them from the conveyor belt. The good apples are then packed into boxes downstream
- Let us not worry about the intricacies of image processing including illumination etc. There is a library available which we invoke and it gives us back 2 features for each piece of fruit related to: (i) color homogenity (c), and (ii) shape conformity (s). Each feature for simplicity is normalized to line in $[0.0, 1.0]$
- We have many examples (each example is described in terms of the 2 features above) of a defective fruit and many examples of a good piece of fruit
- How do we construct a Bayes classifier for this problem?

Example # 3

Example # 3

- Let us discretize "c" into 4 categories and "s" into 3 categories

Good apples (left):

	Color			
Shape	21	6	18	7
	12	16	3	22
	3	4	19	21

Bad apples (right):

	Color			
Shape	1	16	28	17
	2	6	12	2
	33	44	4	7

Figure: Good apples (left), Bad apples (right)

Example # 3

- Let us discretize "c" into 4 categories and "s" into 3 categories

Color (columns), Shape (rows)

21	6	18	7
12	16	3	22
3	4	19	21

Color (columns), Shape (rows)

1	16	28	17
2	6	12	2
33	44	4	7

Figure: Good apples (left), Bad apples (right)

- Given a new apple whose "c" is in Category 3 and "s" is in Category 1,

$$
\begin{aligned}
P(Good|C=3,S=1) &= P(C=3,S=1|Good)P(Good) \\
&= \frac{18}{152}\,\frac{152}{324} = 0.055556 \\
P(Bad|C=3,S=1) &= P(C=3,S=1|Bad)P(Bad) \\
&= \frac{28}{172}\,\frac{172}{324} = 0.08642
\end{aligned}
$$

Example # 3

- Let us discretize "c" into 4 categories and "s" into 3 categories

	Color			
Shape	21	6	18	7
	12	16	3	22
	3	4	19	21

	Color			
Shape	1	16	28	17
	2	6	12	2
	33	44	4	7

Figure: Good apples (left), Bad apples (right)

- Given a new apple whose "c" is in Category 3 and "s" is in Category 1,

$$
\begin{aligned}
P(Good|C=3,S=1) &= P(C=3,S=1|Good)P(Good) \\
&= \frac{18}{152}\frac{152}{324} = 0.055556 \\
P(Bad|C=3,S=1) &= P(C=3,S=1|Bad)P(Bad) \\
&= \frac{28}{172}\frac{172}{324} = 0.08642
\end{aligned}
$$

- So, the apple is...?

Some Comments

Some Comments

- This was an overly simplified scenario. What happens when the number of features runs into hundreds or thousands or more?

Some Comments

- This was an overly simplified scenario. What happens when the number of features runs into hundreds or thousands or more?
- *Curse of dimensionality*

Naive Bayes

Naive Bayes

- *Naive Bayes* is well, Naive. To compensate for limited data, it makes the simplifying assumption that features are independent of each other

Naive Bayes

- *Naive Bayes* is well, Naive. To compensate for limited data, it makes the simplifying assumption that features are independent of each other
- Thus,

$$P(x) = P(x_1)\, P(x_2)\, \ldots, P(x_n)$$

or

$$P(x|\omega_k) = \prod_{i=1}^{n} P(x_i|\omega_k)$$

Naive Bayes

- *Naive Bayes* is well, Naive. To compensate for limited data, it makes the simplifying assumption that features are independent of each other
- Thus,

$$P(x) = P(x_1)\,P(x_2)\,\ldots,P(x_n)$$

or

$$P(x|\omega_k) = \prod_{i=1}^{n} P(x_i|\omega_k)$$

- The assumption is of course, unfounded. However, it has been quite successful in the NLP domain

Naive Bayes

- *Naive Bayes* is well, Naive. To compensate for limited data, it makes the simplifying assumption that features are independent of each other
- Thus,

$$P(x) = P(x_1)\, P(x_2)\ \ldots, P(x_n)$$

or

$$P(x|\omega_k) = \prod_{i=1}^{n} P(x_i|\omega_k)$$

- The assumption is of course, unfounded. However, it has been quite successful in the NLP domain
- How would you use Naive Bayes to classify emails in to “spam” and “normal” cateogories?