Generative Learning Algorithms

Carlos Valle

Departamento de Informática
Universidad Técnica Federico Santa María

cvalle@inf.utfsm.cl

December 3, 2015

Overview

Generative models

- In our credit classification problem we could learn a model for good credits and learn a separate model for bad credits.
- Then, to classify a new credit, we could see which model is the closest to it.
- These approaches try to model $p(y|x)$ by using $p(x|y)$ and $p(y)$ the margin distributions or class priors.
- Note that the strong assumption of these algorithms is that we can directly model $p(x|y)$ and $p(y)$ from train data.
- Actually for supervised tasks discriminative models tend to be more efficient in practice.

Generative models (2)

- We can predict $p(y|x)$ by applying Bayes theorem:
- $$\frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\int_y p(x|y)p(y)dy}$$
- Note that $p(x|y)p(y) = p(x,y)$ is the joint density of the data.
- In binary classification the denominator is given by $p(x) = p(x|y=1)p(y=1) + p(x|y=-1)p(y=-1)$.
- However, we don't need $p(x)$ for maximizing $p(y|x)$:

$$\begin{aligned} \text{argmax}_y p(y|x) &= \text{argmax}_y \frac{p(x|y)p(y)}{p(x)} \\ &= \text{argmax}_y p(x|y)p(y) \end{aligned}$$

Linear discriminant analysis (LDA)

- Suppose that the density function of each class k is a multivariate Gaussian

$$f_k(x) = P(x|y=k) = \frac{1}{(2\pi)^{I/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)},$$

- where $\mu \in \mathbb{R}$ is the mean vector and a covariance matrix $\Sigma \in \mathbb{R}^I \times \mathbb{R}^I$, where $\Sigma \geq 0$ is symmetric and positive semi-definite.
- $|\Sigma|$ is the determinant of the matrix Σ
- $E[X] = \mu$,
  $\text{Cov}(Z) = E[(Z - E[Z])(Z - E[Z])^T] = E[ZZ^T] - E[Z]E[Z]^T$
- Suppose that the covariance matrices are the same for all classes, this is $\Sigma_k = \Sigma$, $\forall k$.

Linear discriminant analysis (LDA) (2)

- Applying Bayes theorem we have

$$P(y=k|X=x) = \frac{f_k(x)p_k}{\sum_{k=1}^{K} f_k(x)p_k}, \; k=1,\ldots,K$$

- where $p_k = P(y=k)$
- To compare two classes we use the log-ratio:

$$\begin{aligned}
\log \frac{P(y=k_1|X=x)}{P(y=k_2|X=x)} &= \log \frac{f_{k_1}(x)}{f_{k_2}(x)} + \log \frac{p_{k_1}}{p_{k_2}} \\
&= \log \frac{p_{k_1}}{p_{k_2}} + -\frac{1}{2}(\mu_{k_1}+\mu_{k_2})^T \Sigma^{-1}(\mu_{k_1}-\mu_{k_2}) \\
&+ x^T \Sigma^{-1}(\mu_{k_1}-\mu_{k_2}) \qquad (1)
\end{aligned}$$

- This is linear in x

Linear discriminant analysis (LDA) (3)

- The decision rule is:

$$
\begin{aligned}
\hat{y} &= \text{argmax}_k Pr(y = k | X = x) \\
&= \text{argmax}_k f_k(x) p_k = \text{argmax}_k \log(f_k(x) p_k) \\
&= \text{argmax}_k \left[\log p_k - \log((2p)^{I/2} |\Sigma|^{1/2}) - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \right] \\
&= \text{argmax}_k \left[\log p_k - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \right]
\end{aligned}
$$

Note that

$$
-\frac{1}{2}(x - \mu_k)^T \Sigma^{-1} (x - \mu_k) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k - \frac{1}{2} x^T \Sigma^{-1} x
$$

Then,

$$
\hat{y} = \text{argmax}_k \left[\log p_k + x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \right]
$$

LDA (4)

- Hence, the discriminant function is given by:
- $\delta_k(x) = \log p_k + x^T \Sigma^{-1} \mu_k^T - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k$
- And the boundary decision between the classes k_1 and k_2 is:

$$\delta_{k_1}(x) = \delta_{k_2}(x)$$

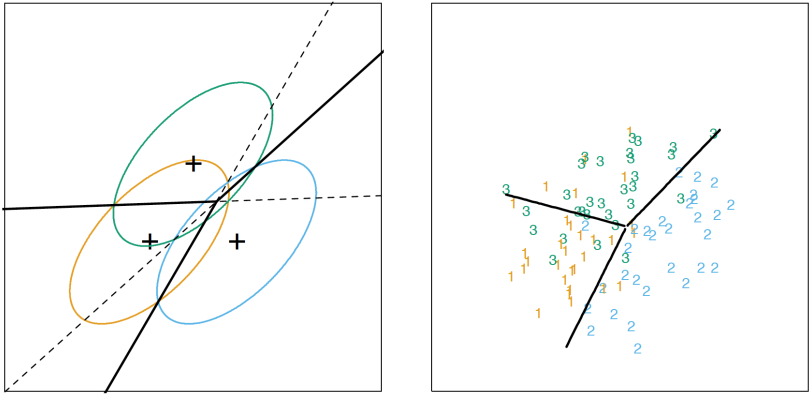
- We estimate p_k, μ_k and Σ using the training set:

$$\begin{aligned} \hat{p}_k &= M_k/M, \ k = 1, \ldots, K \\ \hat{\mu}_k &= \sum_{y_m = k} \mathbf{x}_m / M_k, \ k = 1, \ldots, K \\ \hat{\Sigma} &= \sum_{k=1}^{K} \sum_{y_m = k} (x_m - \hat{\mu}_k)(x_m - \hat{\mu}_k)^T / (M - K) \end{aligned} \tag{2}$$

- In a binary classification problem, we choose the class 2 if

$$x^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2} \hat{\mu}_2^T \hat{\Sigma}^{-1} \hat{\mu}_2 - \frac{1}{2} \hat{\mu}_1^T \hat{\Sigma}^{-1} \hat{\mu}_1 + \log(p_1) - \log(p_2)$$

What LDA is doing?



binary LDA

- We have $k = 2$, $y \in \{0, 1\}$
- $y \sim$ Bernoulli(p)
- $x|y = 0 \sim N(\mu_0, \Sigma)$
- $x|y = 1 \sim N(\mu_1, \Sigma)$
- Then, the probability distributions of y, $x|y = 0$, and $x|y = 1$ are given by
- $p(y) = p^y(1 - p)^{1-y}$,
- $p(x|y = 0) = \frac{1}{(2\pi)^{I/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)}$,
- $p(x|y = 1) = \frac{1}{(2\pi)^{I/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)}$,

binary LDA (2)

- It can be shown that p, μ_0, μ_1, σ that maximizes the log-likelihood

$$
\begin{aligned}
\ell(p, \mu_0, \mu_1, \sigma) &= \log \prod_{m=1}^{M} p(\mathbf{x}_m, y_m; p, \mu_0, \mu_1, \Sigma) \\
&= \log \prod_{m=1}^{M} p(\mathbf{x}_m | y_m; \mu_0, \mu_1, \Sigma) p(y_m; p)
\end{aligned}
$$

are the same that we used in (2)

What if the covariance matrices are different?: QDA

- If we not assume the covariance matrices to be equal $\Sigma_k, k = 1, \ldots, K$, we can't simplify them from equation (1), therefore
- The discriminant function for the k-th class is computed as $\delta_k(x) = \log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k) + \log p_k$
- This is called Quadratic discriminant functions o QDA
- The boundary decision between the classes k_1 and k_2 is:

$$\delta_{k_1}(x) = \delta_{k_2}(x)$$

- The decision boundaries are approximated by quadratic equations in x. In general, this method outperforms LDA, but the number of parameters to estimate is larger.

Logistic regression vs LDA

- Logistic regression directly model $E[y|x]$ to define the decision boundary without making any assumption about $p(x|y)$.
- While LDA assumes that $p(x|y)$ follows a multivariate Gaussian. Moreover, the covariance matrix of each class is the same.
- If these assumptions are not met the performance of LDA will severely drop.
- In practice these assumptions are never correct, and often some of the components of X are qualitative variables.
- It is generally felt that logistic regression is a more robust than LDA, relying on fewer assumptions.
- Note that observations far from the decision boundary (which are down-weighted by logistic regression) play a role in estimating the shared covariance matrix.
- Thererfore, LDA is not robust to gross outliers.

Naive Bayes

- We can model $p(\mathbf{x}|y) = p(x^{(1)}, x^{(2)}, \ldots, x^{(I)}|y)$ as

$$p(x^{(1)}, x^{(2)}, \ldots, x^{(I)}|y) = p(x^{(1)}|y)p(x^{(2)}|y, x^{(1)})p(x^{(3)}|y, x^{(1)}, x^{(2)}) \cdots p(x^{(I)}|y, x^{(1)}, \ldots, x^{(I-1)})$$

- However, we can assume that the input features $x^{(i)}$ are conditionally independent given y. Hence,

$$\begin{aligned} p(x^{(1)}, x^{(2)}, \ldots, x^{(I)}|y) &= p(x^{(1)}|y)p(x^{(2)}|y)p(x^{(3)}|y) \cdots p(x^{(I)}|y) \\ &= \prod_{i=1}^{I} p(x^{(i)}|y) \end{aligned}$$

- Again, our prediction $\hat{y}$ will be:

$$\begin{aligned} \text{argmax}_k p(y = k|\mathbf{x}) &= \text{argmax}_k p(\mathbf{x}|y = k)p(y = k) \\ &= \text{argmax}_k \prod_{i=1}^{I} p(x^{(i)}|y = k)p_k \end{aligned}$$

Naive Bayes: How to estimate the probabilities?

- Given a training set $S = \{x_m, y_m\}_{m=1}^{M}$, we can compute the likelihood of S

$$
\begin{aligned}
L(\mathbf{p}, p(x|y=1), \ldots, p(x|y=k)) &= \prod_{m=1}^{M} p(x_m, y_m) \\
&= \prod_{m=1}^{M} \prod_{i=1}^{I} p(x_m^{(i)}|y_m = k) p_k
\end{aligned}
$$

- Maximizing w.r.t. $\mathbf{p}, p(x|y=1), \ldots, p(x|y=k)$ we obtain

$$\hat{p}_k = \sum_{m=1}^{M} I(y_m = k)/M,\ k = 1, \ldots, K,$$

$$\hat{p}(x^{(i)}|y=k) = \sum_{m=1}^{M} I((y_m = k) \wedge (x_m^{(i)} = x^{(i)}))/M_k,\ k = 1, \ldots, K.$$

Naive Bayes: How to estimates the probabilities? (2)

- As we saw, to estimate $\hat{p}(x^{(i)}|y=k)$ we need that $x^{(i)}$ takes a finite number of values.
- When $x^{(i)}$ is continuous, we need to discrete it.
- What if a new example $\mathbf{x}$ has a feature $x^{(i)}$ different from the values found in the training set?
- $p(\mathbf{x}) = 0$. Thus, $p(y=k|\mathbf{x}) = 0/0, \forall k$
- To correct it, we use Laplace smoothing:
- Let q the number of possible values that $x^{(i)}$ can take.
$\hat{p}(x^{(i)}|y=k) = \sum_{m=1}^{M}[I((y_m = k) \wedge (x_m^{(i)} = x^{(i)})) + 1]/(M_k + q)$, $k = 1, \ldots, K$.

Any questions?