E0 270 Machine Learning **Lecture 2 (Jan 10, 2013)**

Generative Probabilistic Models for Classification

Lecturer: Shivani Agarwal

Disclaimer: These notes are a *brief* summary of the topics covered in the lecture. They are not a substitute for the full lecture.

Outline

- Multivariate normal class-conditional densities
  - Linear discriminant analysis (LDA)
- Conditionally independent features
  - Naïve Bayes
- Extensions to multiclass classification

1 Multivariate Normal Class-Conditional Densities

Let $\mathcal{X} = \mathbb{R}^d$, and consider a binary classification task with label space $\mathcal{Y} = \{\pm 1\}$. Assume that for each class $y \in \mathcal{Y}$, the **class-conditional density** of $\mathbf{x}$ given y, which we denote by $f_y(\mathbf{x})$, is a multivariate normal density:

$$f_y(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_y|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^\top \mathbf{\Sigma}_y^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)\right) ,$$

where $\boldsymbol{\mu}_y \in \mathbb{R}^d$ and $\mathbf{\Sigma}_y \in \mathbb{R}^{d\times d}$ are the unknown class-conditional mean and covariance matrix, respectively. Also, let $p = \mathbf{P}(y=1)$. Then clearly, the conditional probability of a positive class label for any $\mathbf{x} \in \mathcal{X}$ can be obtained using Bayes' rule:

$$\eta(\mathbf{x}) = \mathbf{P}(y=1|\mathbf{x}) = \frac{p\, f_1(\mathbf{x})}{p\, f_1(\mathbf{x}) + (1-p)\, f_{-1}(\mathbf{x})} ,$$

leading to the following Bayes optimal classifier:

$$\begin{aligned} h^*(\mathbf{x}) &= \begin{cases} +1 & \text{if } \frac{p\, f_1(\mathbf{x})}{p\, f_1(\mathbf{x})+(1-p)\, f_{-1}(\mathbf{x})} > \frac{1}{2} \\ -1 & \text{otherwise} \end{cases} \\ &= \begin{cases} +1 & \text{if } \frac{f_1(\mathbf{x})}{f_{-1}(\mathbf{x})} > \frac{1-p}{p} \\ -1 & \text{otherwise} \end{cases} \\ &= \begin{cases} +1 & \text{if } \ln\left(\frac{f_1(\mathbf{x})}{f_{-1}(\mathbf{x})}\right) > \ln\left(\frac{1-p}{p}\right) \\ -1 & \text{otherwise} \end{cases} \\ &= \text{sign}\Big(\ln\Big(\frac{f_1(\mathbf{x})}{f_{-1}(\mathbf{x})}\Big) - \ln\Big(\frac{1-p}{p}\Big)\Big) . \end{aligned}$$

Now, under the above assumption on the class-conditional densities, we have

$$\ln\Big(\frac{f_1(\mathbf{x})}{f_{-1}(\mathbf{x})}\Big) = \frac{1}{2}\Big(\mathbf{x}^\top(\mathbf{\Sigma}_{-1}^{-1} - \mathbf{\Sigma}_1^{-1})\mathbf{x} - 2(\mathbf{\Sigma}_{-1}^{-1}\boldsymbol{\mu}_{-1} - \mathbf{\Sigma}_1^{-1}\boldsymbol{\mu}_1)^\top\mathbf{x} + \boldsymbol{\mu}_{-1}^\top\mathbf{\Sigma}_{-1}^{-1}\boldsymbol{\mu}_{-1} - \boldsymbol{\mu}_1^\top\mathbf{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \ln\Big(\frac{|\mathbf{\Sigma}_{-1}|}{|\mathbf{\Sigma}_1|}\Big)\Big) .$$

This is a quadratic function of $\mathbf{x}$, and we can therefore write the Bayes optimal classifier in this case as

$$h^*(\mathbf{x}) = \operatorname{sign}\big(\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c\big)\,,$$

where

$$\begin{aligned}
\mathbf{A} &= \boldsymbol{\Sigma}_{-1}^{-1} - \boldsymbol{\Sigma}_1^{-1} \\
\mathbf{b} &= -2(\boldsymbol{\Sigma}_{-1}^{-1}\boldsymbol{\mu}_{-1} - \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1) \\
c &= \boldsymbol{\mu}_{-1}^\top \boldsymbol{\Sigma}_{-1}^{-1}\boldsymbol{\mu}_{-1} - \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \ln\Big(\frac{|\boldsymbol{\Sigma}_{-1}|}{|\boldsymbol{\Sigma}_1|}\Big) - 2\ln\Big(\frac{1-p}{p}\Big)\,.
\end{aligned}$$

A classifier of this form is often called a **degree-2 polynomial threshold classifier**, or simply a **quadratic classifier**. Note that if in addition, the class-conditional densities have equal covariance matrices, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_{-1} = \boldsymbol{\Sigma}$, then $\mathbf{A} = \mathbf{0}$, and the Bayes optimal classifier becomes a **linear threshold classifier**, or simply a **linear classifier**.

Of course, in practice, one does not know the parameters of the class-conditional densities, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_{-1}$, or the class probability parameter $p = \mathbf{P}(y = 1)$. In this case, one estimates these quantities from the given training sample $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$, and uses the estimated values $\widehat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\mu}}_{-1}, \widehat{\boldsymbol{\Sigma}}_1, \widehat{\boldsymbol{\Sigma}}_{-1}, \widehat{p}$ to obtain a **plug-in conditional class probability estimate** $\widehat{\eta}_S(\mathbf{x})$, and a corresponding **plug-in classifier**

$$h_S(\mathbf{x}) = \operatorname{sign}\big(\widehat{\eta}_S(\mathbf{x}) - \tfrac{1}{2}\big)\,.$$

For example, a natural approach is to use **maximum likelihood estimation**, which yields

$$\begin{aligned}
\widehat{\boldsymbol{\mu}}_y &= \frac{1}{m_y}\sum_{i:y_i=y} \mathbf{x}_i \\
\widehat{\boldsymbol{\Sigma}}_y &= \frac{1}{m_y}\sum_{i:y_i=y} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_y)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_y)^\top \\
\widehat{p} &= \frac{m_1}{m}\,,
\end{aligned}$$

where $m_y = \big|\{i \in [m] : y_i = y\}\big|$ denotes the number of training examples with class label y. If one assumes the class-conditional covariances are equal, then the maximum likelihood estimate for the common covariance matrix is given by

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{m}\sum_{i=1}^{m} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{y_i})(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{y_i})^\top\,,$$

and the resulting classifier is given by

$$h_S(\mathbf{x}) = \operatorname{sign}\big(\widehat{\mathbf{b}}^\top \mathbf{x} + \widehat{c}\big)\,,$$

where

$$\begin{aligned}
\widehat{\mathbf{b}} &= -2\,\widehat{\boldsymbol{\Sigma}}^{-1}(\widehat{\boldsymbol{\mu}}_{-1} - \widehat{\boldsymbol{\mu}}_1) \\
\widehat{c} &= \widehat{\boldsymbol{\mu}}_{-1}^\top \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_{-1} - \widehat{\boldsymbol{\mu}}_1^\top \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_1 - 2\ln\Big(\frac{1-\widehat{p}}{\widehat{p}}\Big)\,.
\end{aligned}$$

This classifier is called the **linear discriminant analysis (LDA)** classifier.

2 Conditionally Independent Features

Suppose now for simplicity that $\mathcal{X} = \{0,1\}^d$ (binary features), and consider again a binary classification task with label space $\mathcal{Y} = \{\pm 1\}$. The class-conditional distributions are now discrete; let us denote these as $p_1(\mathbf{x})$ and $p_{-1}(\mathbf{x})$. Clearly, in the general case, each of these distributions, defined over the sample space $\mathcal{X} = \{0,1\}^d$ containing 2^d elements, is parametrized by $2^d - 1$ numbers, namely the probabilities of seeing the

different elements in $\mathcal{X}$. However these $2^d - 1$ parameters can be estimated reliably only when all instances in $\mathcal{X}$ have been seen several times, which is unrealistic in a typical learning situation. Consequently, one usually makes some assumptions on the form of these class-conditional distributions that allows them to be represented more compactly.

A common assumption that is made is that given the class label y, the individual features in $\mathbf{x}$ are conditionally independent; i.e. that each class-conditional probability distribution factors as follows:

$$p_y(\mathbf{x}) = \mathbf{P}(\mathbf{x}|y) = \prod_{k=1}^{d} \mathbf{P}(x_k|y)\,.$$

In this case, one needs to estimate only d parameters for each class-conditional distribution (why is this not $d-1$?). For each $y \in \mathcal{Y}$ and $k \in [d]$, denote

$$\theta_{y,k} = \mathbf{P}(x_k = 1|y)\,.$$

Then we can write

$$p_y(\mathbf{x}) = \prod_{k=1}^{d} (\theta_{y,k})^{x_k} (1-\theta_{y,k})^{1-x_k}\,.$$

The conditional probability of a positive label for any $\mathbf{x} \in \mathcal{X}$ is again obtained via Bayes' rule:

$$\eta(\mathbf{x}) = \mathbf{P}(y=1|\mathbf{x}) = \frac{p \cdot p_1(\mathbf{x})}{p \cdot p_1(\mathbf{x}) + (1-p) \cdot p_{-1}(\mathbf{x})}\,,$$

where again $p = \mathbf{P}(y=1)$, leading again to the following Bayes optimal classifier:

$$h^*(\mathbf{x}) = \operatorname{sign}\Big(\ln\Big(\frac{p_1(\mathbf{x})}{p_{-1}(\mathbf{x})}\Big) - \ln\Big(\frac{1-p}{p}\Big)\Big)\,.$$

In this case, we have

$$\ln\Big(\frac{p_1(\mathbf{x})}{p_{-1}(\mathbf{x})}\Big) \;=\; \sum_{k=1}^{d} \Big(x_k \ln\Big(\frac{\theta_{1,k}}{\theta_{-1,k}}\Big) + (1-x_k)\ln\Big(\frac{1-\theta_{1,k}}{1-\theta_{-1,k}}\Big)\Big)\,.$$

This is a linear function of $\mathbf{x}$, and we can therefore write the Bayes optimal classifier in this case as

$$h^*(\mathbf{x}) = \operatorname{sign}\big(\mathbf{w}^\top \mathbf{x} + b\big)\,,$$

where

$$\begin{aligned} w_k &= \ln\Big(\frac{\theta_{1,k}}{\theta_{-1,k}}\Big) - \ln\Big(\frac{1-\theta_{1,k}}{1-\theta_{-1,k}}\Big) \\ b &= \Big(\sum_{k=1}^{d} \ln\Big(\frac{1-\theta_{1,k}}{1-\theta_{-1,k}}\Big)\Big) - \ln\Big(\frac{1-p}{p}\Big)\,. \end{aligned}$$

As can be seen, this again yields a linear classifier. Again, in practice, one estimates the parameters $\theta_{y,k}$ and p from the given training data $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ using maximum likelihood estimation, which yields

$$\begin{aligned} \widehat{\theta}_{y,k} &= \frac{1}{m_y} \sum_{i:y_i=y} \mathbf{1}(x_{ik} = 1) \\ \widehat{p} &= \frac{m_1}{m}\,, \end{aligned}$$

where $m_y = \big|\{i \in [m] : y_i = y\}\big|$ as before. The resulting plug-in classifier, obtained by substituting these parameter estimates in the expression for the Bayes optimal classifier above, is known as the **naïve Bayes** classifier.

Exercise. How does the above derivation change if you have q-ary features, $\mathcal{X} = \{0, \ldots, q-1\}^d$? How many parameters do you now need to estimate for each class? Do you still get a linear classifier?

3 Extensions to Multiclass Classification

Let us see how things change when there are $r > 2$ classes, say $\mathcal{Y} = [r] = \{1, \ldots, r\}$ (such as in the handwritten digit recognition example, where $r = 10$). In this case, we need to consider the conditional probability of different labels given an instance $x \in \mathcal{X}$. For each $y \in \mathcal{Y}$, let $\eta_y(x) = \mathbf{P}(y|x)$ denote the conditional probability of seeing label y given x. Clearly, for all x, $\sum_{y=1}^{r} \eta_y(x) = 1$ (in the binary case, we had $\eta_1(x) = \eta(x)$ and $\eta_{-1}(x) = 1 - \eta(x)$). What does the optimal classifier look like in this case? This depends on how we measure the performance of a classifier $h : \mathcal{X} \to \mathcal{Y}$. Denoting again the marginal distribution over $\mathcal{X}$ by μ and the resulting joint distribution over $\mathcal{X} \times \mathcal{Y}$ by D, say we define again the accuracy of h w.r.t. D as the probability that an example (x, y) drawn randomly from D is classified correctly by h:

$$\mathrm{acc}_D[h] = \mathbf{P}_{(x,y)\sim D}\big(h(x) = y\big)\,.$$

Equivalently, we use again the 0-1 loss function, defined now over labels and predictions in $\mathcal{Y} = [r]$, giving $\ell_{0\text{-}1} : [r] \times [r] \to \{0, 1\}$ defined as

$$\ell_{0\text{-}1}(y, \widehat{y}) = \mathbf{1}(\widehat{y} \neq y)\,,$$

with the corresponding 0-1 error of h w.r.t. D defined as

$$\mathrm{er}_D^{0\text{-}1}[h] = \mathbf{P}_{(x,y)\sim D}\big(h(x) \neq y\big) = \mathbf{E}_{(x,y)\sim D}\big[\ell_{0\text{-}1}(y, h(x))\big]\,.$$

We can write this as

$$\begin{aligned}
\mathrm{er}_D^{0\text{-}1}[h] &= \mathbf{E}_{(x,y)\sim D}\big[\mathbf{1}(h(x) \neq y)\big] \\
&= \mathbf{E}_{x\sim\mu}\big[\mathbf{E}_{y|x}\big[\mathbf{1}(h(x) \neq y)\big]\big] \\
&= \mathbf{E}_{x\sim\mu}\Big[\sum_{y=1}^{r} \eta_y(x) \cdot \mathbf{1}(h(x) \neq y)\Big] \\
&= \mathbf{E}_{x\sim\mu}\Big[\sum_{y\neq h(x)} \eta_y(x)\Big] \\
&= \mathbf{E}_{x\sim\mu}\big[1 - \eta_{h(x)}(x)\big]\,.
\end{aligned}$$

The minimum achievable 0-1 error w.r.t. D is therefore

$$\mathrm{er}_D^{0\text{-}1,*} \;=\; \inf_{h:\mathcal{X}\to[r]} \mathrm{er}_D^{0\text{-}1}[h] \;=\; 1 - \mathbf{E}_{x\sim\mu}\Big[\max_{y\in[r]} \eta_y(x)\Big]\,,$$

and is clearly achieved by any classifier $h^* : \mathcal{X} \to [r]$ satisfying[1]

$$h^*(x) \in \arg\max_{y\in[r]} \eta_y(x)\,.$$

Now, let $\mathcal{X} = \mathbb{R}^d$ and assume again that for each class $y \in \mathcal{Y}$, the class-conditional density $f_y(\mathbf{x})$ is a multivariate normal density with mean vector $\boldsymbol{\mu}_y$ and covariance matrix $\boldsymbol{\Sigma}_y$. Also, for each $y \in \mathcal{Y}$, let $p_y = \mathbf{P}(y)$ denote the overall probability of seeing label y. Then the conditional probability of seeing label y for any $\mathbf{x} \in \mathcal{X}$ can again be obtained by Bayes' rule:

$$\eta_y(\mathbf{x}) = \mathbf{P}(y|\mathbf{x}) = \frac{p_y\, f_y(\mathbf{x})}{\sum_{y'=1}^{r} p_{y'}\, f_{y'}(\mathbf{x})}\,,$$

[1] Note that if our loss function assigns a different loss/penalty for different types of mistakes (e.g. if misclassifying a digit 8 as 9 incurs a smaller loss than misclassifying it as 0), then the minimum achievable error as well as the optimal classifier achieving this error will be different. This is true also in the case of binary classification, where for example the cost mis-diagnosing a cancer patient as normal could be higher than mis-diagnosing a normal patient as having cancer (can you see how the optimal binary classifier would change in this case?). Such problems are often referred to as **cost-sensitive classification**.

leading to the following optimal classifier (under the 0-1 loss above):

$$\begin{aligned} h^*(\mathbf{x}) &\in \arg\max_{y\in[r]} \frac{p_y\, f_y(\mathbf{x})}{\sum_{y'=1}^{r} p_{y'}\, f_{y'}(\mathbf{x})} \\ &= \arg\max_{y\in[r]} p_y\, f_y(\mathbf{x}) \\ &= \arg\max_{y\in[r]} \ln(p_y\, f_y(\mathbf{x})) \\ &= \arg\max_{y\in[r]} \ln p_y - \frac{1}{2}\ln|\boldsymbol{\Sigma}_y| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^\top \boldsymbol{\Sigma}_y^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)\,. \end{aligned}$$

Again, the parameters $\boldsymbol{\mu}_y$, $\boldsymbol{\Sigma}_y$, p_y can be estimated from a given training sample $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ (e.g. using maximum likelihood estimation as before), and a plug-in classifier using the estimated values can then be constructed based on the above. Note that the above classifier amounts to estimating parameters that determine r quadratic functions $g_y : \mathcal{X} \to \mathbb{R}$ for $y \in [r]$, and classifying an instance $\mathbf{x}$ according to a label y with largest value of $g_y(\mathbf{x})$. Similarly, if the class-conditional covariances are assumed to be equal, the above classifier will amount to learning parameters determining r linear functions, and classifying according to the largest value.

Exercise. Let $\mathcal{X} = \{0,1\}^d$ and $\mathcal{Y} = [r]$, and assume the features are conditionally independent given the class label. Can you derive the naïve Bayes classifier in this setting?