Topics

- Classification
  - regression approach to classification
  - elementary decision theory
  - Fisher linear discriminant
  - Generative probabilistic classifiers
  - discriminative classifiers: logistic regression
### Classification

Example: digit recognition (8x8 binary digits)

<table>
<thead>
<tr>
<th>binary digit</th>
<th>actual label</th>
<th>target label in learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Digit Image" /></td>
<td>“2”</td>
<td>1</td>
</tr>
<tr>
<td><img src="image2" alt="Digit Image" /></td>
<td>“2”</td>
<td>1</td>
</tr>
<tr>
<td><img src="image3" alt="Digit Image" /></td>
<td>“1”</td>
<td>0</td>
</tr>
<tr>
<td><img src="image4" alt="Digit Image" /></td>
<td>“1”</td>
<td>0</td>
</tr>
<tr>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
</tr>
</tbody>
</table>
Classification via regression

- We ignore the fact that the output is binary (e.g., 0/1) rather than a continuous variable.

Given a linear regression function

\[ f(x; w) = w_0 + w_1 x_1 + \ldots + w_d x_d \]

we minimize the squared difference between the predicted output (continuous) and the observed label (binary):

\[ J_n(w) = \frac{1}{2} \sum_{i=1}^{n} (y_i - f(x_i; w))^2 \]

- How do we classify any new example \( x \)?
Classification via regression cont’d

\[ f(x; w) = w_0 + w_1 x_1 + \ldots + w_d x_d \]

Any new (test) example \( x \) can be classified according to

\[ \text{label} = 1 \text{ if } f(x; w) > 0.5, \text{ and label} = 0 \text{ otherwise} \]

where \( f(x; w) = 0.5 \) defines the decision boundary.
Classification via regression cont’d

- This is not optimal...

sometimes good

sometimes bad
Suppose we know the distribution of examples in each class, i.e., we know the class-conditional densities $p(x|y=0)$ and $p(x|y=1)$. How do we decide (optimally) which class a new example $x'$ should belong to?

The optimal decisions in the sense of the lowest possible miss-classification error are based on the log-likelihood ratio

$$y = 1 \text{ if } \log \frac{p(x'|y=1)}{p(x'|y=0)} > 0$$

and $y = 0$ otherwise.
Decision theory cont’d

- When the examples fall more often in one class than another, we have to modify the decision rule a bit:
  
  \[ y = 1 \text{ if } \log \frac{p(x'|y = 1)P(y = 1)}{p(x'|y = 0)P(y = 0)} > 0 \]

  and \( y = 0 \) otherwise

- More generally, the Bayes optimal decisions are given by
  
  \[ y' = \arg \max_{y=0,1} \{ p(x'|y)P(y) \} = \arg \max_{y=0,1} \{ P(y|x') \} \]

  (this is optimal only if we have the correct densities and prior frequencies)
Linear regression and projections

- Evaluating any linear regression function (here in 2D)

\[ f(x; w) = w_0 + w_1 x_1 + w_2 x_2 = w_0 + x^T \mathbf{w} \]

amounts to projecting the points \( x = [x_1 \ x_2]^T \) to a line parallel to \( \mathbf{w} \).

Since we are primarily interested in how the points in the two classes are separated by this projection, we can temporarily forget the bias/offset term \( w_0 \).
Beyond regression

- By varying the lines (or $\vec{w}$) we get different levels of separation between the classes.
Fisher linear discriminant

• We find a direction $\vec{w}$ in the input space such that the projected points become “well-separated”.

• Some notation:
  - class 0: $n_0$ samples, mean $\mu_0$, covariance $\Sigma_0$
  - class 1: $n_1$ samples, mean $\mu_1$, covariance $\Sigma_1$
Fisher linear discriminant cont’d

- Estimation criterion: we find $\mathbf{w}$ that maximizes

$$J_{Fisher}(\mathbf{w}) = \frac{(\text{Separation of projected means})^2}{\text{Sum of within class variances}}$$

$$= \frac{(\mathbf{w}^T \mu_1 - \mathbf{w}^T \mu_0)^2}{\mathbf{w}^T (n_1 \Sigma_1 + n_0 \Sigma_0) \mathbf{w}}$$

- The solution

$$\mathbf{w} \propto (n_1 \Sigma_1 + n_0 \Sigma_0)^{-1} (\mu_1 - \mu_0)$$

is Bayes optimal for two normal populations with equal covariances ($\Sigma_1 = \Sigma_0$)
Fisher linear discriminant analysis: example

- Binary digits “1” versus “7”

\[ \vec{w} = c \]

This is roughly speaking the elementwise matrix difference

\[ \vec{w} \approx \]

Tommi Jaakkola, MIT AI Lab
Generative and discriminative classification

- To further refine our classification approach we can adopt one of two general frameworks:

1. Generative (model $p(x|y)$)
   - directly build class-conditional densities over the multi-dimensional input examples
   - classify new examples based on the densities

2. Discriminative (only model $P(y|x)$)
   - only model decisions given the input examples; no model is constructed over the input examples
Generative approach to classification

- We can directly model each class conditional population with a multi-variate normal (Gaussian) distribution

\[
\mathbf{x} \sim N(\mu_1, \Sigma_1), \quad y = 1
\]

\[
\mathbf{x} \sim N(\mu_0, \Sigma_0), \quad y = 0
\]

where

\[
p(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\}
\]

and \( \mathbf{x} = [x_1, \ldots, x_d]^T \).
Mixture classifier: decisions

- Examples \( x \) are classified on the basis of which Gaussian explains the data better

\[
\log \frac{p(x | \mu_1, \Sigma_1)}{p(x | \mu_0, \Sigma_0)} > 0 \quad y = 1 \\
\leq 0 \quad y = 0
\]

or, more generally, when the classes have different a priori probabilities, we use the posterior probability

\[
P(y = 1 | x) \propto p(x | \mu_1, \Sigma_1)P(y = 1)
\]

- The corresponding decision boundaries are

\[
\log \frac{p(x | \mu_1, \Sigma_1)}{p(x | \mu_0, \Sigma_0)} = 0 \quad \text{or} \quad P(y = 1 | x) = 0.5
\]
Mixture classifier: decision rule

- Equal covariances

\[ x \sim N(\mu_1, \Sigma), \quad y = 1 \]
\[ x \sim N(\mu_0, \Sigma), \quad y = 0 \]

- The decision rule is *linear*
Mixture classifier: decision rule

- Unequal covariances

\[
x \sim N(\mu_1, \Sigma_1), \quad y = 1
\]
\[
x \sim N(\mu_0, \Sigma_0), \quad y = 0
\]

- The decision rule is \textit{quadratic}
Maximum likelihood estimation

- We can estimate the class conditional densities $p(x|\mu, \Sigma)$ separately.

For a multivariate Gaussian model

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu)\}$$

the maximum likelihood estimates of the parameters based on a random sample $\{x_1, \ldots, x_n\}$ are given by the sample mean and sample covariance:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i, \quad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T$$
Discriminative classification

- If we are only interested in the classification decisions, why should we bother with a model over the input examples?

- We could try to directly estimate the conditional distribution of labels given the examples or $P(y|x, \theta)$ where $\theta = \{\mu_0, \mu_1, \Sigma_0, \Sigma_1\}$. 
Back to the Gaussians... (1-dim)

- When the classes are equally likely \textit{a priori}, the posterior probability of the label $y = 1$ given $x$ is given by

$$P(y = 1|x, \theta) = \frac{p(x|\mu_1, \sigma^2_1)}{p(x|\mu_1, \sigma^2_1) + p(x|\mu_0, \sigma^2_0)}$$

$$= \frac{1}{1 + \frac{p(x|\mu_0, \sigma^2_0)}{p(x|\mu_1, \sigma^2_1)}}$$

$$= \frac{1}{1 + \exp \left\{ - \log \frac{p(x|\mu_1, \sigma^2_1)}{p(x|\mu_0, \sigma^2_0)} \right\}}$$

where $\theta = \{\mu_0, \mu_1, \sigma^2_1, \sigma^2_0\}$. 

Tommi Jaakkola, MIT AI Lab
Form of the posterior

- Since the decision boundary is *linear* or *quadratic*, we know that

\[
\log \frac{P(x|\mu_1, \sigma_1^2)}{P(x|\mu_0, \sigma_0^2)} = \begin{cases} 
  w_0 + w_1 x, & \text{when } \sigma_1^2 = \sigma_0^2 \\
  w'_0 + w'_1 x + w'_2 x^2, & \text{otherwise}
\end{cases}
\]

for some coefficients \( w \).

When \( \sigma_1^2 = \sigma_0^2 \), we get

\[
P(y = 1|x, \theta) = \frac{1}{1 + \exp \left\{ - \log \frac{p(x|\mu_1, \sigma_1^2)}{p(x|\mu_0, \sigma_0^2)} \right\}}
\]

\[
= \frac{1}{1 + \exp\left\{ - (w_0 + w_1 x) \right\}}
\]
Generalized linear models

- The posterior class probability $P(y = 1|\mathbf{x})$ can often be reduced to a *logistic regression model*

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g\left( w_0 + w_1 x_1 + \ldots + w_d x_d \right)$$

with parameters $\mathbf{w}$.

Here the “squashing function”

$$g(z) = \frac{1}{1 + \exp(-z)}$$

that turns linear predictions into probabilities is known as the *logistic function*. 
Fitting logistic regression models

As in the case of linear regression models we can fit the logistic models using the maximum log-likelihood criterion

\[ l(D; w) = \sum_{i=1}^{n} \log P(y_i | x_i, w) \]

where

\[ P(y = 1 | x, w) = g \left( w_0 + w_1 x_1 + \ldots + w_d x_d \right) \]

(Note: although we can relate the resulting parameters to some class-conditional means and covariances, their values would be rather different from their values in the generative paradigm)