CS195-5: Introduction to Machine Learning
Lecture 7

Greg Shakhnarovich

September 20, 2006
Revised October 24th, 2006
Announcements

• PS1 clarifications:
  – P.5: “quadratic regression” means quadratic in $x$.
  – P.5: Why not try higher order models?

• MLRG today: introduction to sampling methods
Review

- Fisher's criterion: $J_{Fisher}(w) = \frac{\text{separation between projected means}^2}{\text{sum of projected within-class variances}}$

  - Resulting 1D projection:

    $$\hat{w} \propto (N_{-1}S_{-1} + N_{+1}S_{+1})^{-1}(m_{+1} - m_{-1})$$

    where $S_c = \frac{1}{N_c} \sum_{y_i = c}(x_i - m_c)(x_i - m_c)^T$.

- Decision boundary set by $\hat{w}^T x + w_0 = 0$. 
Linear separation of classes

- Classifying using a linear decision boundary (as in Fisher’s method) effectively reduces the data dimension to 1.

- Important questions:
  - What’s the optimal projection?
  - How does one set the bias $w_0$?
  - Can we do better with more complex decision boundaries?
Risk of a classifier

- The risk (expected loss) of a $C$-way classifier $h(x)$:

$$R(h) = \int x \sum_{c=1}^{C} L(h(x), c) p(x, y = c) dx$$

$$= \int x \left[ \sum_{c=1}^{C} L(h(x), c) p(y = c | x) \right] p(x) dx$$

- Clearly, it’s enough to minimize the conditional risk for any $x$:

$$R(h | x) = \sum_{c=1}^{C} L(h(x), c)p(y = c | x).$$
Conditional risk of a classifier

\[ R(h \mid x) = \sum_{c=1}^{C} L(h(x), c) p(y = c \mid x) \]
Conditional risk of a classifier

\[ R(h \mid x) = \sum_{c=1}^{C} L(h(x), c)p(y = c \mid x) \]

\[ = 0 \cdot p(y = h(x) \mid x) + 1 \cdot \sum_{c \neq h(x)} p(y = c \mid x) \]
Conditional risk of a classifier

\[ R(h \mid x) = \sum_{c=1}^{C} L(h(x), c)p(y = c \mid x) \]

\[ = 0 \cdot p(y = h(x) \mid x) + 1 \cdot \sum_{c \neq h(x)} p(y = c \mid x) = \sum_{c \neq h(x)} p(y = c \mid x) \]
Conditional risk of a classifier

\[ R(h \mid x) = \sum_{c=1}^{C} L(h(x), c)p(y = c \mid x) \]

\[ = 0 \cdot p(y = h(x) \mid x) + 1 \cdot \sum_{c \neq h(x)} p(y = c \mid x) = \sum_{c \neq h(x)} p(y = c \mid x) \]

\[ = 1 - p(y = h(x) \mid x). \]

- Thus, to minimize conditional risk given \( x \), the classifier must decide

\[ h(x) = \arg\max_c p(y = c \mid x). \]

- This is the best possible classifier in terms of generalization, i.e. expected misclassification rate on new examples.
Bayes rule

- Some terminology:
  
  *class-conditional* density  \( p_c(x) = p(x \mid y = c) \)
  
  *(also called *likelihood)*
  
  *prior* probability  \( P_c = p(y = c) \)
  
  *posterior* probability  \( p(y = c \mid x) \)
  
  *compound* density/probability of data  \( p(x) = \sum_c p(x, y = c) = \sum_c p_c(x) P_c. \)

- Usually we don’t have direct access to \( p(y \mid x) \). But suppose we know \( p(x \mid y) \) and \( p(y) \).

- Bayes rule: Using the product rule  \( p(a, b) = p(a \mid b) p(b) = p(b \mid a) p(a), \)
  \[
p(y \mid x) = \frac{p(x \mid y) p(y)}{p(x)}.
  \]
Bayes classifier

\[ p(y \mid x) = \frac{p(x \mid y) p(y)}{p(x)}. \]

- The classifier that minimizes conditional risk for given \( p(x \mid y), p(y) \) is called the Bayes classifier

\[ h^*(x) = \arg\max_c p(y = c \mid x) \]

\[ = \arg\max_c \frac{p(x \mid y = c) p(y = c)}{p(x)} \]
Bayes classifier

\[
p(y | x) = \frac{p(x | y)p(y)}{p(x)}.
\]

- The classifier that minimizes conditional risk for given \( p(x | y), p(y) \) is called the Bayesian classifier

\[
h^*(x) = \arg\max_c p(y = c | x)
\]

\[
= \arg\max_c \frac{p(x | y = c)p(y = c)}{p(x)}
\]

\[
= \arg\max_c p(x | y = c)p(y = c)
\]

(Data probability term \( p(x) \) is equal for all \( c \)s.)
Bayes classifier

\[ p(y | x) = \frac{p(x | y)p(y)}{p(x)} . \]

- The classifier that minimizes conditional risk for given \( p(x | y), p(y) \) is called the Bayes classifier

\[ h^*(x) = \arg \max_c p(y = c | x) \]

\[ = \arg \max_c \frac{p(x | y = c)p(y = c)}{p(x)} \]

\[ = \arg \max_c p(x | y = c)p(y = c) \]

\[ = \arg \max_c \{ \log p_c(x) + \log P_c \} . \]

(Data probability term \( p(x) \) is equal for all cs.)
Optimal decision regions

- A decision region is defined for each $c$: $D_c(h) = \{x : h(x) = c\}$.

- If $\forall c, P_c = 1/C$, i.e. classes are equally likely, the optimal decision regions are simply

$$D_c(h^*) = \{x : c = \arg \max_{c'} p_{c'}(x)\}.$$
The risk (probability of error) of Bayes classifier \( h^* \) is called the Bayes risk \( R^* \).

This is the minimal achievable risk for the given \( p(x, y) \) with any classifier!

In a sense, \( R^* \) measures the inherent difficulty of the classification problem.
Bayes risk

- The risk (probability of error) of Bayes classifier $h^*$ is called the Bayes risk $R^*$.

- This is the minimal achievable risk for the given $p(x, y)$ with any classifier!

- In a sense, $R^*$ measures the inherent difficulty of the classification problem.

Easier to express in terms of probability of being correct:

$$R^* = 1 - \int_x dx$$
• The risk (probability of error) of Bayes classifier $h^*$ is called the Bayes risk $R^*$.

• This is the minimal achievable risk for the given $p(x, y)$ with any classifier!

• In a sense, $R^*$ measures the inherent difficulty of the classification problem.

• Easier to express in terms of probability of being correct:

$$R^* = 1 - \int_x \max_c \left\{ p(x \mid c = y) \, P_c \right\} \, dx$$
Discriminant function

- We can construct, for each class $c$, a *discriminant function*

\[
\delta_c(x) \triangleq \log p_c(x) + \log P_c
\]

such that

\[
h^*(x) = \arg\max_c \delta_c(x).
\]

- We will always simplify $\delta_c$ by removing terms and factors that are common for all $\delta_c$ since they won’t affect the decision boundary.

  - For example, if $P_c = 1/C$ for all $c$, we can drop the prior term:

\[
\delta_c(x) = \log p_c(x)
\]
Two-category case

• In case of two classes \( y \in \{\pm 1\} \), the Bayes classifier is

\[
h^*(x) = \arg\max_{c=\pm 1} \delta_c(x) = \text{sign} \left( \delta_+ (x) - \delta_- (x) \right).
\]

• Decision boundary is given by \( \delta_+ (x) - \delta_- (x) = 0 \).
  
  – Sometimes \( f(x) = \delta_+ (x) - \delta_- (x) \) is referred to as a discriminant function.

• With equal priors, this is equivalent to the (log)-likelihood ratio test:

\[
h^*(x) = \text{sign} \left[ \log \frac{p(x \mid y = +1)}{p(x \mid y = -1)} \right].
\]
Linear discriminant functions

- When \( \delta_c \) are linear, the decision boundary is also linear.

- Example: class-conditionals are multivariate Gaussians with common covariance matrix
  \[
  p_c(x) = \mathcal{N}(x; \mu_c, \Sigma)
  \]

- As shown in Problem Apple-9,
  \[
  \delta_c = \mu_c^T \Sigma^{-1} x - \frac{1}{2} \mu_c^T \Sigma^{-1} \mu_c
  \]
Linear discriminant functions

• When $\delta_c$ are linear, the decision boundary is also linear.

• Example: class-conditionals are multivariate Gaussians with common covariance matrix

$$p_c(x) = \mathcal{N}(x; \mu_c, \Sigma)$$

• As shown in Problem Apple-9,

$$\delta_c = \mu_c^T \Sigma^{-1} x - \frac{1}{2} \mu_c^T \Sigma^{-1} \mu_c + \log P_c.$$

• This is a linear (in $x$) discriminant, thus the decision boundary is linear.
Fisher’s linear discriminant analysis revisited

- Assume two Gaussian class-conditionals, with equal covariances.

- The optimal decision boundary is

\[
\delta_{+1}(x) - \delta_{-1}(x) = (\mu_{+1} - \mu_{-1})^T \Sigma^{-1} x - \frac{1}{2} \mu_{+1}^T \Sigma^{-1} \mu_{+1} + \frac{1}{2} \mu_{-1}^T \Sigma^{-1} \mu_{-1} \\
+ \log P_{+1} - \log P_{-1} = 0,
\]

which is exactly the form we got for Fisher’s LDA (plus we have a recipe for how to set \(w_0\)).

- of course, instead of \(\mu_c, \Sigma\) in practice we use ML estimates \(m_c, \sum_c N_c S_c\).

- So, under the assumption above, Fisher’s LDA (with this choice of \(w_0\)) is decision-theoretically optimal, up to estimation errors for means and covariance.
Generative models for classification

- In generative models one explicitly models $p(x, y)$ or, equivalently, $p_c(x)$ and $P_c$, to derive discriminants.

- Typically, the model imposes certain parametric form on the assumed distributions, and requires estimation of the parameters from data.
  - Most popular: Gaussian for continuous, multinomial for discrete.
  - We will see later in this class non-parametric models.

- Often, the classifier is OK even if data clearly don’t conform to assumptions.
Maximum likelihood density estimation

- Let $X = \{x_1, \ldots, x_N\}$ be a set of data points
  - no labels; in the current context $X$ all come from class $c$

- We assume parametric distribution model $p(x; \theta)$.

- The (log)-likelihood of $\theta$ given $X$ (assuming i.i.d. sampling):
  \[
  \ell(X; \theta) \triangleq \sum_{i=1}^{N} \log p(x_i; \theta).
  \]

- ML estimate of $\theta$:
  \[
  \hat{\theta}_{ML} \triangleq \arg\max_{\theta} \ell(X; \theta)
  \]
  - Intuitively: the observed data is most likely (has highest probability) for these settings of $\theta$. 
Gaussians with unequal covariances

- What if we remove the restriction that $\forall c, \Sigma_c = \Sigma$?

- Compute ML estimate for $\mu_c, \Sigma_c$ for each $c$.

- We get discriminants (and decision boundaries) quadratic in $x$:

\[ \delta_c(x) = -\frac{1}{2} x^T \Sigma^{-1}_c x + \mu^T_c \Sigma^{-1}_c x - \langle \text{const in } x \rangle \]

(as shown in Problem Apple-10).

A quadratic form in $x$: $x^T A x$. 
Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?
Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?

- Second-degree curves can be any *conic section*:
Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?
- Second-degree curves can be any conic section:
Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?
- Second-degree curves can be any conic section:
Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?

- Second-degree curves can be any conic section:

- Can all of these arise from two Gaussian classes?
Next time

More on generative models.
Naïve Bayes classifiers.
Discrete data.