Lecture 8: Classification

Assessing Performance of Classification Models

Prof. Alexandra Chouldechova
95-791: Data Mining

April 7, 2016
Course Roadmap

95-791

Predictive Analytics

Descriptive Analytics

Prediction

Classification
Prediction topics

Classification

- Logistic regression
- Similarity-based models
- Bayes classifiers
- Tree-based models
- Cost-sensitive assessment
- Model validation
Prediction topics

Classification

- Logistic regression
- Similarity-based models
- Bayes methods
- Tree-based models
- Cost-sensitive assessment
- Model validation
Agenda

• **Bayes Methods (Generative Models)**
  - Linear discriminant analysis
  - Quadratic discriminant analysis
  - Naive Bayes

• **Assessing performance of Classification Models**
  - Calibration plots
  - Confusion matrices
  - Sensitivity, Specificity, Accuracy, Precision, Recall
  - Cost-based criteria
  - ROC curves
Linear discriminant analysis

\[ \hat{y}_0 = \arg\max_{k=1,\ldots,K} \frac{\pi_k f_k(x_0)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_0)} = \arg\max_{k=1,\ldots,K} \pi_k f_k(x_0) \]

- **Linear discriminant analysis** (LDA)\(^1\) assumes all the \(f_k(x)\) are **Multivariate Normal** \((\mu_k, \Sigma)\)
  - Different means, *same* covariance matrix

---

\(^1\)Not to be confused with Latent Dirichlet Allocation... also abbreviated LDA
What is a Multivariate Normal?

Figure 4.5 from ISL. Bivariate normal density for two choices of $\Sigma$. 
Covariance matrix \( \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \)

\[
\begin{array}{c|ccc}
\mu_k & \text{Class 1} & \text{Class 2} & \text{Class 3} \\
\hline
(2, 2) & (-1, 3) & (0, -1)
\end{array}
\]
Covariance matrix $\Sigma = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}$

$\mu_k =$

- Class 1: $(2, 2)$
- Class 2: $(-1, 3)$
- Class 3: $(0, -1)$
Linear discriminant analysis

Figure 4.6 from ISL

- **Left**: $K = 3$ classes. True $f_k$ are Bivariate Normal. Ovals represent 95% density contours for each class (95% of points generated from a class will fall into the ellipse). Dashed lines are the LDA boundary from the true $f_k$.

- **Right**: 20 points are observed from each class. Solid lines are the estimated LDA decision boundaries. We had to estimate $\mu_k$ and $\Sigma$ from the data.
It's called **Linear discriminant analysis** because the choice $f_k = \text{MVN}(\mu_k, \Sigma)$ always results in *linear decision boundaries*.

The LDA rule amounts to classifying $x_0$ to the class with the highest value of the **discriminant function** $\delta_k$:

$$
\delta_k(x_0) = x_0^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \log(\hat{\pi}_k)
$$
Quadratic discriminant analysis

- **Quadratic discriminant analysis** (QDA) models $f_k(x)$ as $\text{MVN}(\mu_k, \Sigma_k)$: it relaxes LDA's assumption that all the covariance matrices are the same.

- This produces *quadratic* decision boundaries.
\[ \Sigma_1 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 3 & 2.5 \\ 2.5 & 3 \end{pmatrix} \quad \Sigma_3 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \]

\[
\begin{array}{c|ccc}
\mu_k = & \text{Class 1} & \text{Class 2} & \text{Class 3} \\
(2, 2) & (-1, 3) & (0, -1) 
\end{array}
\]
Quadratic discriminant analysis

Figure 4.9 from ISL. Dashed purple curve is the Bayes classifier decision boundary. Solid green curve is QDA, dotted black line is LDA. Left: True boundary is linear. Right: True boundary is quadratic.
Logistic regression vs. Linear discriminant analysis

- When the classes are *well separated*, Logistic regression coefficient estimates are *really unstable*. LDA does not have this problem.
  - You will explore this Logistic regression instability issue on HW4

- **LDA makes assumptions** on the distribution of $X \mid Y = k$
  - When the assumptions hold (or approximately hold), LDA can produce better, *more stable* decision boundaries, even when $n$ is small

- When we have $K = 2$ classes, Logistic regression can be extended in all kinds of ways (additive models, regularized models, etc.), and is highly interpretable

- Logistic regression gets hard to interpret for $K > 2$ classes. LDA is essentially the same regardless of how many classes we have.
Naive Bayes

- Naive Bayes is a popular simple classifier in cases where we have a lot of predictors $p$

- Imagine $p = 1000$ and $n = 2000$. It's going to be extremely difficult to accurately estimate $f_k(x)$ without really strong assumptions on $f_k$

- Naive Bayes says: Let's assume that all components of $X = (X_1, X_2, \ldots, X_p)$ are independent

- Under the independence assumption, $f_k(x)$ simplifies to:

$$f_k(x) = P(X_1 = x_1, \ldots, X_p = x_p \mid Y = k)$$
$$= P(X_1 = x_1 \mid Y = k) \times \cdots \times P(X_p = x_p \mid Y = k)$$
$$= \prod_{j=1}^{p} P(X_j = x_j \mid Y = k)$$

- So now to estimate $f_k(x)$ we just need to estimate $p$ univariate densities: $f_k(x_j) = P(X_j = x_j \mid Y = k)$

\(^2\text{We're actually assuming conditional independence: The inputs are independent given the class labels}\)
Naive Bayes with the Default Data

Prior probabilities

P(Default = ...)  
default  
No  
Yes

default
No
Yes

P(student = Yes | default = ...)

default  
No  
Yes

default
No
Yes
Naive Bayes with the Default Data

Prior probabilities

Balance distribution

Income distribution

Default probability

Student probability
Calculate:
\[ \hat{f}_{Yes}(i, b, s) = \hat{f}_{Yes}(income) \times \hat{f}_{Yes}(balance) \times \hat{f}_{Yes}(student) \]
\[ \hat{f}_{No}(i, b, s) = \hat{\pi}_{No} \hat{f}_{No}(income) \times \hat{f}_{No}(balance) \times \hat{f}_{No}(student) \]

Naive Bayes posterior probability estimate of \( \text{default} = \text{Yes} \):
\[
\hat{P}(\text{default} = \text{Yes} | i, b, s) = \frac{\hat{\pi}_{Yes} \hat{f}_{Yes}(i, b, s)}{\hat{\pi}_{Yes} \hat{f}_{Yes}(i, b, s) + \hat{\pi}_{No} \hat{f}_{No}(i, b, s)}
\]
Naive Bayes vs. LDA vs. QDA

• Naive Bayes scales well to problems where $p$ is large
  ○ If you have enough data to estimate the univariate density of each predictor (i.e., enough to form a nice histogram), you can apply Naive Bayes

• In LDA, we have to estimate $K \times p$ parameters to get $\hat{\mu}_k$'s and another $\frac{1}{2}p(p + 1)$ parameters to estimate the $p \times p$ covariance matrix $\hat{\Sigma}$.

• In QDA, we have to estimate the means and $K \ p \times p$ covariance matrices. That's $\frac{1}{2}Kp(p + 1)$ parameters!

• So why do even bother with methods like LDA or QDA?
  ○ They can capture meaningful interactions. Naive Bayes cannot.
An example where Naive Bayes fails

This is not a difficult classification problem. Logistic regression with formula $y \sim X1 \ast X2$ gives a perfect classifier.
Naive Bayes views $X_1$ and $X_2$ independently, and cannot produce the rule: “If $X_1$ and $X_2$ are both small or both large, classify as $\hat{Y} = 1$"
Assessing the performance of Classifiers
Assessing Classifier Performance

• We now have a bunch of different ways of estimating the conditional probability (aka posterior probability, if we take a Bayes approach)

\[ p_k(x) = \mathbb{P}(Y = k \mid X = x) \]

• Now we can start asking questions about whether the estimate \( \hat{p}_k(x) \) is a good one, and whether it results is a good decision rule

• Let's start with the question:

Is \( \hat{p}_k(x) \) a good estimate of \( p_k(x) \)?
Are we doing a good job of estimating $p_k(x)$?

Is $\hat{p}_k(x)$ a good estimate of $p_k(x)$?

- **Why do we care?**
  - An email whose spam probability is $\hat{p}_k(x) = 0.52$ will get classified to spam the same as an email whose spam probability is $\hat{p}_k(x) = 0.999$. But one email is borderline, while the other is extremely likely to be spam. This is important information.
  - Whether to pursue suspicious insurance claim may depend on the estimated **probability** that it's fraudulent and various cost-related factors.
  - Customer Lifetime Value (CLV) calculations require an estimate of the **probability** that a customer will make a purchase.
Calibration plots

1. Bin the data according to \( \hat{p}_k(x) \): E.g., bins could be
   
   \[ [0, 0.1], (0.1, 0.2], \ldots, (0.9, 1] \.

2. For each bin, calculate the proportion of observations in bin \( b \) that had class \( Y = k \).

3. Plot the midpoints of the bins on the \( x \)-axis and the proportions from Step 2. on the \( y \)-axis

- **Note**: Certain methods are obviously poorly calibrated. E.g., we saw that linear regression can return *negative values*

- In such cases, a popular approach is to use the softmax transformation

\[
\hat{p}_k^* = \frac{e^{\hat{y}_k}}{\sum_{\ell=1}^{K} e^{\hat{y}_\ell}}
\]

These values at least all lie in \([0, 1]\) and sum to 1
Fig. 11.1: *Left*: A simulated two-class data set with two predictors. The *solid black line* denotes the 50% probability contour. *Right*: A calibration plot of the test set probabilities for random forest and quadratic discriminant analysis models.

The RF is well-calibrated. The QDA model **underestimates** the true probability when it's low, and **overestimates** it when it's high.

[source: Applied Predictive Modeling]
The QDA calibration curve looks S-shaped. The technical term to describe this shape is: *sigmoidal*. We can try recalibrating by feeding the $\hat{p}(x)$ into a logistic regression to predict $y_i$, and then use the resulting probabilities. In some cases this actually works.
Evaluating Classifications

- Let's focus again on the **binary classification** setting:
  - \( Y = 1 \): if the event happened
  - \( Y = 0 \): if the event did not happen

- The primary building block of essentially all approaches to evaluating a Classifier is the **confusion matrix**

Table 11.1: The confusion matrix for the two-class problem ("events" and "nonevents." The table cells indicate number of the true positives (\( TP \)), false positives (\( FP \)), true negatives (\( TN \)), and false negatives (\( FN \))

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Nonevent</td>
</tr>
<tr>
<td>Event</td>
<td>( TP )</td>
</tr>
<tr>
<td>Nonevent</td>
<td>( FN )</td>
</tr>
</tbody>
</table>

[source: Applied Predictive Modeling]
The Confusion Matrix

Table 11.1: The confusion matrix for the two-class problem ("events" and "nonevents.") The table cells indicate number of the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN)

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>TP</td>
</tr>
<tr>
<td>Nonevent</td>
<td>FN</td>
</tr>
</tbody>
</table>

- Each cell is a count. The cells in total add up to $n$
- The **diagonal** entries are **correct classifications**
- The **off-diagonal** entries are classification **errors**

[source: Applied Predictive Modeling]
What does it mean to have a good classifier?

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Nonevent</td>
</tr>
<tr>
<td>Event</td>
<td>$TP$</td>
</tr>
<tr>
<td>Nonevent</td>
<td>$FN$</td>
</tr>
<tr>
<td>Nonevent</td>
<td>$TN$</td>
</tr>
</tbody>
</table>

- **Accuracy** = $(TP + TN)/n$
- **Misclassification rate** = $(FN + FP)/n = 1 - Accuracy$
- If $FP$ and $FN$ have the same cost, then our goal is to minimize the **Misclassification rate**
- E.g., if $Y \in \{\text{cat photo, dog photo}\}$, we don't care if our error was mistaking a cat for a dog or vice versa
- If $Y \in \{\text{fraud, not fraud}\}$, the cost of a $FN$ (failing to catch fraud) is typically much higher than the cost of a $FP$ (further investigating a false lead)
What does it mean to have a good classifier?

• Suppose I tell you my classifier has 97% accuracy. Are you impressed?

• Well, what if \( Y \in \{ \text{wins lottery, does not win} \} \)?
  ○ The chances that a ticket wins the Powerball lottery are way smaller than 1% (they're 1 in 175,000,000)
  ○ By classifying all tickets as “does not win”, we'd have an accuracy of 
    \(1 - 1/175,000,000 = 0.99999\ldots\)
  ○ So a 97% accuracy in this case is actually really bad

• We must take the baseline probabilities (i.e., prior probabilities) of each class into account when assessing performance
Sensitivity, Specificity

- We're now going to look at ways of quantifying the performance of a classifier that go beyond the simple notion of **Accuracy**

- **Sensitivity**: aka Recall, Hit rate
  
  \[
  \text{Sensitivity} = \frac{\#\text{observations correctly classified to have the event}}{\#\text{observations that had the event}} = \frac{TP}{TP + FN}
  \]

- **Specificity**: aka True Negative Rate (TNR)

  \[
  \text{Specificity} = \frac{\#\text{observations correctly classified as non-events}}{\#\text{observations that did not have the event}} = \frac{TN}{TN + FP}
  \]
A classifier with high **Sensitivity** is desirable when **False Negatives** (e.g., failing to detect fraud) are more costly than **False Positives** (flagging a case that turns out to be non-fraudulent).

A classifier with high **Specificity** is desirable when **False Positives** (convicting an innocent person of a crime) are more costly than **False Negatives** (failing to convict a guilty person of a crime).

**Sensitivity** and **Specificity** tend to move in opposite directions:
- To increase **Sensitivity**, we can always flag more cases as potentially fraudulent. But this would decrease **Specificity** because more non-fraud cases would now be misclassified as fraud.
An example: Marketing data

- The Marketing data set contains information on bank customers who were contacted by marketers wanting them to open an account
- \( Y = \text{“Yes” if the customer opened an account when contacted, “No” otherwise} \)
- Here's a confusion matrix obtained from fitting a logistic regression model, and classifying \( Y = 1 \) if \( \hat{p}(x) \geq 0.25 \)

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>121</td>
</tr>
<tr>
<td>No</td>
<td>916</td>
</tr>
</tbody>
</table>

- \( n = 121 + 234 + 916 + 7771 = 9042 \)
- \( \text{Accuracy} = (121 + 7771)/n = 87 \)
- \( \text{Misclassification rate} = (234 + 916)/n = 13\% \)
- \( \text{Prevalence} = (121 + 916)/n = 11.5\% \)
Marketing data

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>121</td>
<td>234</td>
</tr>
<tr>
<td>No</td>
<td>916</td>
<td>7771</td>
</tr>
</tbody>
</table>

- \( n = 121 + 234 + 916 + 7771 = 9042 \)
- **Accuracy** = \( (121 + 7771)/n = 87 \)
- **Misclassification rate** = \( (234 + 916)/n = 13\% \)
- **Prevalence** = \( (121 + 916)/n = 11.5\% \)
- **Sensitivity (Recall)** = \( 121/(121 + 916) = 11.6\% \)
- **Specificity** = \( 7771/(7771 + 234) = 97.1\% \)
- So at the cutoff, \( \hat{p}(x) \geq 0.25 \), our classifier has **low Sensitivity** and **high Specificity**
- This is not a good setting for marketing.
Some other measures of performance

- Here are some other quantities that people have names for
- **Positive Predictive Value (PPV):**
  
  \[
  \text{PPV} = \frac{\text{# observations correctly classified to have the event}}{\text{# observations classified to have the event}} = \frac{TP}{TP + FP}
  \]

  - Suppose you take a diagnostic test for a disease. The **PPV** of the diagnostic is the probability that you actually have disease when you test **positive**.

- **Negative Predictive Value: (NPV)**
  
  \[
  \text{NPV} = \frac{\text{# observations correctly classified as non-events}}{\text{# observations classified as non-events}} = \frac{TN}{TN + FN}
  \]

  - Suppose you take a pregnancy test. The **NPV** is the probability that you actually aren't pregnant given that the test comes up **negative**.
Cost-Based Criteria

• All of the criteria we've discussed so far take the form of counts and proportions
• But many real world problems have real costs and benefits
• We may be interested in:
  ◦ Predicting which investments to make to maximize return
  ◦ Improving customer satisfaction through market segmentation
  ◦ Minimize costs associated with fraudulent transactions

• Suppose you work for a clothing retailer and are tasked with mailing out promotional offers
  ◦ It costs you $2.00 to mail a promotion
  ◦ A customer who Responds to the promotion yields you an average gain of $28.40

• We can use the confusion matrix now to calculate profit
Simple profit calculation

Table 11.4: Left: A hypothetical test confusion matrix for a predictive model with a sensitivity of 75% and a specificity of 94.4%. Right: The confusion matrix when a mass mailing is used for all customers

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Observed</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Response Nonresponse</td>
<td>Response Nonresponse</td>
</tr>
<tr>
<td>Response</td>
<td>1,500</td>
<td>1,000</td>
</tr>
<tr>
<td>Nonresponse</td>
<td>500</td>
<td>17,000</td>
</tr>
</tbody>
</table>

- It costs you $2.00 to mail a promotion
- A customer who Responds to the promotion yields you an average gain of $28.40. This is a net gain of $26.40 (benefit of a TP).
- A customer who would have Responded but who you did not reach out to thus loses you $28.40 (cost of a FN)
- Thus the strategy for the Left confusion matrix gives us a profit of:
  \[
  \text{profit} = 26.40TP - 2.00FP - 28.40FN = 23,400
  \]
- If we mailed everyone (strategy on right), the profit would be $16,800

[source: Applied Predictive Modeling]
ROC Curves

- As we vary our probability cutoff, we get different values of all of our performance metrics.
- It is very useful to construct plots that trace out how performance metrics vary as functions of the probability cutoff $\alpha$.
- Perhaps the most widely used plot is the ROC Curve.

![ROC Curve Graph]

Fig. 11.6: A receiver operating characteristic (ROC) curve for the logistic regression model results for the credit model. The dot indicates the value corresponding to a cutoff of 50% while the green square corresponds to a cutoff of 30% (i.e., probabilities greater than 0.30 are called events). The solid black point is the default 50% threshold while the green square corresponds to the performance characteristics for a threshold of 30%. In this figure, the numbers in parentheses are (specificity, sensitivity). Note that the trajectory of the curve between (0, 0) and the 50% threshold is steep, indicating that the sensitivity is increasing at a greater rate than the decrease in specificity. However, when the sensitivity is greater than 70%, there is a more significant decrease in specificity than the gain in sensitivity.

This plot is a helpful tool for choosing a threshold that appropriately maximizes the trade-off between sensitivity and specificity. However, altering the threshold only has the effect of making samples more positive (or negative).
11.3 Evaluating Class Probabilities

Each point on the curve corresponds to the value of (1-Specificity, Sensitivity) calculated at a particular choice of cutoff $\alpha$.

Fig. 11.6: A receiver operator characteristic (ROC) curve for the logistic regression model results for the credit model. The dot indicates the value corresponding to a cutoff of 50% while the green square corresponds to a cutoff of 30% (i.e., probabilities greater than 0.30 are called events).

[source: Applied Predictive Modeling]
• The diagonal is the ROC you would get from randomly picking proportion $\pi_k$ of the observations to classify to class $k$

• Higher ROC is better

• The perfect classifier has $(1 - \text{Specificity}, \text{Sensitivity}) = (0, 1)$
The **AUC** is the *area under the ROC curve*

- **AUC has a nice interpretation:** The AUC is the probability that the classifier will rank a *randomly selected* observation where $y_i = 1$ higher than a *randomly selected* observation where $y_i = 0$
Takeaways

• In the Prediction setting, we focussed entirely on MSE as our performance metric

• There are other options out there, but MSE is by the far the most widely accepted

• In the Classification setting, there are many metrics out there. The set I presented is by no means exhaustive.

• Whatever criterion it is we wish to maximize (e.g., Accuracy, profit, Sensitivity @x% Specificity, Precision @x% Recall, etc.), we can use Cross-validation to estimate the Test set performance according to this metric

Main takeaway:
Classification is way more interesting than Prediction.
Acknowledgements

All of the lectures notes for this class feature content borrowed with or without modification from the following sources:

- 36-462/36-662 Lecture notes (Prof. Tibshirani, Prof. G'Sell, Prof. Shalizi)
- 95-791 Lecture notes (Prof. Dubrawski)
- An Introduction to Statistical Learning, with applications in R (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
- Applied Predictive Modeling, (Springer, 2013), Max Kuhn and Kjell Johnson