Outline

• Logistic regression

• Evaluation of binary classifiers
A task of binary classification: $Y = \{0, 1\}$

Decision boundary takes a form of function $f$ and partitions a feature space into two sets, one for each class.
**Binary classification**

**Hyperplane**

Hyperplane is a linear decision boundary of the form

\[ \Theta^T \mathbf{x} = 0 \]

where direction of \( \langle \theta_1, \theta_2, \ldots, \theta_m \rangle \) is perpendicular to the hyperplane and \( \theta_0 \) determines position of the hyperplane with respect to the origin.
• point if $m = 1$, line if $m = 2$, plane if $m = 3$, …

• we can use hyperplane for classification so that

$$f(x) = \begin{cases} 1 \text{ if } \theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m \geq 0 \\ 0 \text{ if } \theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m < 0 \end{cases}$$

• **linear classifiers** classify examples using hyperplanes
Binary classification
Can we use linear regression?

(Yes) 1

(No) 0
Binary classification
Can we use linear regression?

Fit the data with a linear function $f$

$$f(x) = \Phi^T x$$
Binary classification
Can we use linear regression?

Classify
• if $f(x) \geq 0.5$, predict 1
• if $f(x) < 0.5$, predict 0
Binary classification
Can we use linear regression?

Add one more training instance
Binary classification
Can we use linear regression?

We are heading for the logistic regression algorithm.
Logistic regression

Machine learning process:

1. Formulating the task
2. Getting data
3. Building Logistic regression predictor
4. Final evaluation

Predictor to use
Logistic regression

Logistic regression is a classification algorithm.

Its target hypothesis $f$ for a binary classification has a form of sigmoid function

$$f(x; \Theta) = \frac{1}{1 + e^{-\Theta^\top x}} = \frac{e^{\Theta^\top x}}{1 + e^{\Theta^\top x}}$$

- $g(z) = \frac{1}{1 + e^{-z}}$
- $\lim_{z \to +\infty} g(z) = 1$
- $\lim_{z \to -\infty} g(z) = 0$
Logistic regression

\[ f(x; \Theta) = \frac{1}{1 + e^{-\theta_0 - \theta_1 x_1}} \]
Logistic regression

\[ f(x; \Theta) = \frac{1}{1 + e^{-\theta_0 - \theta_1 x_1}} \]
Logistic regression
Classification rule

Predict a target value using \( \hat{f}(x; \hat{\Theta}) \) so that

- if \( \hat{f}(x; \hat{\Theta}) \geq 0.5 \), i.e. \( \hat{\Theta}^\top x \geq 0 \), predict 1
- if \( \hat{f}(x; \hat{\Theta}) < 0.5 \), i.e. \( \hat{\Theta}^\top x < 0 \), predict 0
Logistic regression
Derivation

Interpretation of $f(x; \Theta)$: it models the conditional probability $Pr(y = 1|x; \Theta)$

$$f(x; \Theta) = Pr(y = 1|x; \Theta)$$

1. categorical attribute $Y = \{0, 1\}$

2. $y = \theta_0 + \theta_1 x_1 \cdots + \theta_m x_m$, see above $\rightarrow$ model $Pr(Y = y|x)$, e.g. $Pr(Y = 1|x)$

3. $Pr(Y = 1|x) = \theta_0 + \theta_1 x_1 \cdots + \theta_m x_m$, see above

4. Model odds($Pr(Y = 1|x)$) = $\frac{Pr(Y=1|x)}{Pr(Y=0|x)} = \frac{Pr(Y=1|x)}{1-Pr(Y=1|x)} \in (0, +\infty)$
Odds, odds ratio

odds = Pr(success)/ Pr(failure)

Example: Titanic data set

```r
> d <- read.csv("train.csv")
> attach(d)
> table(Sex, Survived)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>female</td>
<td>81</td>
<td>233</td>
</tr>
<tr>
<td>male</td>
<td>468</td>
<td>109</td>
</tr>
</tbody>
</table>
```

• the odds of surviving for male:
  \[
  \frac{Pr(\text{Survived} = 1 | \text{Sex} = \text{male})}{Pr(\text{Survived} = 0 | \text{Sex} = \text{male})} = \frac{109}{486} = 0.23
  \]

• the odds of surviving for female:
  \[
  \frac{Pr(\text{Survived} = 1 | \text{Sex} = \text{female})}{Pr(\text{Survived} = 0 | \text{Sex} = \text{female})} = \frac{233}{81} = 2.88
  \]

• the ratio of the odds for female to the odds for male 2.88/0.23 = 12.52
5. Transform $\langle 0, +\infty \rangle$ to $(-\infty, +\infty)$: model

$$\text{logit}(\Pr(Y = 1|x)) = \ln(\text{odds}(\Pr(Y = 1|x))) = \ln\left(\frac{\Pr(Y = 1|x)}{1 - \Pr(Y = 1|x)}\right)$$

6. Use linear regression

$$\ln\left(\frac{\Pr(Y = 1|x)}{1 - \Pr(Y = 1|x)}\right) = \theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m$$

i.e.,

$$\Pr(Y = 1|x) = \frac{1}{1 + e^{-\theta_0 - \theta_1 x_1 - \cdots - \theta_m x_m}}$$

\[
f(x_i; \Theta) = \Pr(Y_i = 1|x_i; \Theta) = \frac{1}{1 + e^{-\Theta^\top x_i}}
\]
Parameter interpretation
Binary features

• Use female = \{1, 0\} instead of Sex = \{female, male\}

• in Linear regression \( y = \theta_0 + \theta_1 \times \text{female} \)
  • \( \theta_0 \) is the average \( y \) for male
  • \( \theta_0 + \theta_1 \) is the average \( y \) for female
  • \( \theta_1 \) is the average difference in \( y \) between female and male

• in Logistic regression \( p = \Pr(Survive = 1|x, \Theta) \), \( \ln \frac{p}{1-p} = \theta_0 + \theta_1 \times \text{female} \)
  • If female == 0
    • \( p = p_1 \rightarrow \ln(\frac{p_1}{1-p_1}) = \theta_0 \rightarrow \frac{p_1}{1-p_1} = e^{\theta_0} \)
    • the intercept \( \theta_0 \) is the log odds for men
  • If female == 1
    • \( p = p_2 \rightarrow \frac{p_2}{1-p_2} = e^{\theta_0+\theta_1} \)
    • odds ratio = \( \frac{p_2}{1-p_2} / \frac{p_1}{1-p_1} = e^{\theta_1} \)
    • the parameter \( \theta_1 \) is the log odds ratio between female and male
Parameter interpretation
Numerical features

- $\theta_i$ gives an average change in $\text{logit}(f(x))$ with one-unit change in $A_i$ holding all other features fixed
Parameter estimates

• **Loss function**

\[ L(\Theta) = - \sum_{i=1}^{n} y_i \log P(y_i|x_i; \Theta) + (1 - y_i) \log(1 - P(y_i|x_i; \Theta)) \]

See Maximum Likelihood Principle for derivation of this loss function.

• **Optimization problem**

\[ \Theta^* = \arg \min_{\Theta} L(\Theta) \]
Parameter estimates

\[ L(\Theta) = - \sum_{i=1}^{n} y_i \log P(y_i | x_i; \Theta) + (1 - y_i) \log (1 - P(y_i | x_i; \Theta)) \]
Parameter estimates

\[ L(\Theta) = - \sum_{i=1}^{n} y_i \log P(y_i|\mathbf{x}_i; \Theta) + (1 - y_i) \log(1 - P(y_i|\mathbf{x}_i; \Theta)) \]
repeat until convergence {

\[ \Theta^{K+1} := \Theta^K - \alpha \nabla f(\Theta^K) \]

}\

- \( \alpha \) is a positive step-size hyperparameter

l.e. simultaneously update \( \theta_j, j = 1, \ldots, m \)

\[ \theta_j^{K+1} := \theta_j^K - \alpha \frac{1}{n} \sum_{i=1}^{n} (f(x_i; \Theta^K) - y_i) x_{ij} \]
Non-linear decision boundary
For hyperplane: \( f(\mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m) \)

Let \( f(\mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2) \) (a higher degree polynomial)

Assume \( \theta_0 = -1, \theta_1 = 0, \theta_2 = 0, \theta_3 = 1, \theta_4 = 1 \)

Predict \( y = 1 \) if \(-1 + x_1^2 + x_2^2 \geq 0\), i.e. \( x_1^2 + x_2^2 \geq 1 \)
Classification of $x$ by $\hat{f}^*$

1. Project $x$ onto $\hat{\Theta}^*$ to convert it into a real number $z$ in the range $(-\infty, +\infty)$
   - i.e. $z = \hat{\Theta}^* \top x$

2. Map $z$ to the range $\langle 0, 1 \rangle$ using the sigmoid function $g(z) = 1/(1 + e^{-z})$

3. Classify $x$ using a classification rule
Multi-class classification

\[ |Y| = N, \ N \geq 3 \]

- **One-to-all**
  - train \( N \) binary classifiers \( f_k \) for the pair \( k \)-th class and \( \{1, \cdots, N\} \setminus \{k\} \) classes
  - classify \( x \) into the class \( k^* = \text{argmax}_k f_k(x) \)

- **One-to-one**
  - train \( \binom{N}{2} \) binary classifiers \( f_i \) for each pair of classes
  - classify \( x \) into the class \( k^* = \max_{k=1,\ldots,N} \sum_{i=1}^{\binom{N}{2}} \delta(f_i(x) = k) \)
Logistic regression
Multi-class classification

One-to-all
Evaluation of binary classifiers

Confusion matrix

Confusion matrix is a square matrix indexed by all possible target class values.

Task: Assign the correct sense of the word *line* in a sentence.

** Comparing the predicted values with the true senses **

<table>
<thead>
<tr>
<th>Truth</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>cord</td>
<td>cord</td>
</tr>
<tr>
<td>cord</td>
<td>268</td>
</tr>
<tr>
<td>division</td>
<td>3</td>
</tr>
<tr>
<td>formation</td>
<td>13</td>
</tr>
<tr>
<td>phone</td>
<td>25</td>
</tr>
<tr>
<td>product</td>
<td>51</td>
</tr>
<tr>
<td>text</td>
<td>12</td>
</tr>
</tbody>
</table>

Correctly predicted examples are displayed on the diagonal.
In binary classification tasks examples are sometimes regarded as divided into two disjoint subsets:

- **positive examples** – “to be retrieved” (ones)
- **negative examples** – “not to be retrieved” (zeros)

### Example confusion matrix for binary classification

```r
> table(test.true, test.pred)
prediction       0  1
   true          0 580 69
   1            37 144
```

---

**Evaluation of binary classifiers**

**Confusion matrix**
Evaluation of binary classifiers
Confusion matrix

<table>
<thead>
<tr>
<th>True class</th>
<th>Predicted class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>True Positive (TP)</td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td>Negative</td>
<td>False Positive (FP)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

Explanation

- ‘Trues’ are examples correctly classified
- ‘Falses’ are examples incorrectly classified
- ‘Positives’ were predicted as positives (correctly or incorrectly)
- ‘Negatives’ were predicted as negatives (correctly or incorrectly)
Proportion of correctly predicted test examples

- **Known positives**
- **Predicted positives**
- **False negatives**
- **True positives**
- **False positives**
Basic performance measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>$\frac{TP}{(TP+FP)}$</td>
</tr>
<tr>
<td>Recall/Sensitivity</td>
<td>$\frac{TP}{(TP+FN)}$</td>
</tr>
<tr>
<td>Specificity</td>
<td>$\frac{TN}{(TN+FP)}$</td>
</tr>
<tr>
<td>1-Specificity (FPR)</td>
<td>$\frac{FP}{(TN+FP)}$</td>
</tr>
<tr>
<td>Accuracy</td>
<td>$\frac{(TP+TN)}{(TP+FP+TN+FN)}$</td>
</tr>
</tbody>
</table>

Very often you need to **combine both good precision and good recall**. Then you usually use **balanced F-score**, so called **F-measure**

$$F = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
Sensitivity vs. Specificity

Perfect classifier – no error

Reality – errors
Sensitivity vs. Specificity

100% sensitive classifier

100% specific classifier
Sensitivity vs. specificity

- 100% sensitivity
- 100% specificity
- Threshold

True Negative  False Negative  False Positive  True Positive
An **ROC curve** plots True Positive Rate vs. False Positive Rate at different classification thresholds (see p. 6).
Area Under ROC (= AUC) is a measure of how good is a distinguishing property of classifier.
Evaluation of binary classifiers
ROC & AUC

- Graphs showing the evaluation of binary classifiers using ROC and AUC.
Evaluation of binary classifiers
ROC & AUC
Summary of Examination Requirements

- Decision boundary, hyperplane, classification rule
- Logistic regression, sigmoid function, probabilistic formulation
- Multi-class classification
- Confusion matrix for binary classification
- Basic performance measures, ROC curve, AUC