Outline

• Evaluation of binary classification (cntnd) – ROC curve
• Model complexity, overfitting, bias and variance
• Regularization – Ridge regression, Lasso
  • Linear regression
  • Logistic regression
Evaluation of binary classifiers
Sensitivity vs. specificity

Confusion matrix

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

<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/TPR</td>
<td>$\frac{TP}{TP+FN} = \frac{TP}{P}$</td>
</tr>
<tr>
<td>Specificity</td>
<td>$\frac{TN}{TN+FP}$</td>
</tr>
<tr>
<td>1-Specificity/FPR</td>
<td>$\frac{FP}{TN+FP} = \frac{FP}{N}$</td>
</tr>
<tr>
<td>Accuracy</td>
<td>$\frac{(TP+TN)}{(TP+FP+TN+FN)}$</td>
</tr>
</tbody>
</table>
Seven training examples

**Classifier’s output** – examples in black circle are positives, other examples are negatives

**Perfect classifier** – no error
Evaluation of binary classifiers
Sensitivity vs. specificity

**Reality** – e.g. 2 miclassified examples
sensitivity $= \frac{2}{3}$, specificity $= \frac{3}{4}$
**Reality** – e.g. 2 misclassified examples
sensitivity = 1, specificity = 1/2
Evaluation of binary classifiers
Sensitivity vs. specificity

**Reality** – e.g. 1 misclassified example
sensitivity = 2/3, specificity = 1
Evaluation of binary classifiers
Sensitivity vs. specificity

Sensitivity (TPR) vs. specificity (TNR)
– as the sensitivity increases, the specificity decreases and vice versa
Evaluation of binary classifiers

ROC curve

An **ROC curve** plots True Positive Rate vs. False Positive Rate at different classification thresholds where $\text{FPR} = 1 - \text{TNR} = \frac{\text{FP}}{\text{N}} = \frac{\text{FP}}{\text{FP} + \text{TN}}$.
Area Under ROC (\(=\) AUC) is a measure of how good is a distinguishing property of classifier.
Evaluation of binary classifiers
ROC & AUC

Curves closer to the top-left corner indicate a better performance.
Model complexity

No universal definition

Heading for the regularization . . . model complexity is the number of hypothesis parameters

$$\Theta = \langle \theta_0, \ldots, \theta_m \rangle$$
Model complexity

Finding a model that minimizes generalization error
... is one of central goals of the machine learning process
Model complexity

Complexity of decision boundary for classification
Bias and variance

1. Select a machine learning algorithm
2. Get $k$ different training sets
3. Get $k$ predictors

- **Bias** measures error that originates from the learning algorithm
  – how far off in general the predictions by $k$ predictors are from the true output value

- **Variance** measures error that originates from the training data
  – how much the predictions for a test instance vary between $k$ predictors
Bias and variance
Bias and variance

**Generalization error** $\text{error}_D(\hat{f})$ measures how well a hypothesis $\hat{f}$ ($f$ is a true target function) generalizes beyond the used training data set, to unseen data with distribution $D$. Usually it is defined as follows

- for **regression**: $\text{error}_D(\hat{f}) = E[\hat{y}_i - y_i]^2$
- for **classification**: $\text{error}_D(\hat{f}) = \Pr(\hat{y}_i \neq y_i)$

**Decomposition of error** $\text{error}_D(\hat{f})$

$$\text{error}_D(\hat{f}) = \text{Bias}^2 + \text{Variance}$$

i.e.,

$$(E[\hat{f}(x)] - f(x))^2 + E[\hat{f}(x) - E[\hat{f}(x)]]^2$$

where $\hat{f}(x)$ is a predicted value, $E[\hat{f}(x)]$ is average predicted value
Bias and variance

- underfitting = high bias
- overfitting = high variance
Bias and variance

overfitting
high variance

underfitting
high bias

good balance
low variance, low bias
Bias and variance
k-Nearest Neighbor

- $k \uparrow$ → smoother decision boundary $\rightarrow \downarrow$ variance and $\uparrow$ bias
- $k \downarrow$ $\rightarrow \uparrow$ variance and $\downarrow$ bias
Bias and variance
k-Nearest Neighbor

5-nearest neighbour

15-nearest neighbour
Prevent overfitting

We want a model in between which is

- powerful enough to model the underlying structure of data
- not so powerful to model the structure of the training data

Let’s prevent overfitting by **complexity regularization**, a technique that regularizes the parameter estimates, or equivalently, shrinks the parameter estimates towards zero.
Regularization

A machine learning algorithm estimates hypothesis parameters \( \Theta = \langle \theta_0, \theta_1, \ldots, \theta_m \rangle \) using \( \Theta^* \) that minimizes loss function \( L \) for training data \( Data = \{ \langle x_i, y_i \rangle \}, x_i = \langle x_{1i}, \ldots, x_{mi} \rangle, y_i \in Y \} \)

\[ \Theta^* = \text{argmin}_\Theta L(\Theta) \]

Regularization

\[ \Theta^*_R = \text{argmin}_\Theta L(\Theta) + \lambda \cdot \text{penalty}(\Theta), \text{ where } \lambda \geq 0 \text{ is a tuning parameter} \]

Infact, the penalty is applied to \( \theta_1, \ldots, \theta_m \), but not to \( \theta_0 \) since the goal is to regularize the estimated association between each feature and the target value.
Regularization
Motivation

\[ \theta_0 + \theta_{age} x \]

- \( \theta_0 \): intercept
- \( \theta_{age} \): coefficient for age

- **Training examples**
- **Test examples**

Regularize \( \theta_{age} \)

**Diagram:**
- X-axis: Age
- Y-axis: Salary
- Unit change

NPFL054, 2019 Hladká & Holub Lecture 9, page 24/38
Regularization
Ridge regression

penalty(Θ) = θ_1^2 + \cdots + θ_m^2 = ℓ_2 \text{ norm}^2

- Let $\theta^*_\lambda, \ldots, \theta^*_m$ be ridge regression parameter estimates for a particular value of $\lambda$
- Let $\theta^*_1, \ldots, \theta^*_m$ be unregularized parameter estimates
- $0 \leq \frac{\theta^2_1 + \cdots + \theta^2_m}{\theta^2_1 + \cdots + \theta^2_m} \leq 1$
- **When** $\lambda = 0$, **then** $\theta^*_\lambda_i = \theta^*_i$ for $i = 1, \ldots, m$
- **When** $\lambda$ is extremely large, **then** $\theta^*_\lambda_i$ is very small for $i = 1, \ldots, m$
- **When** $\lambda$ between, we are fitting a model and shrinking the parameteres
Ridge regression

The diagram shows the relationship between coefficients and log lambda, with different colors representing various values of lambda. The x-axis represents log lambda, while the y-axis represents coefficients. The distribution of coefficients changes as lambda varies.
Regularization
Lasso

\[ \text{penalty}(\Theta) = |\theta_1| + \cdots + |\theta_m| = \ell_1 \text{ norm} \]

- Let \( \theta^*_\lambda_1, \ldots, \theta^*_\lambda_m \) be lasso regression parameter estimates
- Let \( \theta^*_1, \ldots, \theta^*_m \) be unregularized parameter estimates
- \textbf{When} \( \lambda = 0 \), \textbf{then} \( \theta^*_{\lambda_i} = \theta^*_i \) for \( i = 1, \ldots, m \)
- \textbf{When} \( \lambda \) grows, \textbf{then} the impact of penalty grows
- \textbf{When} \( \lambda \) is extremely large, \textbf{then} \( \theta^*_{\lambda_i} = 0 \) for \( i = 1, \ldots, m \)
Lasso

Log Lambda vs. Coefficients

Coefficients range from 98 to 123, with values decreasing as Log Lambda increases.
Ridge regression and Lasso

Ridge regression shrinks all the parameters but eliminates none, while the Lasso can shrink some parameters to zero.
Elastic net

$$\Theta^*_R = \arg\min_{\Theta} [L(\Theta) + \lambda_1 \cdot (|\theta_1| + \cdots + |\theta_m|) + \lambda_2 \cdot (\theta_1^2 + \cdots + \theta_m^2)]$$

$$0 \leq \lambda_1, \lambda_2 \text{ are tuning parameters}$$

!!! In glmnet package

$$\Theta^*_R = \arg\min_{\Theta} L(\Theta) + \lambda (\alpha(|\theta_1| + \cdots + |\theta_m|) + (1 - \alpha)(\theta_1^2 + \cdots + \theta_m^2))$$

$$0 \leq \alpha \leq 1$$
Regularized linear regression

\[ f(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m \]

\[ L(\Theta) = RSS = \sum_{i=1}^{n}(f(x_i) - y_i)^2 \]

\[ \Theta^*_R = \text{argmin}_{\Theta}[RSS + \lambda \cdot \text{penalty}(\Theta)] \]
Ridge regression
Alternative formulation

\[ \Theta^*_R = \arg\min_{\Theta} \sum_{i=1}^{n} (f(x_i) - y_i)^2 \]

subject to \( \theta_1^2 + \cdots + \theta_m^2 \leq s \)

- the gray circle represents the feasible region for Ridge regression
- the contours represent different RSS values for the unregularized model
Ridge regression
Alternative formulation

- If $s$ is large enough, i.e. $\lambda = 0$, so that the minimum RSS value falls into the region of ridge regression parameter estimates then the alternative formulation yields the least square estimates.
Lasso
Alternative formulation

$$\Theta^*_R = \arg\min_\Theta \sum_{i=1}^n (f(x_i) - y_i)^2$$

subject to $|\theta_1| + \cdots + |\theta_m| \leq s$

- the grey square represents the feasible region of the Lasso
- the contours represent different RSS values for the unregularized model

$$\Theta^*_R = \arg\min_\Theta L(\Theta) + \lambda(|\theta_1| + |\theta_2|)$$
Lasso

Alternative formulation

- If $s$ is large enough, i.e. $\lambda = 0$, so that the minimum RSS value falls into the region of loss parameter estimates then the alternative formulation yields the primary solution.
Elastic net
Regularized logistic regression

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

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

\[ \Theta^*_R = \arg\min_{\Theta} [L(\Theta) + \lambda \cdot \text{penalty}(\Theta)] \]
Summary of Examination Requirements

- Binary classifier using ROC curve (True Positive Rate vs. False Positive Rate)
- Model complexity, generalization error, Bias and variance
- Lasso and Ridge regularization for linear and logistic regression