EE445 Mod2-Lec3: Least Squares Classification

References:

- [VMLS]: Chapter 14

Outline

- What is classification?
- Different error rates
- least squares classifier
- Multi-class classifiers

Binary Classification with Least Squares

Classification

- **M2-L2**: goal was to predict an outcome y from some data x
- **M2-L3** (Classification): the outcome y takes on only a finite number of values, and hence is sometimes called a **label**, or in statistics, a **categorical**.
- Example [Binary Classification]: $y \in \{-1, 1\}$ or $y \in \{0, 1\}$ $y \in \{$'True','False'$\}$
- Relationship: $\hat{y} = f(x)$ where $f : \mathbb{R}^n \rightarrow \{-1, +1\}$
- Classifier: f is called the **classifier** since it takes in vectors $x \in \mathbb{R}^n$ and classifies them as either $f(x) = +1$ or $f(x) = -1$.

Classification Examples

- **Email spam detection**.
  - *Feature vector*: $x \in \mathbb{R}^n$ contains features of an email message like word counts etc.
  - *Outcome*: $y = +1$ if an email represented by feature vector x is `SPAM` and -1 otherwise.
- **Fraud detection.**
  - *Feature vector*: $x \in \mathbb{R}^n$ contains features associated with a credit card user such as average monthly spending, median prices of purchases over last week, etc.
  - *Outcome*: $y = +1$ for `fradulent transactions`, and -1 otherwise.
- **Document Classification.**
  - *Feature vector*: $x \in \mathbb{R}^n$ is a word count (or histogram) vector for a document
  - *Outcome*: $y = +1$ if the document has some topic (e.g., politics) and -1 otherwise

Prediction Errors

- For a given data point (x, y) with predicted outcome $\hat{y} = f(x)$, there are four possibilities:
  1. *True Positive*: $y = +1$ and $\hat{y} = +1$ `[correct prediction]`
  2. *True Negative*: $y = -1$ and $\hat{y} = -1$ `[correct prediction]`
  3. *False Positive*: $y = -1$ and $\hat{y} = +1$ `[incorrect prediction, type I error]`
  4. *False Negative*: $y = +1$ and $\hat{y} = -1$ `[incorrect prediction, type II error]`

Error Rates

Consider data set $(x^(1), \ldots, x^{(N)})$, $(y^{(1)}, \ldots, y^{(N)})$ and model f.

- *Error rate*: total number of errors of both kinds divided by the number of examples $(N_{\mathtt{fp}} + N_{\mathtt{fn}})/N$
- *True positive rate* (sensitivity/recall rate): $N_{\mathtt{tp}}/N_{\mathtt{p}}$ where $N_{\mathtt{p}}$ is the number of outcomes with a true positive label $y = +1$
- *False positive rate* (false alarm rate): $N_{\mathtt{fp}}/N_{\mathtt{n}}$ where $N_{\mathtt{n}}$ is the number of outcomes with a true negative label $y = -1$
- *True negative rate* (specificity): $1 - N_{\mathtt{fp}}/N_{\mathtt{n}} = N_{\mathtt{tn}}/N_{\mathtt{n}}$—i.e., fraction of the data points with $y = -1$ for our model correctly classifies $\hat{y} = -1$
- *Precision*: $N_{\mathtt{tp}}/(N_{\mathtt{tp}} + N_{\mathtt{fp}})$—i.e., fraction of true predictions that are correct.

Confusion Matrix

- *Good classifier*: small (near zero) error rate and false positive rate, and high (near one) true positive rate, true negative rate, and precision.
- Which of these metrics is more important depends on the particular application.

	prediction		
outcome	$\hat{y} = +1$	$\hat{y} = -1$	total
$y = +1$	$N_{\mathtt{tp}}$	$N_{\mathtt{fn}}$	$N_{\mathtt{p}}$
$y = -1$	$N_{\mathtt{fp}}$	$N_{\mathtt{tn}}$	$N_{\mathtt{n}}$
all	$N_{\mathtt{tp}} + N_{\mathtt{fp}}$	$N_{\mathtt{fn}} + N_{\mathtt{tp}}$	N

Least Squares Classifier

- Note: sophisticated methods exist for constructing binary classifiers—e.g., logistic regression and support vector machines—which are beyond this lecture.
- Least squares classifier: this is a simple method that works well in many cases
- Process:
  - do ordinary real-valued least squares fitting of the outcome, ignoring that $y \in \{-1, +1\}$
  - i.e., choose basis functions $f_1, \ldots, f_p$ and solve

$$\min_{\theta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{N} (y^{(i)} - \tilde{f}(x^{(i)}))^2 \Big| \ \tilde{f}(x^{(i)}) = \theta_1 f_1(x^{(i)}) + \cdots + \theta_p f_p(x^{(i)}), \ i = 1, \ldots, N \right\}$$

  - f is the least squares fit over out data
  - **final classifier (least squares classifier)**:

$$f(x) = \text{sign}(\tilde{f}(x)) \quad \text{where} \quad \text{sign}(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases}$$

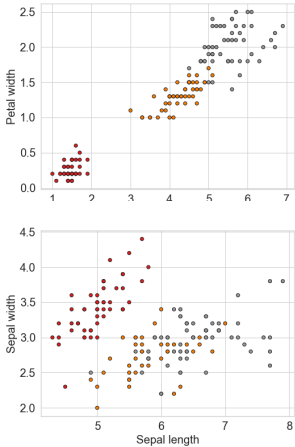
  - e.g., $\tilde{f}(x) = x^\top \beta + v$—i.e., a regression model so that $f(x) = \text{sign}(x^\top \beta + v)$

Intuition for Least Squares Classifier

- The value $\tilde{f}(x)$ is a number "near" $+1$ when $y = +1$ and near -1 when $y = -1$
- Forced to guess one of the two possible outcomes, $\text{sign}(\tilde{f}(x))$ is a good choice—it is the nearest neighbor of $\tilde{f}(x)$ among $\{-1, +1\}$
- $\tilde{f}(x)$ also tells us our confidence in our assignment

Example

- Iris data set: classical ML data set
- Three types of iris:
  - `setosa`, `versicolour`, `virginica`
- Four features:
  - x_1 `sepal length [cm]`, x_2 `sepal width [cm]`
  - x_3 `petal length [cm]`, x_4 `petal width [cm]`
- 50 samples of each type
- Goal: build classifier to detect if iris is `virginica` or not



Iris Data Set Example: Solution

- data matrix

$$A = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(N)} & x_2^{(N)} & x_3^{(N)} & x_4^{(N)} \end{bmatrix}$$

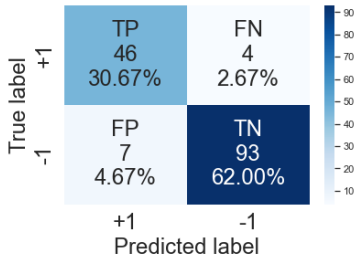
- labels: $y = (\underbrace{-1, \ldots, -1}_{100 \text{ times}}, \underbrace{+1, \ldots, +1}_{50 \text{ times}})$
- $\min_x \|Ax - y\|_2^2$
- **solution**:

$$\hat{y} = \text{sign}(A\hat{x}) \quad \text{where } \hat{x} = (A^\top A)^{-1} A^\top y$$

Iris Data Set Example: Confusion Matrix

- Precision is also known as positive predictive value, and recall is also known as sensitivity in diagnostic binary classification.

- Precision:
$$\frac{N_{\mathtt{tp}}}{N_{\mathtt{tp}} + N_{\mathtt{fp}}}$$
- Accuracy:
$$\frac{N_{\mathtt{tp}} + N_{\mathtt{tn}}}{N}$$
- (F1-score) harmonic mean of precision (P) and recall (R): $\frac{2PR}{P+R}$
  - recall: $N_{\mathtt{tp}}/(N_{\mathtt{tp}} + N_{\mathtt{fn}})$
  - want it near 1



Accuracy=0.927; Precision=0.959
Recall=0.930; F1 Score=0.944

Cross validation

- Just like in the last lecture, we can use cross validation to our least squares classifier.
- see `Mod2-Lec3.ipynb` for example

Receiver Operating Characteristic [ROC] Curves

Modified Classifier with Skewed Decision Boundary

- Modified least squares classifier: skew the decision boundary

$$f(x) = \text{sign}(\tilde{f}(x) - \alpha) = \begin{cases} +1, & \tilde{f}(x) \geq \alpha \\ -1, & \tilde{f}(x) < \alpha \end{cases}$$

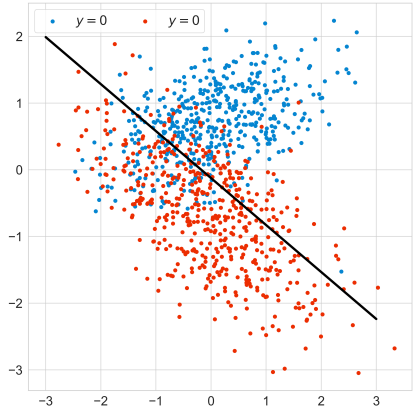
- $\alpha > 0$: the guess $f(x) = +1$ is less frequent $\implies$
  - the numbers in the first column (TP, FP) of the confusion matrix go down, and the numbers in the second column (FN,TN) go up
  - i.e., $\alpha > 0 \implies$ FPR $\downarrow$ which is **good**, yet TPR $\downarrow$ which is **bad**
  - Note: sum of the numbers in each row is always the same
- $\alpha < 0$: the guess $f(x) = +1$ is more frequent
  - $\implies$ TPR $\uparrow$ which is **good**, yet FPR $\uparrow$ which is **bad**
- We choose the decision threshold α depending on how much we care about these different metrics in the application

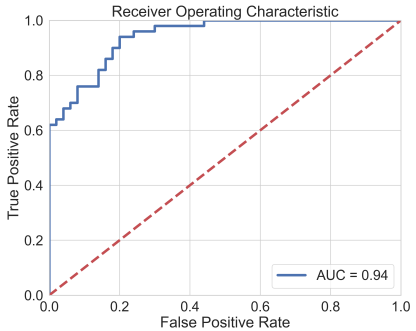
Receiver Operating Characteristic [ROC] Curves

- By sweeping α over a range, we obtain a family of classifiers that vary in their true positive and false positive rates
- Two plots of interest:
  1. the false positive and negative rates, as well as the error rate, as a function of α
  2. [ROC]: true positive rate on the y-axis and false positive rate on the x-axis [More Common to Plot]
- **Cool History Fact:** The name comes from radar systems deployed during World War II, where $y = +1$ means that an enemy vehicle (or ship or airplane) is present, and $\hat{y} = +1$ means that an enemy vehicle is detected.

Example: Mod2-Lec3.ipynb, example 3

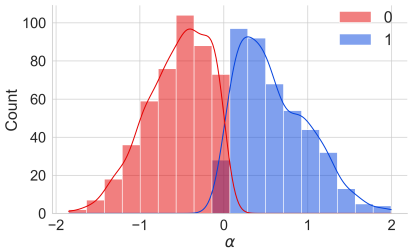
- Randomly generated binary classification problem: $m = 1000$

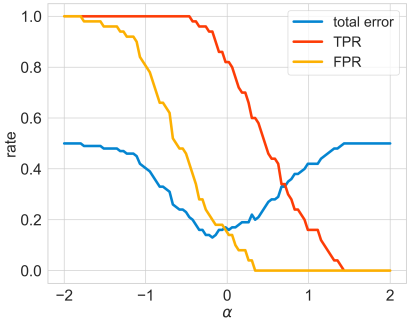




Example: Mod2-Lec3.ipynb, example 3

- Randomly generated binary classification problem: $m = 1000$





Multi-class Classification with Least Squares

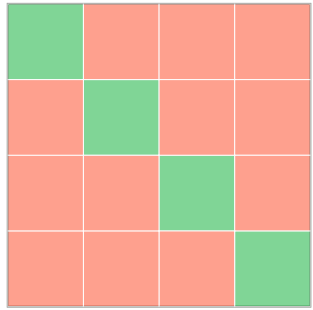
Multi-Class Classifiers

- K Class Classification: # of labels is greater than two ($K > 2$)
  - e.g., Likert scale labels: "Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree"
  - e.g., Iris data set: three types of iris `setosa, versicolour, virginica`
- **Def.** A multi-class classifier is a function $f : \mathbb{R}^n \to \{1, \ldots, K\}$
  - Given a feature vector x, the classifier f returns $f(x) \in \{1, \ldots, K\}$
- Examples
  - Handwritten digit classification (MNIST)
  - Marketing demographic classification—e.g., "college-educated women aged 25–30", "men without college degrees aged 45–55"
  - Disease diagnosis: K possible outcomes for disease
  - Document topic prediction: K possible topics
  - Detection in communications: translate message into K possible signals

Confusion Matrix

- For a multi-class classifier f and a given data point (x, y), with predicted outcome $\hat{y} = f(x)$, there are K^2 possibilities corresponding to all the pairs of values of y, and $\hat{y}$.

- Confusion matrix C: for a given training or test data set with N elements, the numbers of K^2 occurences are arranged into a $K \times K$ matrix where C_{ij} is the number of data points for which $y = i$ and $\hat{y} = j$
- Diagonal of C contains the number of cases for which the prediction is correct



Measures for Prediction Error

- When $K = 2$ we have two types of errors: false positives, false negatives
- More complicated when $K > 2$: From the entries of the confusion matrix we can derive various measures of the accuracy of the predictions

Overall Error Rate

- **Overall Error Rate**: the total number of errors (`np.sum(np.diag(C))`) divided by the data set size (`np.sum(C)`):

$$\frac{1}{N}\sum_{i \neq j} C_{ij} = 1 - \frac{1}{N}\sum_{i} C_{ii}$$ i.e., `err_rate=1-np.sum(np.diag(C))/np.sum(C)`

- This measure implicitly assumes that all errors are equally bad.
- In many applications this is not the case; e.g., some medical misdiagnoses might be worse for a patient than others.

True Label Rate

- **True Label Rate for Class i:**

$$\frac{C_{ii}}{C_i}, \quad \text{where } C_i = \sum_{\ell=1}^{K} C_{i\ell} \text{ i.e., total \# of data points with } y = i$$

- i.e., it is the fraction of data points with label $y = i$ for which we correctly predicted $\hat{y} = i$.

Least Squares Multi-Class Classifier

- The idea behind the least squares Boolean classifier can be extended to handle multi-class classification problems
- *one-vs-others* or *one-vs-all*: for each possible label value, construct a new data set with the Boolean label $+1$ if the label has the given value, and -1 otherwise.
- From these K Boolean classifiers we must create a classifier that chooses one of the K possible labels.
- Select the one with the highest level of confidence (i.e., best least squares fit):

$$f(x) = \arg \max_{k=1,\ldots,K} \tilde{f}_k(x) \quad \text{where } \tilde{f}_k \text{ is the lest squares model for label } k$$

- $\arg\max$: means the index of the largest value among the $\tilde{f}_k(x)$
- $\tilde{f}_k$: real-valued prediction for the Boolean k classifier for class k versus not class k—i.e., it is **NOT** $\text{sign}(\tilde{f}_k(x))$

Example

- consider a multi-class classification problem with 3 labels.
- construct 3 different least squares classifiers: a) 1 versus $\{2$ or $3\}$, b) 2 versus $\{1$ or $3\}$, and c) 3 versus $\{1$ or $2\}$.
- e.g., suppose we have

$$\tilde{f}_1(x) = -0.7, \quad \tilde{f}_2(x) = +0.2, \quad \tilde{f}_3(x) = +0.8$$

- $f(x) = 3$ since $\tilde{f}_3(x)$ is larger than $\tilde{f}_1(x)$ and $\tilde{f}_2(x)$