Machine Learning
Classification

Hiroshi Shimodaira and Hao Tang

22, 26 January 2024

Ver. 1.0a
Topics - you should be able to explain after this week

- Data and data preprocessing
- Features and labels
- Linear classifiers
- Hyperplanes, decision boundaries, and decision regions
- Training of classifiers
- Loss and cost functions
- Logistic regression
- Extension of binary classification to multiclass classification
- Sigmoid and softmax functions
Image data
Pixel image to a feature vector

Turn each cell (pixel) into a number
Unravel into a column vector, a feature vector
⇒ represented digit as point in 64D

\[ \mathbf{x} = (x_1, x_2, \ldots, x_{64})^T, \quad x_i \in [0, 127] \text{ or } x_i \in [0, 1] \]

http://alex.seewald.at/digits/
Image data as a point in a vector space

\[ x = (2, 6, 5)^T \]
Classification of oranges and lemons
A two-dimensional space

Represent each sample as a point \((w, h)\) in a 2D space
Classification

\[ y = -1 \]

\[ y = +1 \]

![Graph showing runtime (sec) vs. source code size (bytes) with two classes: y = -1 (blue dots) and y = +1 (orange crosses).]
Geometry of linear classification

\[ w_1 x_1 + w_2 x_2 + b = 0 \]

\[ \mathbf{w}^\top \mathbf{x} + b = 0 \quad \text{where} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \]

\[ \cdots \text{hyperplane, decision boundary, splitting the space into decision regions} \]

NB: \( \mathbf{w} \) is a normal vector of the hyperplane. \( b \) is not the \( x_2 \) intercept.
Binary classification

\[ h(x) = \begin{cases} 
-1 & \text{if } \mathbf{w}^\top \mathbf{x} + b < 0 \\
+1 & \text{if } \mathbf{w}^\top \mathbf{x} + b \geq 0 
\end{cases} \]  

- The hyperplane \( \mathbf{w}^\top \mathbf{x} + b = 0 \) separates the two classes.
- The function \( h \) labels one class as \(-1\) and the other class as \(+1\).
- The task is called \textit{binary classification}, because there are two classes.
Zero-one loss

\[ \ell_{01}(\hat{y}, y) = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise} \end{cases} = \mathbb{1}_{\hat{y} \neq y} \quad (2) \]

- Think \( \hat{y} \) as the prediction and \( y \) as the label.
- We suffer a loss of 1 if we predict the label wrong.
- In the binary case, \( \ell_{01}(\hat{y}, y) = \mathbb{1}_{\hat{y}y < 0} \).
Classification

- $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$: data set
  - $x_i = [x_{i1} \cdots x_{id}]^\top$, $i = 1, \ldots, n$: input, feature vector, features
  - $y_i$: ground truth, label, gold reference, for $x_i$.

- $f(x) = w^\top x + b$: linear separator, linear predictor
  - $w = [w_1 \cdots w_d]^\top$: weights, weight vector
  - $b \in \mathbb{R}$: bias
  - $\{w, b\}$: parameters $\cdots$ ($\theta = [b \ w^\top]^\top$)

- $h(x) = \text{sgn}(f(x))$, where $\text{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$

NB: This is a non-standard definition of a sign function.
Classification with a linear classifier

(a-1) Linearly separable

(a-2) Linearly non-separable

(b)
Training of Classification

• Given $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, find $\theta$ such that the **zero-one loss**

$$L = \frac{1}{N} \sum_{i=1}^{N} \ell_{01}(h(x_i), y_i)$$  \hspace{1cm} (3)

is minimised. NB: $L$ is called a **cost function**.

• The act of finding the model parameter $\theta$ is called **training**. (We also say “fit the model on the training data” to mean the training)

• In the binary case,

$$L = \frac{1}{N} \sum_{i=1}^{N} \ell_{01}(\text{sgn}(\mathbf{w}^\top x_i + b), y_i) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{y_i(\text{sgn}(\mathbf{w}^\top x_i + b))<0}$$  \hspace{1cm} (4)
• Slightly changing $w$ and $b$ does not change the loss.
• The loss value only changes when the hyperplane flips the sign of a data point, and it either increases by 1 or none at all.
• The loss function (with respect to $w$ and $b$) is like step functions, flat everywhere with discontinuity when the value changes.
• Finding the optimal $w$ and $b$ is inherently combinatorial and hard.
Types of linear classifiers

- Template-based matching with Euclidean distance
- Fisher’s linear discriminant
- Logistic regression
- Support Vector Machine (linear version)
- Perceptron (original version)
- Single-layer neural networks with no hidden nodes
A probabilistic approach

• The range of $f(x) = \mathbf{w}^\top \mathbf{x} + b$: $(-\infty, +\infty)$

• We want to squeeze the range into $[0, 1]$ with a function $g(s)$ so that it can be treated as a probability.

\[ g(f(x)) = g(\mathbf{w}^\top \mathbf{x} + b) \rightarrow p(y = +1 | \mathbf{x}) \]

• A candidate for $g(s)$ is the **logistic (sigmoid) function**:

\[ g(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}} \] (5)

• Logistic regression model:

\[ p(y = +1 | \mathbf{x}, \theta) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x} + b)} \] (6)

\[ p(y = -1 | \mathbf{x}, \theta) = 1 - p(y = +1 | \mathbf{x}) \] (7)
Sigmoid function

\[ \sigma(s) = \frac{1}{1 + \exp(-s)} \]

- When \( s \to \infty \), \( \sigma(s) \to 1 \).
- When \( s \to -\infty \), \( \sigma(s) \to 0 \).
Step function: $u(s) = \begin{cases} 0 & \text{if } s < 0 \\ 1 & \text{if } s \geq 0 \end{cases}$

Sigmoid function vs step function
Interpretation of the logistic regression model

Data distributions $p(x|y)$

Posterior prob. $p(y|x)$

Log odds ratio $\log \frac{p(y=1|x)}{p(y=0|x)}$

Model the log odds ratio with a line: $\log \frac{p(y=1|x)}{p(y=0|x)} = \mathbf{w}^\top \mathbf{x} + b$
Classification with the logistic regression model

For a test input $x$, 

1. calculate the posterior probability with the model.

$$p(y=1|x, \theta) = \frac{1}{1 + \exp(-(w^\top x + b))}$$

2. make a prediction:

$$\hat{y} = \begin{cases} 
+1 & p(y=+1|x, \theta) > \text{threshold}, \\
-1 & p(y=+1|x, \theta) \leq \text{threshold} 
\end{cases} \quad (8)$$

NB: threshold = 0.5 normally – it gives a minimum misclassification rate.
Decision surface - step function version

\[ u(\mathbf{w}^\top \mathbf{x} + b) \]
Decision surface - sigmoid function version

\[ \sigma(\mathbf{w}^\top \mathbf{x} + b) \]
A logistic regression model

\[ p(y = +1|\mathbf{x}, \theta) = \frac{1}{1 + \exp(- (\mathbf{w}^\top \mathbf{x} + b))} \] (9)

\[ p(y = -1|\mathbf{x}, \theta) = 1 - \frac{1}{1 + \exp(- (\mathbf{w}^\top \mathbf{x} + b))} = \frac{\exp(- (\mathbf{w}^\top \mathbf{x} + b))}{1 + \exp(- (\mathbf{w}^\top \mathbf{x} + b))} \] (10)

\[ = \frac{1}{\exp(\mathbf{w}^\top \mathbf{x} + b) + 1} \] (11)

Thus,

\[ p(y|\mathbf{x}, \theta) = \frac{1}{1 + \exp(-y (\mathbf{w}^\top \mathbf{x} + b))} \] (12)
How to train the logistic regression model?

- Use MSE? \( \min_{w,b} \sum_{i=1}^{n} (p(y=+1|x_i, \theta) - y_i)^2 \)  
  NB: the label \( y_i \) needs to be changed to \{0, 1\}.

- Apply the maximum likelihood estimation (MLE):

  Given a data set \( \{(x_1, y_1), \ldots, (x_N, y_N)\} \),
  maximise the likelihood \( L \) of \( w \) and \( b \).

\[
\max_{w,b} L \\
L = \log \prod_{i=1}^{N} p(y_i|x_i, \theta) = \sum_{i=1}^{N} \log \left( \frac{1}{1 + \exp(-y_i(w^\top x_i + b))} \right)
\]

\[
= \sum_{i=1}^{N} - \log \left( 1 + \exp(-y_i(w^\top x_i + b)) \right)
\]
How to find the optimal solutions $w$ and $b$?

- The zero-one loss $\sum_{i=1}^{N} \mathbb{I}y_i(w^\top x_i+b)<0$ is flat, and is hard to optimise.

- The log likelihood of the logistic regression model
  $$L = \sum_{i=1}^{N} -\log(1 + \exp(-y_i(w^\top x_i + b)))$$
  is differentiable.

- However,
  $$\frac{\partial L}{\partial w_i} = 0, \ i = 1, \ldots, d \quad \text{and} \quad \frac{\partial L}{\partial b} = 0 \quad (16)$$

  do not have closed-form solutions.
  \(\rightarrow\) employ gradient ascent.

- We will come back to this in a lecture on optimisation.
Effect of data distributions on decision regions

![Graph showing data and additional points at x1=10]
Effect of data distributions on decision regions (cont.)

1 additional data points

5 additional data points
Classification losses

Suppose we have a labelled data point \((x, y)\).

- Zero-one loss
  \[
  1_{y(w^\top x + b) < 0}
  \]

- Log loss (logistic loss)
  \[
  -\log p(y|x) = \log(1 + \exp(-y(w^\top x + b)))
  \]
Notation caveat

• The log loss notation $- \log p(y|x)$ can be misleading.

• Is $y$ the ground truth or is it a free variable?

• What it really means is $- \log p(y = y^*|x)$ given a pair $(x, y^*)$.

• Or $- \log p(y = y_i|x_i)$ given a pair $(x_i, y_i)$ in a data set.
How to resolve a linearly non-separable case?

Feature transformation

\[
 h(x) = \begin{cases} 
 -1 & \text{if } w^\top x + b < 0 \\
 +1 & \text{if } w^\top x + b \geq 0 
\end{cases} = \text{sgn}(w^\top x + b) \quad (19)
\]

\[
 \downarrow
\]

\[
 h(x) = \begin{cases} 
 -1 & \text{if } w^\top \phi(x) < 0 \\
 +1 & \text{if } w^\top \phi(x) \geq 0 
\end{cases} = \text{sgn}(w^\top \phi(x)) \quad (20)
\]
Feature transformation \textit{(cont.)}

\begin{equation}
    p(y|x, \theta) = \frac{1}{1 + \exp(-y(w^\top x + b))}
\end{equation}

\begin{equation}
    p(y|x, \theta) = \frac{1}{1 + \exp(-y(w^\top \phi(x)))}
\end{equation}
Feature transformation - examples

\((x_1, x_2) \rightarrow (x_1, x_2, x_2^2)\)

\((x_1, x_2) \rightarrow (x_1, x_2, x_1^2, x_2^2)\)
Two-circle example
Two-circle example
What is it meant by linear classifiers?

• A linear classifier is linear in the parameters $w$, **not** in the features.

• A linear classifier can have arbitrary nonlinear features.
Should we consider very complex transformation?

• Not necessarily so.

• Complex models may overfit the training data and may not generalise very well.

• We will come back to this in some lectures later.
How to extend the model to multiclass classification?

• one-vs.-all (one-against-all)

• one-vs.-one
Multiclass classification with logistic regression

Replace the sigmoid with the **softmax function**

- w/o transformation

\[
p(y|x, \theta) = \frac{\exp(w_y^T x)}{\sum_{y' \in \mathcal{Y}} \exp(w_{y'}^T x)} \tag{23}
\]

- w transformation

\[
p(y|x, \theta) = \frac{\exp(w_y^T \phi(x))}{\sum_{y' \in \mathcal{V}} \exp(w_{y'}^T \phi(x))} \tag{24}
\]

NB: we can just use and compare \(w_y^T \phi(x)\) for classification – the denominator is a constant for \(y \in \mathcal{Y}\) and \(\exp()\) is a monotonically increasing function.
Softmax for binary classification

\[ p(y=+1|x, \theta) = \frac{\exp(w_+^1x)}{\exp(w_+^T x) + \exp(w_-^T x)} \]

\[ = \frac{1}{1 + \exp(-(w_+ - w_+^1)^T x)} = \frac{1}{1 + \exp(-w^T x)} \]  \hspace{1cm} (25)

\[ p(y=-1|x, \theta) = \frac{\exp(w_-^1x)}{\exp(w_+^T x) + \exp(w_-^T x)} \]

\[ = \frac{\exp(-(w_+ - w_-^1)^T x)}{1 + \exp(-(w_+ - w_-^1)^T x)} = \frac{\exp(-w^T x)}{1 + \exp(-w^T x)} \] \hspace{1cm} (26)

where \( w = w_+ - w_- \).

\[ \rightarrow \text{the same as the sigmoid.} \]
Summary

• Log loss in the binary case

\[
\sum_{i=1}^{N} \log \left( 1 + \exp(-y_i \mathbf{w}^\top \phi(x_i)) \right)
\] (29)

• Log loss in the multiclass case

\[
\sum_{i=1}^{N} -\mathbf{w}_{y_i}^\top \phi(x_i) + \log \left( \sum_{y' \in \mathcal{Y}} \exp(\mathbf{w}_{y'}^\top \phi(x_i)) \right)
\] (30)
<table>
<thead>
<tr>
<th><strong>Summary (cont.)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>binary classification</strong></td>
</tr>
<tr>
<td>$h(x) = \begin{cases} -1 &amp; \text{if } w^T \phi(x) &lt; 0 \ +1 &amp; \text{if } w^T \phi(x) \geq 0 \end{cases}$</td>
</tr>
<tr>
<td>$p(y</td>
</tr>
</tbody>
</table>
Appendix – softmax

\[
\text{softmax} \left( \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \right) = \begin{bmatrix} \frac{\exp(a_1)}{\sum_{i=1}^{n} \exp(a_i)} \\ \frac{\exp(a_2)}{\sum_{i=1}^{n} \exp(a_i)} \\ \vdots \\ \frac{\exp(a_n)}{\sum_{i=1}^{n} \exp(a_i)} \end{bmatrix}
\] (31)
Appendix – softmax (cont.)

- $\text{softmax}([1\ 2\ 3]^\top) = [0.09\ 0.24\ 0.67]^\top$

- $\text{softmax}([100\ 200\ 300]^\top) = [10^{-87}\ 10^{-44}\ 1.0]^\top$

- Softmax always returns a probability distribution.

- When the dynamic range of the input is large, the result of softmax becomes “sharp.”
Appendix – softmax (cont.)

• Claim: \( \frac{\exp(a_{\text{max}}/\tau)}{\sum_{i=1}^{n} \exp(a_i/\tau)} \rightarrow 1 \) when \( \tau \rightarrow 0 \).

• That means \( \frac{\exp(a_j/\tau)}{\sum_{i=1}^{n} \exp(a_i/\tau)} \rightarrow 0 \) when \( \tau \rightarrow 0 \) for any \( a_j \) that is not the max.

• We have

\[
\frac{\exp(a_m/\tau)}{\sum_{i=1}^{n} \exp(a_i/\tau)} = \frac{\exp(a_m/\tau)}{\exp(a_m/\tau) + \sum_{i \neq m} \exp(a_i/\tau)}
\]

(32)

\[
= \frac{1}{1 + \sum_{i \neq m} \exp((a_i - a_m)/\tau)} \rightarrow 1
\]

(33)

when \( \tau \rightarrow 0 \) because \( a_m \) is the largest and \( a_i - a_m < 0 \).
Quizzes

1. Consider two column vectors such that \( \mathbf{a} = (1, 2, 3)^T \) and \( \mathbf{b} = (-3, 3, -1)^T \).
   - Find \( \mathbf{a} + \mathbf{b} \).
   - Find \( \mathbf{a} - \mathbf{b} \).
   - Find \( \|\mathbf{a}\|, \|\mathbf{b}\| \), and \( \|\mathbf{a} - \mathbf{b}\| \).
   - Find \( \mathbf{a}^T \mathbf{b} \).
   - Find \( \mathbf{a}^T \mathbf{b} \).
   - What is the geometric relationship between \( \mathbf{a} \) and \( \mathbf{b} \)?

2. Considering a classification problem of two classes, whose discriminant function takes the form, \( y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \).
   - Show that the decision boundary is a straight line when \( D = 2 \).
   - Show that the weight vector \( \mathbf{w} \) is a normal vector to the decision boundary.

3. Derive a formula for the Euclidean distance between the origin \((0, 0)\) and a line \( y = ax + b \), where \( a \) and \( b \) are arbitrary constants.
4. Considering a linear classifier of binary classification in a two-dimensional vector space, such that the points \((-2, -3)\) and \((4, 1)\) are on the decision boundary, and the point \((2, -3)\) lies in the \(-1\) class region.
   - Find the parameters \((w, b)\) of the classifier.
   - Find the unit normal vector of \(w\).

5. Consider the following logistic regression model:
   \[
p(y = +1|x) = \frac{1}{1 + \exp(-(wx + b))}
   \]
   Plot \(p(y = +1|x)\) for each of the following cases, where you use a fixed plotting range or show all the plots on a single graph for comparison, and report your findings.
   - \(w = 1, b = 0\)
   - \(w = 1, b = 1\)
   - \(w = -1, b = 1\)
   - \(w = 0.5, b = 1\)
   - \(w = 2, b = 1\)
6. Consider the logistic sigmoid function.

\[ \sigma(x) = \frac{1}{1 + \exp(-x)} \]

- Based on the graph of \( \sigma(x) \), make an educated guess about the shape of the derivative \( \sigma'(x) \) without performing any calculations and illustrate it by hand.
- Find the derivative of \( \sigma(x) \).
- Plot the derivative on a graph.