Logistic regression model

- Defines a linear decision boundary
- Discriminant functions:
  \[ g_1(x) = g(w^T x) \quad \text{and} \quad g_0(x) = 1 - g(w^T x) \]
- where \( g(z) = 1/(1 + e^{-z}) \) - is a logistic function

\[ f(x, w) = g_1(w^T x) = g(w^T x) \]
Logistic regression model. Decision boundary

- **LR defines a linear decision boundary**

  **Example:** 2 classes (blue and red points)

Logistic regression: parameter learning

- **Log likelihood**

  \[
  l(D, w) = \sum_{i=1}^{n} y_i \log \mu_i + (1 - y_i) \log (1 - \mu_i)
  \]

- **Derivatives of the loglikelihood**

  \[
  \frac{\partial}{\partial w_j} l(D, w) = \sum_{i=1}^{n} -x_{i,j} (y_i - g(z_i))
  \]

  Nonlinear in weights !!

  \[
  \nabla_w l(D, w) = \sum_{i=1}^{n} -x_i (y_i - g(w^T x_i)) = \sum_{i=1}^{n} -x_i (y_i - f(w, x_i))
  \]

- **Gradient descent:**

  \[
  w^{(k)} \leftarrow w^{(k-1)} - \alpha(k) \nabla_w [l(D, w)]_{w^{(k-1)}}
  \]

  \[
  w^{(k)} \leftarrow w^{(k-1)} + \alpha(k) \sum_{i=1}^{n} [y_i - f(w^{(k-1)}, x_i)] x_i
  \]
Generative approach to classification

Idea:
1. Represent and learn the distribution \( p(x, y) \)
2. Use it to define probabilistic discriminant functions

E.g. \( g_0(x) = p(y = 0 \mid x) \quad g_1(x) = p(y = 1 \mid x) \)

Typical model \( p(x, y) = p(x \mid y) p(y) \)
- \( p(x \mid y) = \text{Class-conditional distributions (densities)} \)
  - binary classification: two class-conditional distributions
    \[
    p(x \mid y = 0) \quad p(x \mid y = 1)
    \]
- \( p(y) = \text{Priors on classes} \) - probability of class \( y \)
  - binary classification: Bernoulli distribution
    \[
    p(y = 0) + p(y = 1) = 1
    \]

Quadratic discriminant analysis (QDA)

Model:
- Class-conditional distributions
  - multivariate normal distributions
    \[
    x \sim N(\mu_0, \Sigma_0) \quad \text{for} \quad y = 0 \\
    x \sim N(\mu_1, \Sigma_1) \quad \text{for} \quad y = 1
    \]
  - Multivariate normal \( x \sim N(\mu, \Sigma) \)
    \[
    p(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right]
    \]
- Priors on classes (class 0,1)
  - Bernoulli distribution
    \[
    p(y, \theta) = \theta^y (1 - \theta)^{1-y} \quad y \in \{0,1\}
    \]
QDA

2 Gaussian class-conditional densities
QDA: Making class decision

Basically we need to design discriminant functions

Two possible choices:

- **Likelihood of data** – choose the class (Gaussian) that explains the input data \( (x) \) better (likelihood of the data)
  \[
  \frac{p(x \mid \mu_1, \Sigma_1)}{g_1(x)} > \frac{p(x \mid \mu_2, \Sigma_2)}{g_2(x)} \quad \text{then} \quad y=1 \\
  \text{else} \quad y=0
  \]

- **Posterior of a class** – choose the class with better posterior probability
  \[
  p(y = 1 \mid x) > p(y = 0 \mid x) \quad \text{then} \quad y=1 \\
  \text{else} \quad y=0
  \]

  \[
  p(y = 1 \mid x) = \frac{p(x \mid \mu_1, \Sigma_1) p(y = 1)}{p(x \mid \mu_0, \Sigma_2) p(y = 0) + p(x \mid \mu_1, \Sigma_1) p(y = 1)}
  \]
QDA: Quadratic decision boundary

Linear discriminant analysis (LDA)
- When covariances are the same
  \[ \mathbf{x} \sim N(\mu_0, \Sigma), \ y = 0 \]
  \[ \mathbf{x} \sim N(\mu_1, \Sigma), \ y = 1 \]
LDA: Linear decision boundary

Contours of class-conditional densities

LDA: linear decision boundary

Decision boundary
**Generative classification models**

**Idea:**
1. **Represent and learn the distribution** \( p(x, y) \)
2. **Use it to define probabilistic discriminant functions**

**E.g.** \( g_o(x) = p(y = 0 \mid x) \quad g_1(x) = p(y = 1 \mid x) \)

**Typical model** \( p(x, y) = p(x \mid y) p(y) \)

- **\( p(x \mid y) = \text{Class-conditional distributions (densities)} \)**
  - Binary classification: two class-conditional distributions
    \( p(x \mid y = 0) \quad p(x \mid y = 1) \)
- **\( p(y) = \text{Priors on classes} \) - probability of class \( y \)**
  - Binary classification: Bernoulli distribution
    \( p(y = 0) + p(y = 1) = 1 \)

---

**Naïve Bayes classifier**

- **A generative classifier model with an additional simplifying assumption:**
  - All input attributes are conditionally independent of each other given the class.

So we have:

\[
p(x, y) = p(x \mid y) p(y)
\]

\[
p(x \mid y) = \prod_{i=1}^{d} p(x_i \mid y)
\]
Learning parameters of the model

Much simpler density estimation problems
• We need to learn:
  \[ p(x \mid y = 0) \text{ and } p(x \mid y = 1) \text{ and } p(y) \]
• Because of the assumption of the conditional independence we need to learn:
  for every variable \( i \): \( p(x_i \mid y = 0) \text{ and } p(x_i \mid y = 1) \)
• Much easier if the number of input attributes is large
• Also, the model gives us a flexibility to represent input attributes different of different forms !!!
• E.g. one attribute can be modeled using the Bernoulli, the other as Gaussian density, or as a Poisson distribution

Making a class decision for the Naïve Bayes

Discriminant functions
• **Likelihood of data** – choose the class that explains the input data \( (x) \) better (likelihood of the data)
  \[
  g_1(x) = \prod_{j=1}^{d} p(x_j \mid \Theta_{1,j}) > \prod_{i=1}^{d} p(x_i \mid \Theta_{2,i}) \quad \text{then } y=1 \\
  g_0(x) = \prod_{j=1}^{d} p(x_j \mid \Theta_{1,j}) \quad \text{else } y=0
  \]
• **Posterior of a class** – choose the class with better posterior probability \( p(y = 1 \mid x) > p(y = 0 \mid x) \)
  \[
  p(y = 1 \mid x) = \frac{\left( \prod_{i=1}^{d} p(x_i \mid \Theta_{1,i}) \right)p(y = 1)}{\left( \prod_{i=1}^{d} p(x_i \mid \Theta_{1,i}) \right)p(y = 0) + \left( \prod_{i=1}^{d} p(x_i \mid \Theta_{2,i}) \right)p(y = 1)}
  \]
Back to logistic regression

- **Two models with linear decision boundaries:**
  - Logistic regression
  - Generative model with 2 Gaussians with the same covariance matrices
    \[
    x \sim N(\mu_0, \Sigma) \quad \text{for} \quad y = 0
    \]
    \[
    x \sim N(\mu_1, \Sigma) \quad \text{for} \quad y = 1
    \]

- **Two models are related!!!**
  - When we have 2 Gaussians with the same covariance matrix the probability of \( y \) given \( x \) has the form of a logistic regression model!!!
    \[
    p(y = 1 \mid x, \mu_0, \mu_1, \Sigma) = g(w^T x)
    \]

When is the logistic regression model correct?

- **Members of the exponential family can be often more naturally described as**
  \[
  f(x \mid \theta, \phi) = h(x, \phi) \exp \left\{ \frac{\theta^T x - A(\theta)}{a(\phi)} \right\}
  \]
  \( \theta \) - A location parameter \( \phi \) - A scale parameter

- **Claim:** A logistic regression is a correct model when class conditional densities are from the same distribution in the exponential family and have the same scale factor \( \phi \)

- **Very powerful result!!!!**
  - We can represent posteriors of many distributions with the same small network
Linear units

**Linear regression**

\[ f(x) = w^T x \]

**Logistic regression**

\[ f(x) = p(y = 1 \mid x, w) = g(w^T x) \]

Gradient update:

\[ w \leftarrow w + \alpha \sum_{i=1}^{n} (y_i - f(x_i))x_i \]

Online:

\[ w \leftarrow w + \alpha (y - f(x))x \]

Gradient-based learning

- The **same simple gradient update rule** derived for both the linear and logistic regression models
- Where the magic comes from?
- Under the log-likelihood measure the function models and the models for the output selection fit together:
  - **Linear model + Gaussian noise**
    \[ y = w^T x + \varepsilon \quad \varepsilon \sim N(0, \sigma^2) \]
  - **Logistic + Bernoulli**
    \[ y = \text{Bernoulli}(\theta) \]
    \[ \theta = p(y = 1 \mid x) = g(w^T x) \]
Generalized linear models (GLIM)

Assumptions:
- The conditional mean (expectation) is:
  \[ \mu = f(\mathbf{w}^T \mathbf{x}) \]
  - Where \( f(.) \) is a response function
- Output \( y \) is characterized by an exponential family distribution with a conditional mean \( \mu \)

Examples:
- Linear model + Gaussian noise
  \[ y = \mathbf{w}^T \mathbf{x} + \epsilon \quad \epsilon \sim N(0, \sigma^2) \]
- Logistic + Bernoulli
  \[ y \approx \text{Bernoulli}(\theta) \]
  \[ \theta = g(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}} \]

Generalized linear models

- A canonical response functions \( f(.) \):
  - encoded in the distribution
  \[ p(\mathbf{x} \mid \theta, \varphi) = h(x, \varphi) \exp \left\{ \frac{\theta^T \mathbf{x} - A(\theta)}{a(\varphi)} \right\} \]
- Leads to a simple gradient form
- Example: Bernoulli distribution
  \[ p(x \mid \mu) = \mu^x (1 - \mu)^{1-x} = \exp \left\{ \log \left( \frac{\mu}{1 - \mu} \right) x + \log(1 - \mu) \right\} \]
  \[ \theta = \log \left( \frac{\mu}{1 - \mu} \right) \quad \mu = \frac{1}{1 + e^{-\theta}} \]
  - Logistic function matches the Bernoulli
When does the logistic regression fail?

- Quadratic decision boundary is needed

When does the logistic regression fail?

- Another example of a non-linear decision boundary
Non-linear extension of logistic regression

- use feature (basis) functions to model nonlinearities
- the same trick as used for the linear regression

Linear regression
\[ f(x) = w_0 + \sum_{j=1}^{m} w_j \phi_j(x) \]

Logistic regression
\[ f(x) = g\left(w_0 + \sum_{j=1}^{m} w_j \phi_j(x)\right) \]

\( \phi_j(x) \) - an arbitrary function of x

Evaluation of classifiers
For any data set we use to test the classification model on we can build a **confusion matrix:**

- Counts of examples with:
  - class label $\omega_j$ that are classified with a label $\alpha_i$

<table>
<thead>
<tr>
<th>predict</th>
<th>$\omega = 1$</th>
<th>$\omega = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha = 1$</td>
<td>140</td>
<td>17</td>
</tr>
<tr>
<td>$\alpha = 0$</td>
<td>20</td>
<td>54</td>
</tr>
</tbody>
</table>

**Evaluation**

For any data set we use to test the model we can build a **confusion matrix:**

<table>
<thead>
<tr>
<th>predict</th>
<th>$\omega = 1$</th>
<th>$\omega = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha = 1$</td>
<td>140</td>
<td>17</td>
</tr>
<tr>
<td>$\alpha = 0$</td>
<td>20</td>
<td>54</td>
</tr>
</tbody>
</table>

Error: ?
Evaluation

For any data set we use to test the model we can build a confusion matrix:

\[
\begin{array}{c|cc}
\text{target} & \omega = 1 & \omega = 0 \\
\hline
\text{predict} & \alpha = 1 & 140 & 17 \\
            & \alpha = 0 & 20 & 54 \\
\end{array}
\]

Error: \(37/231\)
Accuracy: \(194/231\)

Evaluation for binary classification

Entries in the confusion matrix for binary classification have names:

\[
\begin{array}{c|cc}
\text{target} & \omega = 1 & \omega = 0 \\
\hline
\text{predict} & \alpha = 1 & TP & FP \\
            & \alpha = 0 & FN & TN \\
\end{array}
\]

TP: True positive (hit)
FP: False positive (false alarm)
TN: True negative (correct rejection)
FN: False negative (a miss)
Additional statistics

- Sensitivity (recall)
  \[ SENS = \frac{TP}{TP + FN} \]

- Specificity
  \[ SPEC = \frac{TN}{TN + FP} \]

- Positive predictive value (precision)
  \[ PPT = \frac{TP}{TP + FP} \]

- Negative predictive value
  \[ NPV = \frac{TN}{TN + FN} \]

Binary classification: additional statistics

- Confusion matrix

<table>
<thead>
<tr>
<th>predict</th>
<th>1</th>
<th>0</th>
<th>PPV = 140/150</th>
<th>NPV = 180/200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>140</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>180</td>
<td>SENS = 140/160</td>
<td>SPEC = 180/190</td>
</tr>
</tbody>
</table>

Row and column quantities:
- Sensitivity (SENS)
- Specificity (SPEC)
- Positive predictive value (PPV)
- Negative predictive value (NPV)
Classifiers

Project datapoints to one dimensional space:

**Defined for example by:** \( w^T x \) or \( p(y=1|x,w) \)

---

### Binary decisions: Receiver Operating Curves

- **Probabilities:**
  - \( SENS \) \( p(x > x^* \mid x \in \omega_2) \)
  - \( SPEC \) \( p(x < x^* \mid x \in \omega_1) \)
Receiver Operating Characteristic (ROC)

- ROC curve plots:
  - SN = $p(x > x^* \mid x \in \omega_2)$
  - 1-SP = $p(x > x^* \mid x \in \omega_1)$
  - for different $x^*$

SENS
$p(x > x^* \mid x \in \omega_2)$

1-SPEC
$p(x > x^* \mid x \in \omega_1)$

ROC curve

CS 2750 Machine Learning
Receiver operating characteristic

• **ROC**
  – shows the discriminability between the two classes under different decision biases

• **Decision bias**
  – can be changed using different loss function

• **Quality of a classification model:**
  – Area under the ROC
  – Best value 1, worst (no discriminability): 0.5