Questions you should be able to answer after this week

• Ensemble learning (committee method)
• Theoretical background of ensemble learning
• Bagging
• Boosting
• AdaBoost - training algorithm, optimisation problem, loss function
• Applications of boosting
Ensemble learning - committee method

- Single models: logistic regression, SVM, decision tree
- Mixture models: GMM
- Multiple models:
  \[ f_1(x), f_2(x), \ldots, f_M(x) \ldots \text{base models / weak learners} \]
  - Model linear averaging
    \[
y = F(x) = \frac{1}{M} \sum_{i=1}^{M} f_i(x)
    \]
    \[
y = F(x) = \frac{1}{M} \sum_{i=1}^{M} \alpha_i f_i(x)
    \]
  - Training strategies for multiple models
    * Train each model separately
    * Train models one-by-one sequentially
Theoretical background for ensemble learning

A Committee model for a regression task

\[ y = F(x) = \frac{1}{M} \sum_{i=1}^{M} f_i(x) \]

\[ f_i(x) = f(x) + \varepsilon_i(x) \]

\[ E_{\text{avr}} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E} \left[ \{ f_i(x) - f(x) \}^2 \right] = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E} [ \varepsilon_i(x)^2 ] \]

\[ E_{\text{comm}} = \mathbb{E} \left[ \{ F(x) - f(x) \}^2 \right] = \mathbb{E} \left[ \left\{ \frac{1}{M} \sum_{i=1}^{M} f_i(x) - f(x) \right\}^2 \right] = \mathbb{E} \left[ \left\{ \frac{1}{M} \sum_{i=1}^{M} \varepsilon_i(x) \right\}^2 \right] \]

(Assuming that the errors have zero mean and are uncorrelated)

\[ = \frac{1}{M} \left\{ \frac{1}{M} \sum_{i=1}^{M} \mathbb{E} [\varepsilon_i(x)^2] \right\} = \frac{1}{M} E_{\text{avr}} \]
A Committee model of majority voting for a classification task

Base models \( \{f_i(x)\} \) – binary classifiers with classification accuracy \( p \).

\[
y_i = \mathbb{1}(f_i(x) > 0), \quad \text{where } y_i \in \{0, 1\}
\]

Let \( S \) denote the number of votes for class 1,

\[
S = \sum_{i=1}^{M} y_i
\]

Accuracy of the committee model:

\[
Pr(S > M/2) = 1 - B(M/2, M, p)
\]

where \( B(k, n, p) \) is the cdf of the binomial distribution of \( k \) with parameters \( n \) and \( p \).
Background theories for ensemble learning (cont.)

Accuracy of the committee model

NB: strong assumption - each base model makes independent errors
Brief history of committee methods

1992  Stacking (Wolpert)
1994  Bagging (Brieman)
1995  AdaBoost (Freund & Schapire)
      Random Forests (Tin Kam Ho)
1997  Speech recognition with ROVER (Fiscus)
2001  Face detection with AdaBoost (Viola & Jones)

- Train each $f_i(x)$ on a training dataset $D_i$ of size $n$ - sampled from the original data set $D$ uniformly and with replacement.
- Employ the same training algorithm over all $\{f_i(x)\}$

After Bootstrap aggregating (bagging) method of Wikimedia common
Boosting

\[ F(x; \theta) = \sum_{m=1}^{M} \alpha_m f_m(x; \theta_i) \]

\[ \min_{\theta, \alpha} L(F) = \min_{\theta, \alpha} \sum_{i=1}^{n} \ell(y_i, F(x_i; \theta)) \]

- \( f_m \in \{-1, +1\}, \ m = 1, \ldots, M \)
- No closed-form solutions normally
- Fit additive models sequentially

Adapted from boosted_regr_trees.ipynb
**AdaBoost – adaptive boosting** (Freund, Schapire 1995)

Training data set \( \{(x_1, y_1), \ldots, (x_n, y_n)\} \). Classifiers: \( \{f_1(x; \theta_1), \ldots, f_M(x; \theta_M)\} \)

**Step 1** Initialise the weights \( w = (w_1, \ldots, w_n) \), where \( w_i = \frac{1}{n} \).

**Step 2** For \( m = 1 \) to \( M \):

(a) Fit a classifier \( f_m(x) \) to the training data using \( w \).

(b) Compute: \( \text{err}_m = \frac{\sum_{i=1}^{n} w_i \mathbb{1}_{y_i \neq f_m(x_i)}}{\sum_{i=1}^{n} w_i} \)

(c) Compute: \( \alpha_m = \frac{1}{2} \ln \left( \frac{1 - \text{err}_m}{\text{err}_m} \right) \) \ldots cf. logit function

(d) Update the weights: \( w_i \leftarrow w_i e^{\alpha_m \mathbb{1}_{y_i \neq f_m(x_i)}} \), \( i = 1, \ldots, n \)

**Step 3** Output the final model: \( F(x) = \text{sign} \left( \sum_{m=1}^{M} \alpha_m f_m(x) \right) \)
AdaBoost: example

Weak learners: one-level decision trees (i.e. one nodes) – “decision stumps”.

From Figure 14.2 of C.Bishop’s *Pattern Recognition and Machine Learning*
Optimisation problem in AdaBoost

Assume we have trained $f_1(x), \ldots, f_{m-1}(x)$ and obtained $\alpha_1, \ldots, \alpha_{m-1}$.

\[
F_{m-1}(x) = \alpha_1 f_1(x) + \cdots + \alpha_{m-1} f_{m-1}(x)
\]

\[
F_m(x) = F_{m-1}(x) + \alpha_m f_m(x)
\]

We now want to train $f_m(x)$ and estimate $\alpha_m$ with the exponential loss function $L_m$ defined below.

\[
L_m = \sum_{i=1}^{n} e^{-y_i F_m(x_i)} = \sum_{i=1}^{n} e(-y_i F_{m-1}(x_i) - y_i \alpha_m f_m(x))
\]

\[
= \sum_{i=1}^{n} w_{m,i} e(-y_i \alpha m f_m(x)), \quad \text{where } w_{m,i} = e(-y_i F_{m-1}(x_i))
\]

\[
= e^{-\alpha m} \sum_{y_i = f_m(x)} w_{m,i} + e^{\alpha m} \sum_{y_i \neq f_m(x)} w_{m,i}
\]

\[
= (e^{\alpha m} - e^{-\alpha m}) \sum_{y_i \neq f_m(x)} w_{m,i} + e^{-\alpha m} \sum_{i=1}^{n} w_{m,i} \quad \rightarrow \quad \min_{f_m} \sum_{y_i \neq f_m(x)} w_{m,i}
\]
Optimisation problem in AdaBoost (cont.)

We now would like to minimise \( L_m \) with respect to \( \alpha_m \)

\[
\frac{\partial L_m}{\partial \alpha_m} = \frac{\partial}{\partial \alpha_m} \left( e^{-\alpha_m} \sum_{y_i=f_m(x)} w_m,i + e^{\alpha_m} \sum_{y_i \neq f_m(x)} w_m,i \right)
\]

\[
= -e^{-\alpha_m} \sum_{y_i=f_m(x)} w_m,i + e^{\alpha_m} \sum_{y_i \neq f_m(x)} w_m,i = 0
\]

\[
e^{-\alpha_m} \sum_{y_i=f_m(x)} w_m,i = e^{\alpha_m} \sum_{y_i \neq f_m(x)} w_m,i
\]

Taking logarithm yields:

\[
-\alpha_m \ln \left( \sum_{y_i=f_m(x)} w_m,i \right) = \alpha_m \ln \left( \sum_{y_i \neq f_m(x)} w_m,i \right)
\]

Hence

\[
\alpha_m = \frac{1}{2} \ln \left( \frac{\sum_{y_i=f_m(x)} w_m,i}{\sum_{y_i \neq f_m(x)} w_m,i} \right) = \frac{1}{2} \ln \left( \frac{(\sum_{y_i=f_m(x)} w_m,i)/\sum_{i=1}^{n} w_{m,i}}{(\sum_{y_i \neq f_m(x)} w_m,i)/\sum_{i=1}^{n} w_{m,i}} \right) = \frac{1}{2} \ln \left( \frac{1 - \text{err}_m}{\text{err}_m} \right)
\]
Exponential loss in AdaBoost

\[ \ell(y, F(x)) = e^{-yF(x)} \]

- Differentiable and an approximation of the ideal misclassification error function.
- Its sequential minimisation leads to the simple AdaBoost algorithm.
- It penalises large negative values of \( yF(x) \), putting a lot of weight on misclassified samples \( \rightarrow \) very sensitive to outliers (mislabelled examples)
Applications of boosting – face detection

How would you detect a face?

(R. Vaillant, C. Monrocq and Y. LeCun, 1994)

How does album software tag your friends?
Viola–Jones Face detection (2001)

- Face detector consists of linear combination of 'weak' classifiers that utilise five types of primitive features.
- The detector is trained on a training data set of a large number of positive and negative samples.
- Scan the input image with a sub-window (24 x 24 pixels) to detect a face.

Viola & Jones' paper: https://doi.org/10.1023/B:VISI.0000013087.49260.fb
A nice demo: http://vimeo.com/12774628
Other methods and software tools for boosting

• LogitBoost

\[ L_m = \sum_{i=1}^{n} \log \left( 1 + e^{-y_i F_m(x_i)} \right) \]

• Gradient boosting

• Extreme gradient boosting – dominating approach for small, tabular data sets
  ○ XGBoost (eXtreme Gradient Boosting) [Tianqi Chen & Carlos Guestrin, 2016] - a software tool