Regression with GP’s

- Combining models: (Bishop 4.1-4.4)
  - Bayesian model averaging vs. model combination methods
  - **Committees:**
    - Bootstrap aggregation
    - Random subspace methods
    - **Boosting**
    - Decision trees
    - Random forests
Boosting

- Committee consists of multiple base classifiers
- The performance of the committee can be significantly better than that of any of the base classifiers
- **AdaBoost: adaptive boosting**
  - Boosting can give good results even if the base classifiers have a performance that is only slightly better than random
  - Base classifiers are simple models/weak learners
  - Can also be extended to regression.

\[
\text{boosting} \quad \text{vs. bagging (bagging): decreasing variance} \\
\text{boosting: decreasing bias and} \\
(\text{variance})
\]
Boosting

- Base classifiers are trained in sequence
- Note this contrast with other committee methods such as bagging
- Each base classifier is trained using a weighted form of the dataset
- The weighting coefficient associated with each datapoint depends on the performance of previous classifiers.
- Points that are misclassified by one of the base classifiers are given greater weight when used to train the next base classifier in the sequence.
- When all classifiers are trained, predictions are combined through a weighted majority voting scheme.

\[ Y_M(x) = \text{sign} \left( \sum_{m=1}^{M} \alpha_m y_m(x) \right) \]
Boosting: binary classification

- Dataset \( \{(x_n, t_n)\}_{n=1}^{N} \) with \( t_n \in \{-1, +1\} \)

- Each data point has an associated weighting parameter \( w_n \)

- The weights are initialized to \( w_n = 1/N \)

- We assume we have a procedure to train a base classifier \( m \) such that it produces a function \( y_m(x) \in \{-1, +1\} \)

- Adaboost:
  - At each stage a new classifier is trained on weighted dataset
  - Weights for data points that were misclassified by previous classifier are increased
  - When all classifiers are trained, committee is formed by weighted base classifiers
Adaboost

1. Initialize weights: $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$

2. for $m = 1, \ldots, M$:
   
   (a) Fit classifier $y_m(x)$ to minimize $J_m = \sum_{n=1}^{N} w_n^{(m)} I[y_m(x_n) \neq t_n]$

   (b) compute weighted error rates $\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I[y_m(x_n) \neq t_n]}{\sum_{n=1}^{N} w_n^{(m)}}$

   and $\alpha_m = \ln \left( \frac{1 - \epsilon_m}{\epsilon_m} \right)$

   (c) Update weights $w_n^{(m+1)} = w_n^{(m)} \exp\{ \alpha_m I[y_m(x_n) \neq t_n] \}$

3. Make predictions $Y_M(x) = \text{sign} \left( \sum_{m=1}^{M} \alpha_m y_m(x) \right)$
Adaboost

- Prediction $Y_M(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right)$
- Prediction weights $\alpha_m = \ln\left(\frac{1 - \epsilon_m}{\epsilon_m}\right)$
- Weighted error rates $\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I[y_m(x_n) \neq t_n]}{\sum_{n=1}^{N} w_n^{(m)}}$
- Greater weights for more accurate classifiers!
Boosting example: decision stumps

\[ m = 1 \]
\[ -1 \leq x \leq 1 \]
\[ y = \text{sign}(\sum a_m y_m) \]

\[ W_n = \frac{W_n}{\sum_{i=1}^{n} W_n} \]

\[ m = 6 \]
\[ m = 10 \]
\[ m = 150 \]
Interpretation of Adaboost

- Sequential minimization of exponential error function

- Error function $E_m = \sum_{n=1}^{N} \exp\{-t_n f_m(x_n)\}$

- Linear combination of base classifiers $y_l(x)$
  \[ f_m(x) = \frac{1}{2} \sum_{l=1}^{m} \alpha_l y_l(x) \]

- Goal: minimize $E$ with respect to $\{\alpha_l\}$ and parameters of base classifiers $y_l(x)$

- Sequential minimization:
  - Fix parameters of $y_1(x), \ldots, y_{m-1}(x)$ and $\alpha_1, \ldots, \alpha_{m-1}$
  - Minimize $E$ w.r.t. parameters of $y_m(x)$ and $\alpha_m$
Derivation of Adaboost

- Error function

\[
E_m = \sum_{n=1}^{N} \exp\{-t_nf_m(x_n)\} = \sum_{n=1}^{N} \exp\{-t_nf_{m-1}(x_n) - \frac{1}{2}t_n\alpha_my_m(x_n)\}
\]

\[
= \sum_{n=1}^{N} w_n^{(m)} \exp\{-\frac{1}{2}t_n\alpha_my_m(x_n)\}, \quad w_n^{(m)} = \exp(-t_nf_{m-1}(x_n))
\]

- Let \( T_m \) be the set of correctly classified data points \( y_m(x) \) \((t_ny_m(x_n) = 1)\)

- Let \( M_m \) be the set of correctly classified data points \( y_m(x) \) \((t_ny_m(x_n) = -1)\)

- Error function:

\[
E_m = e^{-\alpha_m/2} \sum_{n \in T_m} w_n^{(m)} + e^{+\alpha_m/2} \sum_{n \in M_m} w_n^{(m)}
\]

\[
= \left(e^{\alpha_m/2} - e^{-\alpha_m/2}\right) \sum_{n=1}^{N} w_n^{(m)}I[y_m(x_n) \neq t_n] + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}
\]
Derivation of Adaboost

- Error function

\[ E_m = \left( e^{\frac{\alpha_m}{2}} - e^{-\frac{\alpha_m}{2}} \right) \sum_{n=1}^{N} w_n^{(m)} I[y_m(x_n) \neq t_n] + e^{-\frac{\alpha_m}{2}} \sum_{n=1}^{N} w_n^{(m)} \]

- Minimization w.r.t. \( y_m(x) \) minimizes

\[ J_m = \sum_{n=1}^{N} w_n^{(m)} I[y_m(x_n) \neq t_n] \]

- Minimization w.r.t. \( \alpha_m \): \( \frac{\partial E_m}{\partial \alpha_m} = 0 \)

\[ \alpha_m = \ln \left( \frac{1 - \epsilon_m}{\epsilon_m} \right) \]

\[ \epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I[y_m(x_n) \neq t_n]}{\sum_{n=1}^{N} w_n^{(m)}} \]
Derivation of Adaboost

- After we found $y_m(x)$ and $\alpha_m$, we minimize $E_{m+1}$ w.r.t. $y_{m+1}(x)$ and $\alpha_{m+1}$

$$E_{m+1} = \sum_{n=1}^{N} \exp\{-t_n f_{m+1}(x_n)\}$$

$$= \sum_{n=1}^{N} \exp\{-t_n f_{m-1}(x_n) - \frac{1}{2} t_n \alpha_m y_m(x_n) - \frac{1}{2} t_n \alpha_{m+1} y_{m+1}(x_n)\}$$

$$= \sum_{n=1}^{N} w_n^{(m)} \exp\{-\frac{1}{2} t_n \alpha_m y_m(x_n)\} \exp\{-\frac{1}{2} t_n \alpha_{m+1} y_{m+1}(x_n)\}$$

$$= \sum_{n=1}^{N} w_n^{(m+1)} \exp\{-\frac{1}{2} t_n \alpha_{m+1} y_{m+1}(x_n)\}$$

- Weights are updated: $w_n^{(m+1)} = w_n^{(m)} \exp\{-\frac{1}{2} t_n \alpha_m y_m(x_n)\}$
Derivation of Adaboost

- Weight updates: \( w_{n}^{(m+1)} = w_{n}^{(m)} \exp\left\{-\frac{1}{2} t_{n} \alpha_m y_m(x_n) \right\} \)

- Use: \( t_{n} y_{m}(x_n) = 1 - 2I[y_{m}(x_n) \neq t_n] \)

- Then: \( w_{n}^{(m+1)} = w_{n}^{(m)} \exp\left\{-\alpha_m / 2\right\} \exp\{\alpha_m I[y_{m}(x_n) \neq t_n]\} \)

- Term \( \exp\{-\alpha_m / 2\} \) is independent of \( n \) and can be discarded

- Prediction:

\[
\text{sign} \left( f_m(x) \right) = \text{sign} \left( \frac{1}{2} \sum_{l=1}^{m} \alpha_l y_l(x) \right) = \text{sign} \left( \sum_{l=1}^{m} \alpha_l y_l(x) \right)
\]
Advantages and disadvantages

- Exponential error function makes Adaboost very simple algorithm
- Very sensitive to outliers for which $t_n y_m(x)$ is large negative
- Exponential error function cannot be interpreted as log-likelihood of well-defined probabilistic model
- Doesn’t generalize straightforwardly to $K > 2$