Machine Learning

Ensemble Methods
Bias, Variance, Noise

- Classification errors have different sources
  - Choice of hypothesis space and algorithm
  - Training set
  - Noise in the data
- The expected error sources are often characterized using
  - Bias – Error due to the algorithm and hypothesis
  - Variance – Error due to training data
  - Noise – Noise in the data
Bias, Variance, Noise

- In regression we can show the decomposition of the error into these components
- Assume that a data point \((x, y)\) and training sets \(D\) are drawn randomly from a data distribution
- The squared regression error for a data point \(x\) is

\[
E_{D,y}[(y - h(x))^2] = E_{D,y}[y^2 - 2yh(x) + h(x)^2]
\]

\[
= E_{D,y}[y^2] - 2E_{D,y}[y]E_{D,y}[h(x)] + E_{D,y}[h(x)^2]
\]

\[
= E_{D,y}[y^2] - E[y]^2 + E_{y}[y]^2 - 2E_{y}[y]E_{D}[h(x)] + E_{D}[h(x)]^2
\]

\[
+ E_{D}[h(x)^2] - E_{D}[h(x)]^2
\]
Bias, Variance, Noise

Using the relation \( E[(x-E[x])^2] = E[x^2] - E[x]^2 \) we can transform this:

\[
E_{D,y}[(y-h(x))^2] = E_y[y^2] - E_y[y]^2 \\
+ E_y[y]^2 - 2E_y[y]E_D[h(x)] + E_D[h(x)]^2 \\
+ E_D[h(x)^2] - E_D[h(x)]^2 \\
= E_y[(y-f(x))^2] \quad \text{⇒ Noise} \\
+ (f(x) - E_D[h(x)])^2 \quad \text{⇒ Bias}^2 \\
+ E_D[(h(x) - E_D[h(x)])^2] \quad \text{⇒ Variance}
\]
Bias, Variance, Noise

- Each of the terms explains a different part of the expected error
  - Noise describes how much the target value varies from the true function value
  - Bias describes how much different the average (best) learned hypothesis is from the true function
  - Variance describes how much the learned hypotheses vary with changes in the training data
Linear Regression Example

- 50 datasets with 20 data points each
Linear Regression Example

- Noise

Ditterich and Ng
Linear Regression Example

- Bias

Ditterich and Ng

© Manfred Huber 2015
Linear Regression Example

- Variance
Bias Variance Tradeoff

- Bias can be minimized by appropriate hypothesis spaces (containing the function) and algorithm
  - Requires knowledge of the function or a more complex hypothesis space

- Variance can be minimized by a hypothesis space that requires little data and does not overfit
  - Requires knowledge of function or use of a simpler hypothesis space

- Bias and variance often trade off against each other and are hard to optimize at the same time
Bias Variance Tradeoff

- Typical bias/variance tradeoff:

Hastie, Tibshirani, Friedman
Influencing Bias and/or Variance: Ensemble Methods

- Ensemble methods can change bias and/or variance of an existing classifiers

Classifier 1
Classifier 2
\cdots
Classifier N

Input Features

Class Predictions

Combiner
Bagging

- One way to address variance is by averaging over multiple learned hypotheses
  - Bagging samples the initial \( n \) training examples with replacement to generate \( k \) bootstrap sample sets (usually of the same set size, \( n \))
  - A classifier/regression function is learned on each of the \( k \) training sets
  - The final classification/regression function is determined through majority vote or averaging
Bagging

- The variance of the ensemble classifier/regression function can have lower variance
  - Resulting variance depends on correlation between the hypotheses
    - If all classifiers are the same there is no gain
    - If the classifiers change strongly there will be gain of up to a factor of $1/k$
  - Bootstrap sampling could lead to a small degradation in the learned classifiers/functions

- Bagging mainly helps with methods that are very sensitive to changes in the data set
Bagging

- Bagging generally has no influence on bias
  - Does only average the hypotheses
- Reduces bias for classifiers/regression approaches that are sensitive to training data
  - Averaging the hypotheses can reduce the variance of the final classifier/regression function
- Can we also influence bias using ensemble classification?
Boosting

- Boosting takes a different approach by modifying the training set systematically to allow subsequent classifiers to focus on misclassified examples
  - Originally proposed in the theory of weak learners
    - Even minimally better than random performance can be used to build better learners as an ensemble
  - Only used for classification
  - Whole ensemble formed by weighted voting
Boosting

- Start with equally weighted samples
- Learn a classifier on the weighted data
  - Weight indicates contribution to the error function
  - Compute the error on the training set
  - Increase weight of samples that were misclassified
  - Go back to learning a new classifier until sufficient are learned (often more than 100)
- Perform classification as the weighted sum or predictions of all the models
Boosting

\[
D_1 \{ w_1^{(i)} = \frac{1}{n} \} \\
h_1(x)
\]

\[
D_2 \{ w_2^{(i)} \} \\
h_2(x)
\]

\[
D_3 \{ w_3^{(i)} \} \\
h_3(x)
\]

\[\vdots\]

\[
D_k \{ w_k^{(i)} \} \\
h_k(x)
\]

\[
h(x) = \text{sign} \left( \sum \alpha_i h_i(x) \right)
\]
Boosting Example: AdaBoost

- Assuming classes as 1 and -1, the normalized error of the $m^{th}$ classifier is
  \[ \varepsilon_m = \frac{\sum_{i=1}^{n} \omega_m^{(i)} (1 - \delta_{h(x^{(i)}, y^{(i)})})}{\sum_{i=1}^{n} \omega_m^{(i)}} \]
- Learn the $m^{th}$ classifier and continue if $\varepsilon_m < \frac{1}{2}$
  \[ h_m = \arg\min_h E_m \]
- From this we can compute the classifier weight
  \[ \alpha_m = \frac{1}{2} \ln \left( 1 - \varepsilon_m / \varepsilon_m \right) \]
- And adjust the weights for the data items
  \[ \omega_{m+1} = \omega_m e^{-\alpha_m y^{(i)} h_m(x^{(i)})} \]
Boosting Example: AdaBoost

- The final classifier produces
  \[ h(x) = \text{sign}\left(\sum_{j=1}^{k} \alpha_j h_j(x)\right) \]

- Construction iteratively minimizes an exponential energy function over the misclassifications and with it the misclassification rate
Boosting Example
Boosting can improve bias and variance

- Usually leads to larger improvements than bagging
- Boosting can lead to improvements even with stable classifiers (as opposed to bagging)

But:
- Boosting can hurt performance on very noisy data
- Boosting is also more common to lead to degraded performance than bagging

Instead of weights on data samples, boosting can also use resampling
Ensemble Methods

- Ensemble methods can be used to improve the performance of existing classifiers
  - Bagging improves variance by averaging solutions
  - Boosting can improve bias and variance through weighing of data samples for each classifier to focus on misclassified items

- A range of other ensemble methods have been proposed and built to achieve better performance than a single classifier
  - E.g. Mixture of Experts