Ensemble Methods: Bagging

Introduction

M. R. Hasan
Readings

• Bishop: 14.2
• Geron: chapter 7
What We Will Cover

- Introduction to Ensemble Learning
- Various Ensemble Methods
- Bagging
Introduction to Ensemble Learning
Wisdom of Crowd

• Suppose we ask a complex question to thousands of random people, then aggregate their answers.
• In many cases we will find that this aggregated answer is better than an expert’s answer.

This is called the wisdom of the crowd.
Ensemble Learning

• Similarly, if we aggregate the predictions of a group of predictors (such as classifiers or regressors), we will often get better predictions than with the best individual predictor.
Ensemble Learning

• A group of predictors is called an ensemble or a committee.
• Thus, this technique is called Ensemble Learning.
• An Ensemble Learning algorithm is called an Ensemble method.
Ensemble Learning

• We often use Ensemble methods near the end of a project.
• Once we have already built a few good predictors, to combine them into an even better predictor.

• In fact, the winning solutions in Machine Learning competitions often involve several Ensemble methods.

• Example: Netflix prize competition

• https://netflixprize.com/
Ensemble Learning

• Ensemble methods work best when the predictors are diverse, i.e., as independent from one another as possible.
• There are two ways to create diverse classifiers.
• We can train them using very different algorithms.
• This increases the chance that they will make very different types of errors, improving the ensemble’s accuracy.
Ensemble Learning

• Another way to get diverse classifiers is to train the same algorithm on different subset of the data.
Ensemble Learning

- Thus, there are **two approaches** to build an ensemble.
  - Use very *different* training algorithms on the training set.
  - Use the *same* training algorithm for every predictor and train them on different random subsets of the training set.
Ensemble Learning: Use Different Training Algorithms

• For example, we can train a group of different classifiers on the same dataset.

• Then, aggregate the predictions of each classifier and predict the class that gets the most votes.

This majority-vote classifier is called a hard voting classifier.
Ensemble Learning

• Another approach is to use the **same training algorithm** for every predictor and train them on **different random subsets** of the training set.

For example, we can train a group of **Decision Tree classifiers**, each on a **different random subset** of the training set.
Ensemble Learning

• To make predictions, we just obtain the predictions of all individual trees, then predict the class that gets the most votes.

• Such an ensemble of Decision Trees is called a Random Forest.

Despite its simplicity, this is one of the most powerful Machine Learning algorithms available today.
Ensemble Learning

• The **same training algorithm** trained on **different random subsets** of the training set.

• There are **two ways to sample** from the training set.
  - Bagging (bootstrap aggregating): when sampling is performed with replacement.
  - Pasting: When sampling is performed **without replacement**.
Ensemble Learning

• There is another type of ensemble method known as **boosting**.

• It combines several *weak learners* into a strong learner.

• The general idea of most boosting methods is to train predictors sequentially, **each trying to correct its predecessor**.
Ensemble Learning

- We will discuss **two most popular Ensemble methods**.
  - Bagging
  - Boosting
- We will also explore Random Forests (it could use the bagging method as well as other methods).
How Do the Ensemble Methods Improve Performance?
Ensemble Learning

• The ensemble methods improve performance by **reducing the error** in prediction.
• Let’s see how this works.
• The simplest way to construct an ensemble is to **average the predictions of a set of individual models**.
• Such a procedure can be motivated from a **frequentist perspective** by considering the **trade-off** between bias and variance.
Ensemble Learning

• This trade-off decomposes the error due to a model into the bias component and the variance component.

• The bias that arises from differences between the model and the true function to be predicted.

• The variance component represents the sensitivity of the model to the individual data points.

The goal of the ensemble methods is to reduce both bias and variance by aggregating multiple learners.
Ensemble Learning

• Let’s see which models have low bias and which have low variance.

• Generally:
  - Weak models have low variance.
  - Sophisticated models have low bias.

• Let’s illustrate this.
Ensemble Learning: Combating the Bias-Variance Tradeoff

- Simple (a.k.a. weak) learners!
- E.g., Naïve Bayes, logistic regression, decision stumps (or shallow decision trees).
- **Good**: Low variance, don’t usually overfit.
- **Bad**: High bias, can’t solve hard learning problems.
Ensemble Learning: Combating the Bias-Variance Tradeoff

• Sophisticated learners:
  • Kernel SVMs, Deep Neural Networks, Deep Decision Trees.
• **Good**: Low bias, have the potential to learn with Big Data.
• **Bad**: High variance, difficult to generalize.

How do we **combine** the good properties (low bias & low variance) from these models?
Ensemble Learning

• In classification ensemble methods, instead of training a single classifier, we train many classifiers.

• Output class: (Weighted) vote of each classifier.
Ensemble Learning: Bagging

• The idea is that classifiers that are most “sure” will vote with more conviction.

• Therefore, using sophisticated learners, given the errors are uncorrelated, the expected error goes down.

• On average, the aggregate does better than single classifier!

• This is the essence of Bagging.
Ensemble Learning: Boosting

• However, with weak learners we have can cast the story differently.
• Say that each weak learner is good at different parts of the input space of the problem.
• After aggregating, on average, they do better than single classifier!

This is the main idea of Boosting.
Bagging
Bagging

- In bagging we train a group of **sophisticated learners**.

- For improved performance the **errors** of each individual learner needs to be **uncorrelated**.

- This can be done by **introducing variability** in the dataset.
Bagging

- The problem is we have **only a single data set**.
- So we have to find a way to **introduce variability** between the different models within the committee.
- One approach is to use **bootstrap data sets**.
Bagging

• In bootstrapping we use the same training algorithm for every predictor, but to train them on different random subsets of the training set.
• When sampling is performed with replacement, this method is called bagging (short for bootstrap aggregating).
• When sampling is performed without replacement, it is called pasting.
Bagging

- Both bagging and pasting allow training instances to be sampled several times across multiple predictors.
- But only bagging allows training instances to be sampled several times for the same predictor.
Bagging

• Once **all predictors are trained**, the ensemble can make a prediction for a new instance by simply **aggregating the predictions of all predictors**.

• The aggregation function is typically the **statistical mode** (i.e., the most frequent prediction, just like a hard voting classifier) for classification, or the **average** for regression.
Bagging

- Each **individual predictor has a higher bias & variance** than if it were trained on the original training set.
- But **aggregation reduces both bias and variance**.
- Generally, the net result is that the ensemble has a similar bias but a **lower variance** than a single predictor trained on the original training set.
How Does the Bagging Reduce Error?
Bagging

- We will show that under certain assumptions, the expected error made by the committee is $M$ times smaller than the average error made by the models acting individually.

$$E_{COM} = \frac{1}{M} E_{AV}.$$  

This is a dramatic assertion!

This is only possible if certain conditions are met.

Here $M$ is the number of bootstrap datasets.
Bagging

• We will prove the dramatic assertion on the reduction of error by the committee by using a **regression problem**.
• Consider a **regression problem** in which we are trying to predict the value of a **single continuous variable**.
• Suppose we generate **M bootstrap data sets**.

\[
E_{COM} = \frac{1}{M} E_{AV}.
\]
Bagging

\[
E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}.
\]

- Then, use each data set to train a separate copy \( y_m(x) \) of a predictive model where \( m = 1, \ldots, M \).
- The ensemble prediction is given by:

\[
y_{\text{COM}}(x) = \frac{1}{M} \sum_{m=1}^{M} y_m(x).
\]
Bagging

• Suppose the true regression function that we are trying to predict is given by $h(x)$.

• Hence, the output of each of the models can be written as the true value plus an unbiased error in the form:

\[ y_m(x) = h(x) + \epsilon_m(x). \]
Bagging

Then, for each model the average sum-of-squares error takes the following form:

$$y_m(x) = h(x) + \epsilon_m(x).$$

The average error made by the models acting individually is therefore:

$$E_{AV} = \frac{1}{M} \sum_{m=1}^{M} E_x [\epsilon_m(x)^2].$$

Here $E_x[\cdot]$ denotes a frequentist expectation with respect to the distribution of the input vector $x$. 

The average error made by the models acting individually is therefore:
Bagging

- The expected error from the committee (ensemble):

\[
y_m(x) = h(x) + \epsilon_m(x).
\]

\[
E_{COM} = \mathbb{E}_x [\{y_{COM}(\hat{x}) - h(\hat{x})\}^2]
\]

\[
E_{COM} = \mathbb{E}_x \left[ \frac{1}{M} \sum_{m=1}^{M} y_m(\hat{x}) - h(\hat{x}) \right]^2
\]

\[
E_{COM} = \mathbb{E}_x \left[ \frac{1}{M^2} \left\{ \sum_{m=1}^{M} y_m(\hat{x}) - M h(\hat{x}) \right\}^2 \right]
\]

\[
E_{COM} = \mathbb{E}_x \left[ \frac{1}{M^2} \left\{ \sum_{m=1}^{M} y_m(\hat{x}) - \sum_{m=1}^{M} h(\hat{x}) \right\}^2 \right]
\]

Taking \( \frac{1}{M} \) outside the square.
Bagging

• The expected error from the committee (ensemble):

\[
E_{COM} = \mathbb{E}_x \left[ \frac{1}{M^2} \left\{ \sum_{m=1}^{M} y_m(\hat{x}) - \sum_{m=1}^{M} h(\hat{x}) \right\}^2 \right]
\]

\[
E_{COM} = \mathbb{E}_x \left[ \frac{1}{M^2} \left\{ \sum_{m=1}^{M} [y_m(\hat{x}) - h(\hat{x})] \right\}^2 \right]
\]

Here, the error of the \textit{m}th prediction is given by:

\[
y_m(\vec{x}) - h(\vec{x}) = \epsilon_m(\vec{x})
\]
Bagging

- The expected error from the committee (ensemble):

\[
E_{COM} = \frac{1}{M^2} \mathbb{E}_x \left[ \left\{ \sum_{m=1}^{M} \epsilon_m(\tilde{x}) \right\}^2 \right]
\]

\[
E_{COM} = \frac{1}{M^2} \mathbb{E}_x \left[ \sum_{m=1}^{M} \epsilon_m(\tilde{x}) \sum_{l=1}^{M} \epsilon_l(\tilde{x}) \right]
\]

\[
E_{COM} = \frac{1}{M^2} \mathbb{E}_x \left[ \sum_{m=1}^{M} \sum_{l=1}^{M} \epsilon_m(\tilde{x})\epsilon_l(\tilde{x}) \right]
\]

Note that we are computing the expectation of the product of the errors of two predictors (index m and l).

\[y_m(x) = h(x) + \epsilon_m(x).\]
Bagging

\[ y_m(x) = h(x) + \epsilon_m(x) \]

- The expected error from the committee (ensemble):

\[
E_{COM} = \frac{1}{M^2} \mathbb{E}_x \left[ \sum_{m=1}^{M} \sum_{l=1}^{M} \epsilon_m(\tilde{x}) \epsilon_l(\tilde{x}) \right]
\]

Since Expectation is a linear operation:

\[
\mathbb{E} \left[ \sum_{i=0}^{n} X_i \right] = \sum_{i=0}^{n} \mathbb{E}[X_i]
\]

Separating the cross terms:

\[
E_{COM} = \frac{1}{M^2} \left[ \sum_{m=1}^{M} \mathbb{E}_x [\epsilon_m(x)^2] + \sum_{m\neq l}^{M} \mathbb{E}_x [\epsilon_m(\tilde{x}) \epsilon_l(\tilde{x})] \right]
\]
Bagging

We will make **two assumptions**.

- **Assumption 1**: the errors made by different predictors are *uncorrelated*.

Thus:

\[
E_{COM} = \frac{1}{M^2} \left[ \sum_{m=1}^{M} \mathbb{E}_x [\epsilon_m(x)^2] + \sum_{m \neq l} \mathbb{E}_x [\epsilon_m(\tilde{x})\epsilon_l(\tilde{x})] \right]
\]

Assumption 2: the mean of the unbiased error is zero.

\[
\mathbb{E}_x [\epsilon_m(\tilde{x})] = 0
\]

Hence, the cross term **vanishes**.

\[
E_{COM} = \frac{1}{M^2} \left[ \sum_{m=1}^{M} \mathbb{E}_x [\epsilon_m(\tilde{x})^2] \right]
\]
Bagging

The expected error from the committee (ensemble):

\[ y_m(x) = h(x) + \epsilon_m(x). \]

\[
E_{COM} = \frac{1}{M^2} \left[ \sum_{m=1}^{M} \mathbb{E}_x [\epsilon_m(\vec{x})^2] \right]
\]

\[
E_{AV} = \frac{1}{M} \left[ \sum_{m=1}^{M} \mathbb{E}_x [\epsilon_m(\vec{x})^2] \right]
\]

\[
E_{COM} = \frac{1}{M} E_{AV}
\]
Bagging

- This is a **dramatic result**!
- It suggests that the average error of a model can be **reduced by a factor of M** simply by averaging M versions of the model.

\[ E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}. \]
Bagging

• Unfortunately, it depends on the **key assumption** that the *errors* due to the individual models are **uncorrelated**.

\[
E_{COM} = \frac{1}{M} E_{AV}.
\]

\[
\mathbb{E}_x[\epsilon_m(\hat{x})] = 0
\]

\[
\mathbb{E}_x[\epsilon_m(\hat{x})\epsilon_l(\hat{x})] = \mathbb{E}_x[\epsilon_m(\hat{x})]\mathbb{E}_x[\epsilon_l(\hat{x})] \quad \text{for} \quad m \neq l
\]
Bagging

- In practice, the errors are typically highly correlated.
- Hence, the reduction in overall error is generally small.
- It can, however, be shown that the expected ensemble error will not exceed the expected error of the constituent models:

\[ E_{\text{COM}} \leq E_{\text{AV}} \]

\[ E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}. \]
Ensemble Learning

- In order to achieve more significant improvements, we turn to a more sophisticated technique for building ensembles, known as boosting.
Bagging

- In bagging, **predictors can all be trained in parallel**.
- We can use **different CPU cores** or even **different servers**.
- Similarly, **predictions can be made in parallel**.
- This is one of the reasons why bagging and pasting are such popular methods: they **scale very well**.
Bagging

- Illustration (moons dataset).
- Decision boundary of a single Decision Tree with the decision boundary of a **bagging ensemble of 500 trees**.
Bagging

- Illustration (moons dataset).
- The ensemble’s predictions will likely **generalize much better** than the single Decision Tree’s predictions.
- The ensemble has a comparable bias but a **smaller variance**.