Alin Tomescu

6.867 Machine learning | Week 5, Thursday, October 3rd, 2013| Lecture 9

Lecture 9

Exam: No lecture on that day. October 17th (evening exam)

- All the lecture materials that we will complete by the end of next week (1 thorugh 11)
- One more problem set this Friday, which will be fair material for the exam
- 5th problem set with exam-like questions
- Review session on that day in the lecture hall
- Closed book exam
- You will not need a calculator (but you can bring one)
- Not an algebra test

No lecture on October 15th

Classification trees

How can we improve this to address over-fitting?

Bootstrap average (Bagging)

For $b = 1, \dots, B$

(1) Draw n samples from the training set S_n with replacement $\Rightarrow S_n^b$
(2) Build a tree $\hat{T}^b$ based on our S_n^b

How can we predict?

$$\hat{h}_B(x) = \frac{1}{B} \sum_{b=1}^{B} h\left(x; \hat{T}^b\right)$$

Note that this is not a binary prediction, you can get $\hat{h}_B(x) = -0.25$, since $\hat{h}_B(x) \in [-1,1]$

How many training examples will not be selected in the bootstrap sample S_n^b:

The probability that an element will not be selected is $\left(1 - \frac{1}{n}\right)$, since we have n elements:

$$\left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$$

The reason to do bootstrap average: introduce trees that make errors in an uncorrelated fashion.

Unfortunately, this is not enough, because the trees still have a bunch of the training set in common. Thus, their predictions are still correlated. We want to decorrelate the predictions even more.

Random forest

Input S_n, B, m and $x \in \mathbb{R}^d, d = 100$

For $b = 1, \dots, B$

Alin Tomescu

6.867 Machine learning | Week 5, Thursday, October 3rd, 2013| Lecture 9

(1) Draw S_n^b with replacement (just like before)

(2) Build a tree $\hat{T}^b$ based on S_n^b, but…

   a. At each node in the tree you draw m coordinates at random and then find the best split among the m

      i. That means that instead of using all the $d = 100$ dimensions for making the decision at the current node in the tree, we will only use m.

$$\hat{h}_{rf}(x) = \frac{1}{B}\sum_{b=1}^{B} h(x; \hat{T}_m^b)$$

http://www.dabi.temple.edu/~hbling/8590.002/Montillo_RandomForests_4-2-2009.pdf

http://www.youtube.com/watch?v=3kYujfDgmNk

How do we choose m?

For each tree, I can evaluate an error $\hat{\varepsilon}^b(m) = \text{error on out of "bag" examples (the ones that were left out when we}$ selected S_n^b. We can apply our tree on those examples to get an error measurement.

We can the average all of them:

$$m = \text{argmin}_\text{m} \frac{1}{B}\sum_{b=1}^{B} \hat{\varepsilon}^b(m)$$

Ensemble methods

Random forests are one. We start with a too strong of a predictor, we randomize and take an average and as a result get a predictor that is stronger and generalizes well.

Boosting

Boosting looks at week learners and it optimizes the ensemble obtaining a strong predictor.

An example of an ensemble

$$h_m(x) = \sum_{i=0}^{m} \alpha_i h(x, \theta_i)$$

$$\alpha_i = votes$$

$$h(x, \theta_i) = weak\ learner$$

There is too much power in this ensemble so we have to do a poor job at optimizing the solution so we can generalize well.

Train the ensemble to minimize the training error. We find $\vec{\alpha}$ and $\vec{\theta}$

$$\sum_{i=1}^{n} Loss\left(y^{(i)} h_m\left(x^{(i)}\right)\right)$$

$$z = y^{(i)} h_m\left(x^{(i)}\right) = \text{agreement}$$

But this is computationally hard and also provides an overfitting solution. So we don't want this.

$Loss(z)$ can be the hinge less or logistic loss $Loss(z) = \log(1 + e^{-z})$, or exponential loss $Loss(z) = e^{-z}$.

We will not use hinge loss, because we want a smooth decrease in the loss function so as to guide the ensemble optimization.

Forward fitting

You start with $h_0(x) = 0$.

(1) Fix $\hat{h}_{m-1}(x)$, what is the final piece that I should add that best complements what I have. Find α_m and θ_m that minimize $J(\alpha_m, \theta_m) = \sum_{i=1}^{n} Loss\left(y^{(i)} h_{m-1}\left(x^{(i)}\right) + y^{(i)} \alpha_m h(x; \theta_m)\right)$. But even this is too much of a hard problem.