U.C. Berkeley — CS270: Algorithms
Professor Vazirani and Professor Rao
Lecturer: Umesh Vazirani Scribe: Anupam

Lecture 6
Feb 02,2012
Last revised March 1, 2012

Lecture 6

1 Boosting

Consider the problem of learning the concept class of hyperplanes. We have m labeled examples (x_i, l_i) with $l_i \in \{\pm 1\}$ and wish to find a hyperplane such that $l_i = sgn(h.x_i)$ for $(1-\nu)$ fraction of the points. Another way of to say this is that we want a hypothesis with error ν with respect to the uniform distribution. In practice, it is often easier to come up with good weak classifiers i.e. hyperplanes that are correct more often than wrong. How to combine several weak classifiers to get a strong one?

We formalize the problem as follows: a weak learner is an algorithm takes as input a distribution D on examples and produces a weak classifier $h(\cdot)$, such that $\Pr_{x\sim D}[h(x) = l(x)] \geq \frac{1}{2} + \gamma$. We wish to find a strong classifier h' that is correct on $1-\nu$ fraction of the training set, that is $\Pr_{x\sim D}[h'(x) = l(x)] \geq 1-\nu$. The AdaBoost algorithm uses the weak learner to find a strong classifier, it is an extremely influential application of the multiplicative weights framework to learning.

1.1 Adaboost

The algorithm can be formulated as a two player game where the row player plays examples (x_i, l_i) and the column player plays hypotheses $h \in H$. The loss for the row player is 1 if $h(x) = l(x)$ and 0 otherwise. The row player suffers a loss if the example is classified correctly, so the row player wants to play examples that fool the hypothesis played by the column player.

The algorithm repeats the following steps for $T = \frac{2}{\gamma^2} \log \frac{1}{\mu}$ rounds:

1. The row player follows the experts algorithm starting with a uniform distribution on the examples.
2. The column player invokes the weak learner on the row player's distribution to produce a hypothesis with expected payoff more than $1/2 + \gamma$.
3. The output hypothesis $h(x)$ is the majority of $h_1(x), h_2(x), \cdots, h_T(x)$.

Claim 1
If the multiplicative factor ϵ for the experts algorithm equals γ, the majority hypothesis correctly classifies $1-\mu$ fraction of the examples after $T = \frac{2}{\gamma^2} \log \frac{1}{\mu}$ rounds.

Proof: The proof modifies the analysis of the experts algorithm for lecture 4, there we observed that $W(T) \geq (1-\epsilon)^{L^*}$ as the total weight is at least the weight of the best expert.

Define S_{bad} be the set of examples misclassified by the majority hypothesis, examples in S_{bad} are good experts as they do not suffer a loss more than $T/2$. Here we observe that the total weight $W(T)$ is at least the weight of the experts in S_{bad},

$$|S_{bad}|(1-\epsilon)^{T/2} \leq W(T). \tag{1}$$

The upper bound in the analysis of the experts algorithm was $W(t+1) \leq (1-\epsilon L_t)W(t)$, applying the inequality $(1-\epsilon x) \leq e^{-\epsilon x}$ valid for all $x \in \mathbb{R}$ we have the upper bound $W(T) \leq W(0)e^{-\epsilon L}$. The column player uses the weak learner to ensure that the row player loses at least $\frac{1}{2}+\gamma$ in each round, hence

$$W(T) \leq e^{-\epsilon L}n \leq e^{-\epsilon(\frac{1}{2}+\gamma)T}n \tag{2}$$

Combining the upper and lower bounds, using $\epsilon = \gamma$ and taking logarithms we have,

$$\ln\left(\frac{|S_{bad}|}{n}\right) + \frac{T}{2}\ln(1-\gamma) \leq -\gamma T\left(\frac{1}{2}+\gamma\right) \tag{3}$$

Using the approximation $-\gamma-\gamma^2 \leq \ln(1-\gamma)$ from lecture 4 we have,

$$\ln\left(\frac{|S_{bad}|}{n}\right) \leq -\frac{\gamma^2 T}{2} = \log\mu \tag{4}$$

The fraction of mistakes is at most μ, hence the majority hypothesis classifies at least $1-\mu$ fraction of the examples correctly.

□

2 Congestion minimization in the experts framework

Recall the congestion minimization problem from lecture 1: given graph $G = (V,E)$ and pairs of vertices $(s_i,t_i), i \in [k]$ find paths connecting (s_i,t_i) such that the maximum congestion over an edge is minimized. Using the experts framework, we will find a flow that approximately minimizes congestion, the flow can be rounded off probabilistically to obtain approximately optimal paths.

The analysis of the expert algorithm with gains from lecture 4 shows that if the gains belong to $[0,\rho]$, then for the update rule $w_i(t+1) \rightarrow (1+\epsilon)^{\frac{g_i}{\rho}}w_i(t)$ the gain G of the experts algorithm is close to the gain G^* for the best expert,

$$G \geq (1-\epsilon)G^* - \frac{\rho\log n}{\epsilon} \tag{5}$$

2.1 The toll congestion game

The toll player plays an $e \in E$ and the routing player plays a routing r between (s_i,t_i). The gain for the toll player is $c(e,r)$, the congestion on the edge e in routing r.

Mixed strategies for the routing player are probability distribution on routings (s_i,t_i), notice that the mixed strategies are flows. The best response to a flow f is to play the edge with maximum congestion under f. The value of the game is $C^* = \min_f \max_e c(e,f)$, the optimal congestion over (s_i,t_i) flows.

A mixed strategy for the toll player is a probability distribution w_e on the edges, each toll strategy induces a metric on the graph where the length of edge e is equal to w_e. The expected payoff for routing r against toll strategy w is the sum of the lengths of the (s_i, t_i) paths in r under the toll metric,

$$A(r, w) = \sum_e w_e c(e, r) = \sum_{p_i \sim (s_i, t_i)} w(p_i) \tag{6}$$

The best response to a toll strategy w routes along shortest (s_i, t_i) paths in the toll metric. The number of strategies for the routing player is exponential, but the best response to a given a toll strategy can be found efficiently by computing shortest paths in the toll metric.

2.2 Congestion minimization algorithm

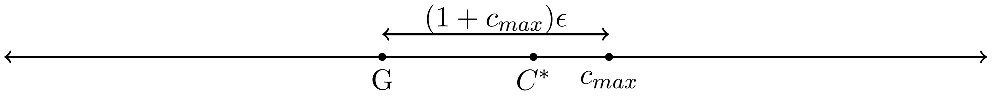
The algorithm for congestion minimization repeats the following steps for $T = \frac{k \log m}{\epsilon^2}$ rounds:

1. The toll player follows the experts algorithm, the initial weights are $w_e = 1$, the update rule is $w_i(t+1) \to (1+\epsilon)^{g_i^t/k} w_i(t)$.
2. The routing player plays the optimal response to the toll player's strategy, which is to route along the shortest (s_i, t_i) paths under the metric w.
3. The output of the algorithm is the flow $f = \frac{1}{T} \sum_t f(t)$ obtained by averaging the responses of the routing player.

The maximum congestion on an edge for the flow f is denoted by c_{max}, the following claim shows that c_{max} is within $O(k\epsilon)$ of the optimal congestion over flows,

Claim 2
The value of c_{max} is within $O(k\epsilon)$ of the optimal congestion C^* for $T = \frac{k \log m}{\epsilon^2}$.

Proof: Let G be the average gain for the experts algorithm over T rounds. The gain for every round is less than the value of the game C^* as the toll player plays first and then the routing player gets to choose the best response. The gain for the best expert in retrospect against $f = \sum f(i)/T$ is c_{max}. The value C^* is less than c_{max} as in this case the toll player gets to choose a best response.



The analysis of the experts algorithm (5) bounds the gap between c_{max} and G,

$$G \geq c_{\max}(1 - \epsilon) - \frac{k \log m}{\epsilon T} \tag{7}$$

Substitute $T = \frac{k \log m}{\epsilon^2}$ to obtain $c_{max} - C^* \leq c_{max} - G \leq (1 + c_{max})\epsilon = O(k\epsilon)$.
□

To approximate C^* within a multiplicative factor $(1 + \epsilon)$, the running time of the algorithm is $O(k^2 m \log n)$ as $O(k \log m)$ rounds are required and each round involves computing k shortest paths which can be done in time $O(km)$, say using breadth first search.

2.3 Rounding

The algorithm produces an approximately optimal flow which needs to be rounded to a routing. The flow f produced by the experts algorithm is $f = \sum f_i/T$, where each f_i is a routing as it is the best response to some toll strategy. For each pair of nodes (s_i, t_i) there are a total of T paths available and the rounding strategy is to choose one of these paths uniformly at random.

If edge e has congestion c_e in the flow f then the number of paths passing through e can be bounded by,

$$|P_e| \leq c_e T$$

The indicator random variable $X_i = 1$ if the path through (s_i, t_i) in the rounded strategy goes through e. The expected congestion for the edge e for the randomized strategy is,

$$E[\sum_{i\in[k]} X_i] = \sum_i \frac{|P_{e,i}|}{T} = \frac{P_e}{T} \leq c_e \tag{8}$$

The expected congestion for the randomized rounding procedure is at most the congestion for the original flow. The random variables X_i are defined to be independent so that we use concentration bounds and argue that with high probability the rounding procedure finds a routing with low congestion.