15-859(A) Machine Learning Theory

Lecture 02/17/04, Avrim Blum

Learning from noisy data; intro to SQ model

Recap: Weak and Strong learning

If we can get a weak bias over *every* distribution, then can get a strong bias over every distribution.

Idea: keep modifying distribution to extract new information.

Adaboost:

- Modify the distribution so that the previous hypothesis is 50/50. Do this by multiplying weight on correct points by $\beta = \frac{error}{1-error}$.
- Then take weighted vote over the hypotheses. Weight of h_t is proportional to $\ln(1/\beta_t)$.
- Use upper and lower bounds on total weight of example points to bound the error of the combination. (We analyzed fixed β in class.)

Learning when there is no perfect hypothesis

- Hoeffding/Chernoff bounds: minimizing training error will approximately minimize true error: just need $O(1/\varepsilon^2)$ samples versus $O(1/\varepsilon)$.
- What about polynomial-time algorithms? Seems harder.
  - Given data set S, finding conjunction with fewest mistakes is NP-hard.
  - Open problem: can you weak-learn (find a poly-time-computable hypothesis with error $< 1/2 - \varepsilon$) given only the assumption that the data is 90% consistent with some conjunction?
- One way to make progress: make assumptions on the "noise" in the data. E.g., Random Classification Noise model.

Learning from Random Classification Noise (Ch 5)

- PAC model, but assume labels coming from noisy channel.
- "noisy" Oracle $EX^{\eta}(c, D)$. η is the *noise rate*.

  Example x is drawn from D.

  With probability $1 - \eta$ see label $\ell(x) = c(x)$.

  With probability η see label $\ell(x) = 1 - c(x)$.
- E.g., if h has non-noisy error p, what is the noisy error rate?

Algorithm A *PAC-learns C with random classification noise* if for any $c \in C$, any distribution D, any $\eta < 1/2$, any $\varepsilon, \delta > 0$, given access to noisy examples from $EX^{\eta}(c, D)$, A finds a hypothesis h that is ε-close to c, with probability $1 - \delta$. Allowed time

$$poly\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta}, n, size(c)\right).$$

Example: Learning monotone conjunctions

Let's assume η is known.

Any ideas?

What if η is not known?

- Just sort x_i by $\Pr_{x \leftarrow EX^\eta(c,D)}[\ell(x) = 1 | x_i = 0]$.

  [not worrying about x_i such that $\Pr[x_i = 0] < \varepsilon/2n$.]

- Then consider the $\leq n$ hypotheses corresponding to a conjunction of a prefix of this ordering.

- Noisy data preserves relative quality ordering of hypotheses, so can test on new sample and pick best.

contd

Algorithm A *PAC-learns* C *with random classification noise* if for any $c \in C$, any distribution D, any $\eta < 1/2$, any $\varepsilon, \delta > 0$, given access to noisy examples from $EX^\eta(c, D)$, A finds a hypothesis h that is ε-close to c, with probability $1 - \delta$. Allowed time

$$poly\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta}, n, size(c)\right).$$

Q: Is this a plausible goal? I mean, we are asking the learner to get closer to c than the data is.

A: OK because noisy error rate is linear in true error rate (squashed by $1 - 2\eta$).

Learning monotone conjunctions

Let $p_i = \Pr_{x \leftarrow EX(c,D)}[c(x) = 1 \wedge x_i = 0]$. [i.e., without noise]

Any hypothesis conjunction that includes all x_i such that $p_i = 0$ and no x_i such that $p_i > \varepsilon/n$ is good. [total error $\leq \varepsilon$]

- So, just need to estimate this probability to additive error $\varepsilon/2n$. Can rewrite as

$$\Pr_{x \leftarrow EX(c,D)}[c(x) = 1 | x_i = 0] \cdot \Pr[x_i = 0]$$

- Second part is unaffected by noise.
- Let q_i be the first part. Use fact that:

$$\begin{aligned}\Pr_{x \leftarrow EX^\eta(c,D)}[\ell(x) = 1 | x_i = 0] &= q_i(1-\eta) + (1-q_i)\eta \\ &= \eta + q_i(1-2\eta).\end{aligned}$$

  So, can approximate q_i from observations if $\Pr[x_i = 0]$ is not too tiny: just apx LHS to additive $O((\varepsilon/n)(1-2\eta))$. And if $\Pr[x_i = 0]$ is too tiny, then we're OK too!

Generalizing our algorithm

Basic idea of conjunction-learning alg:

- See how we could learn in non-noisy model by just asking about probabilities of certain events with some "slop".
- Try to estimate these probabilities from noisy data by breaking into
  - parts predictably affected by noise.
  - parts unaffected by noise

Next topic:

- Formalize this in the notion of a "statistical query algorithm".
- Show how any SQ algorithm can be used to learn with classification noise.
- Can actually characterize the kinds of things that can or can't be done with SQ algorithms.

Example: conjunctions

- Ask for $\Pr[c(x) = 1 \wedge x_i = 0]$ with $\tau = \varepsilon/2n$.
- Produce conjunction of all x_i such that $\hat{P}_\chi \leq \varepsilon/2n$.

Open problem

Can noise tolerance be boosted?

Say for concept class C there exists alg A such that for any $c \in C$, any distribution $\mathcal{D}$, any $\eta < 0.1$, A PAC-learns from $EX^\eta(c, D)$.

Does this imply there must exist an algorithm B that succeeds for all $\eta < 1/2$, with running time $poly(\frac{1}{1-2\eta})$?

Seems the answer may be no. There's a subclass of parity that we can learn in polynomial time for constant η, but the best known algorithm has running time $\left(\frac{1}{1-2\eta}\right)^{\sqrt{\log n}}$, so can't handle $\eta = 1/2 - 1/n$, say.

But maybe we can boost from $\eta = 10\%$ to any *constant* $\eta < 1/2$.

Hard part: given data source with 20% noise, how to run alg A?

The Statistical Query Model

- No noise.
- Algorithm asks "what is the probability a labeled example will have property χ? Please tell me up to additive error τ."
- Formally, $\chi : X \times \{0,1\} \rightarrow \{0,1\}$. Must be poly-time computable. $\tau \geq 1/poly(\cdots)$.

  If $P_\chi = \mathbf{E}_{c,D}[\chi(x, c(x))] = \Pr_{c,D}(\chi = 1)$ then world responds with $\hat{P}_\chi \in [P_\chi - \tau, P_\chi + \tau]$.

  (can extend to $\chi : X \times \{0,1\} \rightarrow [0,1]$)
- May repeat this $poly(\cdots)$ many times. Algorithm may also ask to see unlabeled examples.
- Algorithm must output a hypothesis with error less than ε. (No δ in this model.)

How to estimate Pr$[\chi = 1]$

- CLEAN $= \{x : \chi(x,0) = \chi(x,1)\}$.
- NOISY $= \{x : \chi(x,0) \neq \chi(x,1)\}$.

$$\begin{aligned} \Pr[\chi = 1] &= \Pr[\chi = 1 \wedge x \in \text{CLEAN}] + \\ &\quad \Pr[\chi = 1 \wedge x \in \text{NOISY}]. \end{aligned}$$

- Step 1: First part is easy to estimate from noisy data.

  (easy to test if a given x is in CLEAN)

[Running example: $\chi = 1$ if $c(x) = 1 \wedge x_i = 0$.]

How powerful are SQ algs?

- Most algs in practice are (roughly) SQ algorithms.

  E.g., ID3, gradient descent.
- Most are already tolerant to CN.

Can we quantify/characterize the kinds of things doable with SQ algorithms?

For next time...

SQ $\Rightarrow$ PAC+CN

Given query χ need to estimate from noisy data. Idea:

- Break χ into part predictably affected by noise and part unaffected by noise.
- Estimate these parts separately.
- Can draw fresh examples for each query, or estimate many queries on same sample if the $VCdim$ of possible queries of alg is small.

[Running example: $\chi = 1$ if $c(x) = 1 \wedge x_i = 0$.]

Proof, contd.

To estimate: $\Pr[\chi = 1 \wedge x \in \text{NOISY}]$

- First estimate $\Pr[x \in \text{NOISY}]$.
- Then estimate $\Pr_\eta[\chi = 1 | x \in \text{NOISY}]$.
- Then write $\Pr[\chi = 1 | x \in \text{NOISY}]$ in terms of $\Pr_\eta[...]$.
- Just need to estimate $\Pr_\eta[\chi = 1 | x \in \text{NOISY}]$ up to additive error $O(\tau(1 - 2\eta))$.
- If don't know η can guess and verify.

[Running example: $\chi = 1$ if $c(x) = 1 \wedge x_i = 0$.]