Topics in Learning Theory

Lecture 10: Research Topics in Learning Theory

Key Concepts from the Previous Lectures

- Supervised Learning with Regularized Empirical Risk Minimization
  - Test Error = Training Error + Model Complexity
- How to Estimate Model Complexity
  - Concentration — exponential probability inequality
  - Covering numbers
  - Rademacher complexity
- Regularization and model complexity
- Kernel methods — deal with infinity dimensional L_2 regularization
- Boosting — deal with infinity dimensional L_1 regularization/sparsity

Additional Research Topics

- Fast Convergence in statistical learning
- Online Learning
- Clustering (unsupervised learning)
- Semi-supervised learning
- Active learning
- Complex Output Prediction and complex regularization
- Sparsity

Fast Convergence

- Standard convergence rate: $\sqrt{1/n}$
- Fast $1/n$ convergence is possible under Bernstein like condition: $Var(\phi(f(x), y) - \phi(f_*(x), y)) \leq bE(\phi(f(x), y) - \phi(f_*(x), y))$
- Binary classification example: statistical margin condition (Tsybakov noise condition)

$$Var(\phi(f(x), y) - \phi(f_*(x), y)) \leq b[E(\phi(f(x), y) - \phi(f_*(x), y))]^\alpha$$

  for some $\alpha \in [0, 1]$.

  – let $f_*(x) = P(Y = 1|X)$

- the difficult case is around $P(Y = 1|X) = 0.5$
- the condition

- Convergence rate:
  - $\alpha = 0$ means no fast convergence $\sqrt{1/n}$ rate
  - $\alpha = 1$ means $1/n$ rate
  - general α implies a rate in-between

- Modern technique:
  - localized Rademacher complexity and Bernstein style concentration inequality

- Related question: how to adapt to unkown α?

Online Learning

- We observe data sequentially $(X_1, Y_1), (X_2, Y_2), \ldots$
- At each time t,
  - Nature reveals X_t
  - Statistician makes prediction $f_t(X_t)$
    * f_t depends on $(X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1})$
  - Nature reveals Y_t, and suffer loss $\phi(f_t(X_t), Y_t)$
- Goal: find prediction rules to minimize cummulate loss

$$\sum_{t=1}^{n} \phi(f_t(X_t), Y_t).$$

- Regret: Compare performance to best of $f \in \mathcal{H}$

$$\sum_{t=1}^{n} \phi(f_t(X_t), Y_t) - \inf_{f \in \mathcal{H}} \sum_{t=1}^{n} \phi(f(X_t), Y_t)$$

- Related: stochastic gradient descent

- Traditional:
  - perceptron (2-norm regularization), winnow (entropy regularizaton)

- Modern: convex game change $\phi(f_t(X_t), Y_t)$ to $\phi_t(w_t)$,
  - $w_t \in \Omega$, where Ω is a convex set.
  - $\phi_t(w)$ is convex in w

Convex online learning

– algorithm:

$$w_t = P_\Omega(w_t') \quad w_t' = w_{t-1} - \eta \nabla_w \phi_t(w_{t-1}),$$

where $P_\Omega(w)$ is the closet point in Ω to w.

– regret bound: $\|\nabla_w \phi_t(w)\|_2 \leq b$, then

$$\sum_{t=1}^{n} \phi_t(w_t) - \inf_{w \in \Omega} \sum_{t=1}^{n} \phi(w)$$

Analysis

Given any $w \in \Omega$

$$
\begin{aligned}
&\|w_t - w\|_2^2 - \|w_{t-1} - w\|_2^2 \le \|w'_t - w\|_2^2 - \|w_{t-1} - w\|_2^2 \\
=&\|w'_t - w_{t-1}\|_2^2 + 2(w'_t - w_{t-1})(w_{t-1} - w) \\
=&\eta^2 b^2 - 2\eta \nabla_w \phi_t(w_{t-1})(w_{t-1} - w) \le \eta^2 b^2 + 2\eta(\phi_t(w) - \phi_t(w_{t-1})).
\end{aligned}
$$

Summing over $t = 1$ to n:

$$
\sum_{t=1}^{n} \phi_t(w_{t-1}) \le \sum_{t=1}^{n} \phi_t(w) + \frac{1}{2\eta}\|w_0 - w\|_2^2 + \frac{n\eta}{2} b^2.
$$

Taking optimal η, we have

$$\frac{1}{n}\sum_{t=1}^{n}\phi_t(w_{t-1}) \leq \frac{1}{n}\sum_{t=1}^{n}\phi_t(w) + b\|w_0 - w\|_2/\sqrt{n}.$$

- $\sqrt{1/n}$ convergence rate: same as batch setting
- Other developments: faster rates, other update rules, etc

Clustering

- Partition data into groups so that points within each cluster are close and points between clusters are not close
  - example optimization problem (k-means): find $c_1, \ldots, c_k$ to minimize

$$\min_{c_1,\ldots,c_k} \sum_{i=1}^{n} \min_j \|x - c_j\|_2^2$$

- non-convex optimization problem

Clustering Research

- When can a nonconvex clustering problems be solved efficient?
- Imposing assumptions
- An example assumption: k cluters that are well separated
  - points within each cluster are very close to each other
  - points between different clusters are very far from each other
- one can find clusters accurately
  - example algorithm: find one point, then furthest point away, and so on...

Active learning

- Supervised learning: (X_i, Y_i) are random
- Active learning:
  - obtaining label is expensive
  - how to choose X_i to label? want to label as few examples as possible.
- Related to experimental design in statistics

Confidence based active learning algorithm

- The basic idea
  - if the current classifier can make confidence prediction on a point, it does not carry much information
  - thus select thos points to label where the current classifier does not make confident predictions — gain more information
- Example: margin based active learning: iterate the following steps
  - train a linear classifier with the current set of labeled data
  - randomly draw a sample: skip if it is larger than a certain margin (more confident), accept to label otherwise (less confident)

Theory

- Example where active learning is effective:
- assumption:
  - x is uniformly distributed in a d-dimensional ball
  - there is a perfect linear classifier
- in order to achieve error $\leq \epsilon$ with fixed probability
  - supervised learning: require $\tilde{O}(d/\epsilon)$ examples — vc theory
  - (margin based) active learning: needs $\tilde{O}(d \ln(1/\epsilon))$ examples

A more general positive result

- assumption:
  - binary classification with hypothesis from finite VC-class
  - there exists a perfect classifier
- conclusion: active learning helps asymptotically
  - faster rate of convergence then supervised learning

Semi-supervised learning

- Labels are expensive but unlabeld data can be abundant.
  - how to take advantage of unlabeld data to improve performance?
- Require assumptions.

Example: graph semisupervised learning

- Given labeled data $(X_1, Y_1), \ldots, (X_n, Y_n)$ and unlabeled data $X_{n+1}, \ldots, X_m$.
- Form undirected graph using data $X_1, \ldots, X_m$:
  - connect each point to its k nearest neighbors
- define regularization conditon/kernel using the graph (graph Laplacian):

$$\sum_{j' \in N_k(j)} (f(X_j) - f(X_{j'}))^2$$

  - intuition: conntected nodes should have similar labels

- Questions: when is this method effective? How does the graph Laplacian regularization operator behave (when $m \to \infty$), etc

Example: multiview learning

- Assume we can decompose each X into two parts X^1 and X^2 representing two views: for example, multiple camera angles for face recognization or speech + face

- Assume each view is sufficient in predicting the target with a linear classifier w^1 and w^2 separately.

- Then we can require $w^{1T} x^1 \approx w^{2T} x^2$

- Solving the following co-regularization formulation:

$$\min[\sum_{\ell=1}^{2} \sum_{i} \phi(w^{\ell T} X_i^{\ell}, Y_i) + \lambda \sum_{j} (w^{1T} X_j^1 - w^{2T} X_j^2)^2]$$

where i goes through labeled data and j goes through unlabeled data

Complex Prediction: multi-task learning

- Consider multiple prediction problems, indexed by ℓ: observe samples (X_i^ℓ, Y_i^ℓ).

- Complex objective function: can we benefit by solving multiple problems joint?
  - Yes if there are shared components

- Need to design complex regularization to couple the multiple problems

Example: sharing mean

- We have linear classifier w^ℓ for the ℓ-th problem. Assume 2-norm regularization, if solving independently:

$$w^\ell = \arg\min_{w^\ell} [\sum_i \phi(w^{\ell T} X_i, Y_i) + \lambda \|w^\ell\|_2^2]$$

- Joint regularization: sharing a mean vector $\bar{w}$: each weight is the mean vector plus a small variation

$$[\bar{w}, w^\ell] = \arg\min_{\bar{w},[w^\ell]} [\sum_{i,\ell} \phi(w^{\ell T} X_i, Y_i) + \lambda \sum_\ell \|w^\ell - \bar{w}\|_2^2]$$

Example: sharing low-dimension space

- We have linear classifier w^ℓ for the ℓ-th problem, and separate shared low-dimensional projection of X to QX.

- Joint regularization: sharing a mean vector $\bar{w}$: each weight is the mean vector plus a small variation

$$[Q, \bar{w}, w^\ell] = \arg \min_{Q, \bar{w}, [w^\ell]} [\sum_{i,\ell} \phi(w^{\ell T}[X_i, QX_i], Y_i) + \lambda \sum_\ell \|w^\ell\|_2^2]$$

Sparsity

- Assumption w is sparse or can approximated by a sparse weight
- empirical risk minimization

$$w = \arg\min_w \sum_i \phi(w^T X_i, Y_i) \quad \text{s.t. } \|w\|_0 \leq b$$

  - non-convex sparse constraint
  - when can it be solved efficiently?
  - study the effectiveness of approximate solutions: L_1 and greedy algorithms – very active research topic

L_1 regulariazation

- Relax L_0 regularization to L_1 regularization (convex):

$$\hat{w} = \arg\min_w \sum_i \phi(w^T X_i, Y_i) \quad \text{s.t. } \|w\|_1 \leq b$$

- Example result:
  - under some assumptions, it produces the same set of nonzeros as L_0 regularization, thus can be used to solve the non-convex problem.
  - can allow $d \gg n$: the assumption roughly requires small blocks of matrix $\frac{1}{n}\sum_{i=1}^{n} \phi''(\hat{w}^T X_i, Y_i) X_i X_i^T$ to be close to diagonal.

Some example applications

- Prediction problems with sparse target

- Sparse principal component analysis (sparse eigenvalue problem)

$$w = \arg \max_{w:\|w\|_2=1} w^T A w \quad \text{s.t. } \|w\|_0 \leq b$$

- Graphical model learning (whether variables are correlated)

$$W = \arg\max \ln(S^{-1}W) \quad \text{s.t. } \|W\|_0 \leq b$$

and W is positive semi-definite.

Where to Learn More

- Major Conferences on Machine Learning:
  - COLT (Conference on learning theory)
  - NIPS (Neuro-information Processing System)
  - ICML (international conference of machine learning)

- All Proceedings and Papers are online

- Questions: email me *tongz@rci.rutgers.edu*