COS 598D: Overcoming intractability in machine learning.
Sanjeev Arora, Princeton University. Spring 2015.

Lecture 1: A whirlwind survey of machine learning and ML theory.

Various meanings of learning. Usual assumption: data consists of iid samples from some distribution. (Philosophical aside: De Finetti's theorem, exchangeability.)
A running example in this lecture will be linear classifiers.

- Unsupervised vs Supervised.

  Unsupervised: Unlabeled data. Usually need some kind of model for how the data was generated (the "story") and then recover the model parameters. Examples: k-means and other forms of clustering, loglinear models, bayes nets, ..
  Often NP-hard.

  Supervised: Training data is labeled by a human (labels could be binary, or in [1..k]). Algorithm needs to predict labels on future data. Examples: Decision trees, SVMs, k-NN.

  Generalization bounds: connect performance on training data with that on unseen data (i.e. the entire distribution).

  Rough idea: Suppose there is a classifier that is representable with M bits and has error at most \epsilon on the full distribution. Then if the training set has size at least f(M + K, \epsilon), then any classifier describable in K bits that has error at most \epsilon on the training set will have error at most 2\epsilon on the entire distribution. Furthermore, if any classifier has error at most \epsilon/2 on the entire distribution, it has error at most \epsilon on the sample. (Thus the method is complete and sound: if there exists a good classifier, it can be found by examining a small set of training points, and conversely every classifier that is good enough on the samples is also good enough for the entire distribution.)

  Proof sketch: Chernoff bounds. Only 2^M classifiers that can be represented using M bits. If any M-bit classifier has error more than 2\epsilon fraction of points of the distribution, then the probability it has error only \epsilon fraction of training set is $< 2^{-M}$. Hence whp no bad classifier can be good on the training points.

  There is a more general theory for computing the number of training points that involves VC dimension. Pls see online sources.

  The classical philosophical principle Occam's razor is related to this.

  Example of training: Perceptron algorithm for linear classifier. (Can think of as a way to determine weights of features.) Completely nonbayesian description.

Can also turn into convex program using the notion of a margin.

- Discriminative vs Generative.

  Discriminative: Only know P(label| data). (Example 1: label = linear threshold corresponds to SVM. Note this is deterministic. Example 2: Logistic regression. Smoothened version of SVM.) Examples of Discriminative learners: decision trees, SVMs, kernel SVMs, deep nets, logistic regression etc. SVMs and logistic regression can be solved by convex optimization.

  Generative: Know an expression for P(label, data). Estimate P(label|data) by Bayes rule and calculating P(label, data)/P(data). For an example see the application to spam classification in Lecture notes by Cynthia Rudin on Naïve Bayes.

  Also see the chapter in Mitchell's book, which shows that logistic regression corresponds to a naïve bayes estimator where coordinates are iid Gaussian.

  For a more hands-on viewpoint with worked out examples, see Chris Manning's lecture notes.
  https://web.stanford.edu/class/cs124/lec/Maximum_Entropy_Classifiers.pdf

- (Aside: Logistic regression is great right? How about if we stack up logistic regression units in a circuit? We get deep nets. Training these is a nontrivial task and we only know of heuristic algorithms. We don't know of a good bayes interpretation of such deep net classifiers.)

- There can be advantages to each. Discriminative needs fewer assumptions. Generative is easier to adapt to semisupervised settings.

  See

  Relevant chapter on generative vs discriminative in Tom Mitchell's book.

  On discriminative vs generative: A comparison of logistic regression and naïve bayes, by Ng and Jordan in NIPS 2001.

  Generative or Discriminative? Getting the best of both worlds by Bishop and Lasserre. Bayesian Statistics 2007.

- *Regularization*. Technique used to avoid overfitting. For this it is better to use a less complicated solution, and adding a regularizer to the objective can help with this. (related to the generalization theory; a rough analogy is to restrict yourself

to solutions that can be described with fewer # of bits. This is only a rough intuition)

We will focus a lot on unsupervised learning.

Max Likelihood and Maximum Entropy Principle.

Given a choice between many possible distributions that fit the data, pick the one with maximum entropy.

Example: We are given a die that, when thrown, produces an expected value 4.7. What is the chance that it produces 5? Solution: Let
p_i = prob it produces i. Then average of i p_i is 4.7. Compute values of p_i 's that maximize entropy subject to this average.

Example 2: If we are only given the mean of a distribution, the max entropy distribution consistent with this is the exponential. (If the variable is n-variate, the distribution is loglinear.)

Example 3: If we are only given the mean and the covariances of a distribution then the max entropy distribution consistent with that is the gaussian.

Max likelihood method

Find the parameter vector Theta that maximizes the likelihood of seeing the data.

(Aside: Amazingly, Shannon in 1948 also invented NLP in addition to information theory by describing n-gram models for languages and suggesting measuring them using his entropy measure, which we can think of as max likelihood.
http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf)

Example: Max log likelihood expression for logistic regression from
http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression

Recall that in logistic regression, we had a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ of m labeled examples, where the input features are $x^{(i)} \in \Re^{n+1}$. (In this set of notes, we will use the notational convention of letting the feature vectors x be $n + 1$ dimensional, with $x_0 = 1$ corresponding to the intercept term.) With logistic regression, we were in the binary classification setting, so the labels were $y^{(i)} \in \{0, 1\}$. Our hypothesis took the form:

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)},$$

and the model parameters θ were trained to minimize the cost function

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

This expression is convex, and so gradient descent methods find the best fit very fast. In fact, this computational ease is the reason for the popularity of logistic regression ---this popularity dates back to pre-computer days when people were using slide rules etc.

Unfortunately, when you compute the log likelihood for most other settings, the expression turns out to be nonconvex. Such nonconvex optimization often turns out to be NP-hard (as has been proved for many settings).

Simple example: mixture of spherical gaussians of the same radius. Maximising the log likelihood is tantamount to k-means clustering.

(From Arora-Kannan: Learning Mixtures of Separated Nonspherical Gaussians; https://www.cs.princeton.edu/~arora/pubs/gaussians.pdf)

Trying to overcome this intractability is a major goal in this course.

4. Max-likelihood estimation. Now we describe an algorithm for max-likelihood fit of a mixture of k spherical Gaussians of equal radius to (possibly) unstructured data. First we derive a combinatorial characterization of the optimum solution in terms of the *k-median* (*sum of squares*, *Steiner version*) problem. In this problem, we are given M points $x_1, x_2, \ldots, x_M \in \Re^n$ in $\Re^n$ and an integer k. The goal is to identify k points $p_1, p_2, \ldots, p_k$ that minimize the function

$$\sum_{i=1}^{M} |x - p_{c(j)}|^2, \tag{26}$$

where $p_{c(j)}$ is the point among $p_1, \ldots, p_k$ that is closest to j and $|\cdot|$ denotes Euclidean distance.

THEOREM 13. *The mixture of k spherical Gaussians that minimizes the log-likelihood of the sample is exactly the solution to the above version of k-median.*

PROOF. Recall the density function of a spherical Gaussian of variance σ (and radius $\sigma\sqrt{n}$) is

$$\frac{1}{(2\pi\sigma)^{n/2}} \exp\left(-\frac{|x-p|^2}{2\sigma^2}\right).$$

Let $x_1, x_2, \ldots, x_M \in \Re^n$ be the points. Let $p_1, p_2, \ldots, p_k$ denote the centers of the Gaussians in the max-likelihood solution. For each data point x_j let $p_{c(j)}$ denote the closest center. Then the mixing weights of the optimum mixture $w_1, w_2, \ldots, w_k$ are determined by considering, for each i, the fraction of points whose closest center is p_i.

The log-likelihood expression is obtained by adding terms for the individual points to obtain

$$-\left[\text{Constant} + \frac{Mn}{2}\log\sigma + \sum_j \frac{|x_j - p_{c(j)}|^2}{2\sigma^2}\right].$$

The optimum value $\hat{\sigma}$ is obtained by differentiation,

$$\hat{\sigma}^2 = \frac{2}{Mn}\sum_j |x_j - p_{c(j)}|^2, \tag{27}$$

which simplifies the log-likelihood expression to

$$\text{Constant} + \frac{Mn}{2}\log\hat{\sigma} + \frac{Mn}{4}.$$

Thus the goal is to minimize $\hat{\sigma}$, which from (27) involves minimizing the familiar objective function from the sum-of-squares version of the k-median problem. □