Math 690: Topics in Data Analysis and Computation

Lecture notes for Fall 2017

Scribed by Yixin Lin, Shen Yan

August 31

We ended last time on a few basics of machine learning.

An important point is to be careful not to "cheat" when measuring the performance of your models! Of course you want to know how well your model works on your current dataset, but you can fail on datasets that you haven't seen before. This is the concept of **generalization**: we want our models to generalize to the unseen samples as well.

Machine learning basics

Goal: $y = f^*(x)$.

$$x_i \sim P_x, \text{ iid}, \quad y_i = f^*(x_i)$$

Training set $D_{Tr} = \{x_i, y_i\}_{i=1}^{n_{Tr}}$

Testing set (independent of training set, but from the same distribution) $D_{Te} = \{x_j, y_j\}_{j=1}^{n_{Te}}$

Training error: error on the training dataset.

Test error: error using trained model on test set. (also known as generalization error)

Recall that test error is

$$\varepsilon_{Te}(\theta) = \frac{1}{n_{Te}} \sum_{j=1}^{n_{Te}} (y_j - f_\theta(x_j))^2 \to^{n\to\infty} \int (f^*(x) - f_\theta(x))^2 p(x) dx$$

We are trying to search for parameters $\theta \in \Theta$ such that $f_\theta(x) \approx f^*(x)$.

The issues of overfitting and under-fitting: f^* may not be in the family we are considering. When the family is too small, the trained model underfits. Thus we want to enlarge our family of models. However, we may need more samples to estimate f^* when considering a large family. If we don't have enough samples, the trained model overfits and poorly generalizes to unobserved samples.

Breakdown of error

We can break down the error into three terms: the approximation error, estimation error, and optimization error. ([ref] This is Part I of "The Tradeoffs of Large-scale Learning" by Leon Botton, 2007.)

Our true objective is minimizing the error

$$\varepsilon(\theta) = \int (f^*(x) - f_\theta(x))^2 dp(x) = \mathbb{E}_{x\sim p} |y - f_\theta(x)|^2$$

but we know that our family of models may not be big enough. As long as f^* is not in the family of functions we are considering, then this error will never be zero:

$$\min_{\theta\in\Theta} \varepsilon(\theta) = \varepsilon(\theta^*) > 0$$

Even if you have infinite training samples, if your family is small, then the error *will never go to zero.* We call this the **bias** term.

We don't know p, so we don't know how to compute the integral. So what we really do in practice is minimize the training error. Instead of $\min_\theta \varepsilon(\theta)$, we minimize the following:

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \hat{\varepsilon}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2$$

which is a finite sample approximation of the integral, and in general $\hat{\theta} \neq \theta^*$. This means that $\varepsilon(\hat{\theta}) > \varepsilon(\theta^*)$.

Let us write $\varepsilon(\hat{\theta}) = \varepsilon(\theta^*) + (\varepsilon(\hat{\theta}) - \varepsilon(\theta^*))$. $(\varepsilon(\hat{\theta}) - \varepsilon(\theta^*))$ is the **estimation error**.

The third term of optimization will be that we want to minimize our previous error, but we are using a computer to approximate and compute this. The third term of the breakdown is **optimization error** and caused by imperfection in the optimization of

$$\arg\min_{\theta\in\Theta\varepsilon(\hat{\theta})}$$

This imperfection causes there to be an additional term:

$$\Rightarrow \varepsilon(\theta_{\text{sol}}) = \varepsilon(\hat{\theta}) + \varepsilon_{\text{opt}}$$

$$\varepsilon = \varepsilon_{\text{approx}} + \varepsilon_{\text{est}} + \varepsilon_{\text{opt}}$$

We mainly first focus on the first two terms in this class.

No free lunch theorem

[ref] "The Lack of A Priori Distinctions Between Learning Algorithms" Wolpert '96.

The question is, do we have a model better than another? No machine learning model is uniformly better than another.

What do we mean by *better*?

No free lunch theorem for optimization

Our goal is to minimize some function $f : V \to S$. For any two models A and B, the *averaged* performance of A and B are identical.

What do we mean by averaged performance? Over $\sum_f$, f is uniformly averaged over S^V.

No free lunch theorem for machine learning

For any two models A and B, there are "as many" targets for which model A has lower generalization error than model B, and vice versa. In other words, I can always find some dataset for A to work worse than B, and vice versa. So we need more assumptions about the distribution of the data.

The take-home message is that for the question of "which algorithm is better", the answer is *it depends on the data.*

Topic 1: Principal Component Analysis in High Dimension

Two perspectives: the linear algebra perspective, and the probability perspective

Linear algebra perspective

Suppose I have data vectors

$$x_1, \cdots, x_n \in \mathbb{R}^D$$

with center of mass in the origin: $\sum_{i=1}^n x_i = 0$. (Here, $D = p$, the dimensionality of the data.)

Our goal is to find the "optimal" d-subspace to maximize the projected variation of data. (Maximizing variation is minimizing residual; verifying this is a homework problem.)

Let $d = 1$, $w \in \mathbb{R}^p$, and constrain $\|w\| = 1$.

$$\max_w \sum_{i=1}^n (w^T x_i)^2 = \sum_{i=1}^n w^T x_i x_i^T w$$

$$= w^T (\sum_{i=1}^n x_i x_i^T) w$$

$$S \triangleq \frac{1}{n} \sum_{i=1}^n x_i x_i^T$$

So our goal is

$$\max_w w^T S w \text{ s.t. } \|w\| = 1$$

We need to also add the constraint that $w_k^T w_k = 1$, $w_k^T w_l = 0$ for all $k \neq l$. Equivalently, $w_k^T w_l = \delta_{kl}$. In other words, the vectors should be mutually orthogonal.

Objective will be

$$\max_{w_1, \cdots, w_d} \sum_{k=1}^d w_k^T S w_k$$

with the constraint of orthogonality.

We know the solution to this problem from linear algebra: the first d eigenvectors of S.

S has p non-negative eigenvalues $\lambda_1, \cdots, \lambda_p$ (sorted by magnitude $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$) associated with p eigenvectors $v_1, \cdots, v_p$.

To solve this problem, compute the eigendecomposition of the covariance matrix can be computed by the singular value decomposition (SVD) of X.

So far there is no probability.

Probability perspective

Suppose $\{x_i\}_{i=1}^n \sim^{\text{i.i.d.}} P$, in $\mathbb{R}^p$.

$$\mathbb{E} x_i = 0, \mathbb{E} x_i x_i^T = \Sigma_{p \times p}$$

Our goal:

$$\max_{w_1,\cdots,w_d|w_k^T w_l=\delta_{kl}} \sum_{k=1}^{d} \mathbb{E}_{x\sim P}(x^T w_k)^2$$

where the notation $\mathbb{E}_{x\sim p} f(x) = \int f(x)dP(x)$, we interchange $dP(x)$ with $p(x)dx$, $p(x)$ being the probability density. The objective is thus

$$\sum_{k=1}^{d} w_k^T(\mathbb{E}_{x\sim p} xx^T)w_k = \sum_{k=1}^{d} w_k^T \Sigma w_k.$$

The solution is obtained by the eigendecomposition of Σ.

However, the population covariance matrix Σ is usually not available, and we approximate it by the sample covariance S, which goes back to the PCA of finite samples (the 1st perspective).

September 5

Last time we saw PCA from two perspectives: linear algebra, and probability (population covariance matrix).

Why can PCA be solved by eigenproblem? Recall we want to

$$\max_{w_1,\cdots,w_d,w_k^Tw_l=\delta_{kl}} \sum_k w_k^T S w_k$$

Solved by the eigendecomposition of the matrix S.

Courant-Fischer Minimax Theorem For any Hermitian or real-symmetric matrix, we know it has n real eigenvalues with eigenvectors forming a orthonormal basis.

Let A be an $n \times n$ Hermition matrix. It has n eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.

$$\lambda_k(A) = \sup_{\dim(V)=k} \inf_{\|v\|=1, v\in V} v^*Av$$

$$\lambda_k(A) = \inf_{\dim(V)=n-k+1} \sup_{\|v\|=1, v\in V v^*Av} v^*Av$$

Proof: $A = U\Lambda U^*$, verify for each k.

(Ex) We mentioned that PCA can be viewed as maximizing the projected variation, which is equivalent to minimizing the residual after projection.

$$P_{w_k} = w_k w_k^T \text{ is the projection matrix}$$

$$\min_{w_1,\cdots,w_d,w_k^Tw_l=\delta_{kl}} \sum_{k=1}^d \sum_{i=1}^n \|x_i - P_{w_k} w_i\|^2$$

Hint for exercise: $\|(I - ww^T)x_i\|^2$

Population covariance matrix $\Sigma = \mathbb{E}x_i x_i^T = ?$, can't compute the integral. Approximate it with $S = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$.

Covariance estimation Given $\{x_i\}_{i=1}^n \sim^{\text{iid}} P_x$ in $\mathbb{R}^p$. Our goal is to estimate

$$\mu = \mathbb{E}x_i, \Sigma = \mathbb{E}(x_i - \mu)(x_i - \mu)^T$$

as the sample mean and covariance

$$\hat{\mu} = \frac{1}{n}\sum_i x_i, \hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^n (x_i - \hat{\mu})(x_i - \hat{\mu})^T$$

Why do we use these as estimators? They are unbiased, but other statistics may be unbiased as well. The reason is that these are max-likelihood estimators(MLE) when the data is distributed as Gaussian. Note that $\frac{1}{n-1}\sum_{i=1}^n (x_i - \hat{\mu})(x_i - \hat{\mu})^T$ is the unbiased estimator, while $\frac{1}{n}\sum_{i=1}^n (x_i - \hat{\mu})(x_i - \hat{\mu})^T$ is the MLE.

Proof sketch: suppose $x_i \sim N(\mu, \Sigma)$. Then

$$p(x_i) = \frac{\exp[-(x_i - \mu)^T \Sigma^{-1}(y_i - \mu)/2]}{(2\pi)^{p/2}|\Sigma|^{1/2}}$$

$$\log p(x_i|\mu, \Sigma) = -\frac{(x_i - \mu)^T \Sigma^{-1}(x_i - \mu)}{2} - \frac{1}{2}\log|\Sigma| + c$$

$$\log p(\{x_i\}_{i=1}^n|\mu,\Sigma) = \log \prod_{i=1}^n p(x_i|\mu,\Sigma)$$

$$= \sum_{i=1}^n \Big\{ -\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) - \frac{1}{2}\log|\Sigma| + c\Big\}$$

$$= nc - \frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^n (x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$$

$\max_\mu \Rightarrow \hat{\mu}^{\text{MLE}} =$ sample mean.

$$= nc - \frac{n}{2}\log|\Sigma| - \frac{n}{2}\text{Tr}(\Sigma^{-1}S) \text{ where } S = \frac{1}{n}\sum_{i=1}^n (x_i-\hat{\mu})(x_i-\hat{\mu})^T$$

$$= |\Sigma| - \frac{n}{2}\text{Tr}(\Sigma^{-1}S)$$

$$\max_\Sigma -\frac{1}{2}\log|\Sigma| - \frac{1}{2}\text{Tr}(\Sigma^{-1}S)$$

$$\max_\Sigma \frac{1}{2}\log|\Sigma^{-1}| - \frac{1}{2}\text{Tr}(\Sigma^{-1}S)$$

$$\max_\Sigma c + \frac{1}{2}\log|\Sigma^{-1}S| - \frac{1}{2}\text{Tr}(\Sigma^{-1}S)$$

Let $B = \Sigma^{-1}S$. $\log|B| - \text{Tr}(B) = \sum_{i=1}^p \log(x_i) - \lambda i$ can be purely written in the eigenvalues of B, which results in $\lambda_i = 1$ means that $B = I$ (is identity). Exercise.

Covariance estimation: asymptotic consistency How well does $S \approx E$ as $\lim_n \to \infty$. (Assume $\mu = 0$.)

$$S = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$$

1. Law of large numbers: $S \to^{n\to+\infty} \mathbb{E}S = \Sigma$

2. rate $\sim n^{-1/2}, S_{p\times p}$

$$\mathbb{E}|S_{kl} - \Sigma_{kl}|^2 \le \frac{c}{n}$$

This result comes from an element-wise LLN:

$$S_{kl} = \frac{1}{n}\sum_{i=1}^n x_i(k)x_i(l) \to^{\text{dist}} N(0,\frac{c}{n})$$

Using Frobenius norm: $\|A\|_{\text{Fro}} = (\sum_{i,j} A_{ij}^2)^{1/2}$

$$\mathbb{E}\|S-\Sigma\|_{\text{Fr}} = \mathbb{E}\sum_{k,l}|S_{kl}-\Sigma_{kl}|^2 \le \frac{c}{n}\cdot p^2$$

What if p is large (e.g. $p \approx n$ or $p \gg n$)? This may be large...

However, if the p is large but the true covariance matrix Σ has low rank, then there may be no curse of dimensionality. For example, consider $\Sigma = uu^T$, where $u = (1,0,0,\cdots,0)$, $x_i = \alpha_i u$,

$\alpha_i = N(0,1)$. In this case, $\mathbb{E}x_i x_i^T = uu^T$, $S = (\frac{1}{n}\sum \alpha_i^2)uu^T$, by LLN we always get the same convergence rate regardless of p.
Noisy PCA: What we observe is that the noisy patches $y_i = x_i + z_i$ for some $x_i \sim P_x$, clean patches, and $z_i \sim N(0, \sigma^2 I)$

Goal: $\Sigma_x = \mathbb{E}x_i x_i^T$. Estimate Σ_x or the principle components of Σ_x.

$$S_y = \frac{1}{n}\sum_{i=1}^{n} y_i y_i^T$$

$$\mathbb{E}S_y = \Sigma_y = \Sigma_x + \alpha^2 I_p, \text{ exercise}$$

If we're in the classical case, where p is fixed, $\hat{\Sigma}_y, \hat{\sigma} \Rightarrow \hat{\sigma}_x = \hat{\sigma}_y - \sigma^2 I$. Inconsistency of $p \approx n, p \gg n$.