CS 189 Lecture Notes, Fall 2020

Machine Learning

Professor: Anant Sahai, Jennifer Listgarten, Jitendra Malik
Vishal Raman

Contents

§1 August 31th, 2020

§1.1 Basics

We would like to understand patterns from data, which means being able to predict. If a pattern exists, it will be confirmed with prediction, to some extent.

Definition 1.1 (Supervised Learning)**.** We have data $\mathbf{x}_i$, outputs y_i. We have a success if given $\mathbf{x}$, not in our data, we can predict $\hat{y}$ which is "correct".

There are two kinds of outputs we can consider:

- Continuous scalars of vectors, which are real or complex valued. We consider problems of this form "regression problems".
- Categorical "quantities", ex. "dog" vs "cat" vs "banana". We consider problems of this form "classification problems". We immediately have the problem of representation for categorical variables.

The usual solution is as follows:

Definition 1.2 (One-hot Encoding)**.** We encode a categorical variable as a vector, where entries correspond to boolean variables on the categories. In the case of binary variables, it suffices to take $y = \pm 1$, for each category.

Example 1.3

$\hat{y} = [1, 0, 0]$ corresponds to Dog, Not Cat, Not Banana.

Question: Can all problems be reduced to regression?
Answer: It turns out that regression-type problems can be mended as optimization problems. For this reason, we try to cast ML as optimization problems, which will sometimes have a regression-type feeling to them.
The levels of an ML problem:

1. Problem: We have data(a collection of (x_i, y_i) pairs) and want to predict $\hat{y}$ with x_i.
2. Model: what kind of pattern are we looking for? (finding all possible patterns is impossibly difficult)
3. Approach: a learning algorithm, which is done by framing and solving an Optimization problem.

Definition 1.4 (Least-Squares Problem)**.** Find $\mathbf{w}$ that minimizes $\|A\mathbf{w} - \mathbf{b}\|^2$.

Example 1.5

We would like to predict the orbit of Cerus. Problem: We have take (x, y) pairs in the plane. Model: Find an elliptical orbit for Cerus.

We model the problem as a Least-Squares problem. We have points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Any conic section is of the form $w[0]x^2 + w[1]y^2 + w[2]xy + w[3]x + w[4]y + w[5] = 1$ (this is a somewhat relaxed approach to the problem, but it is easy to implement).
We find $\arg\min_{\mathbf{w}} \|[x_i^2, y_i^2, x_i y_i, x_i, y_i, 1]\mathbf{w} - [\mathbf{1}]\|^2$. We call the $x_i^2, y_i^2, x_i y_i, x_i, y_i$ features.

§1.2 Features

Definition 1.6 (Features)**.** The components of the pattern that we are trying to learn. It is a mapping from the input data.

Picking the features is where ML is more an art than a science, and usually requires domain knowledge to be successful.

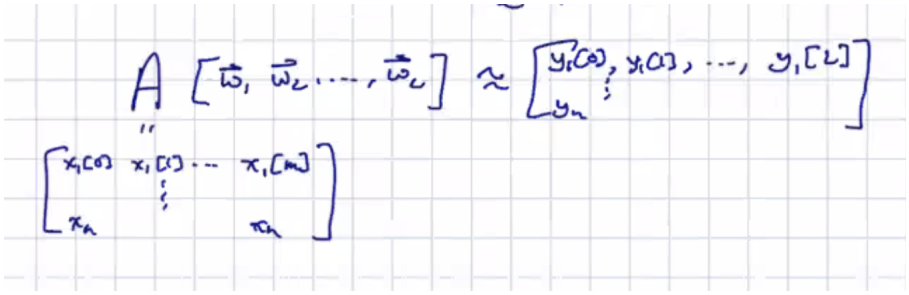
Definition 1.7 (Linear Regression)**.** The desired pattern, a linear combination of features.

§1.3 Classification Comments

Definition 1.8 (Signature Approach)**.** Every class ℓ is associated to a signature $\mathbf{w}_\ell$. Given features $\mathbf{x}$, we compute $\arg\max_\ell \mathbf{x}^T\mathbf{w}_\ell$.

The learning problem is to find the $\mathbf{w}_\ell$. For the binary case, with two signatures, we wish to check $\mathbf{x}^T\mathbf{w}_1 > \mathbf{x}^T\mathbf{w}_2 \Rightarrow \mathbf{x}^T(\mathbf{w}_1 - \mathbf{w}_2) > 0$, so it reduces to a single signature.

We can learn the $\mathbf{w}_\ell$ using one-hot-encoding to set $\mathbf{y}$.



It suffices to take $\hat{w} = (A^TA)^{-1}A^TY$.

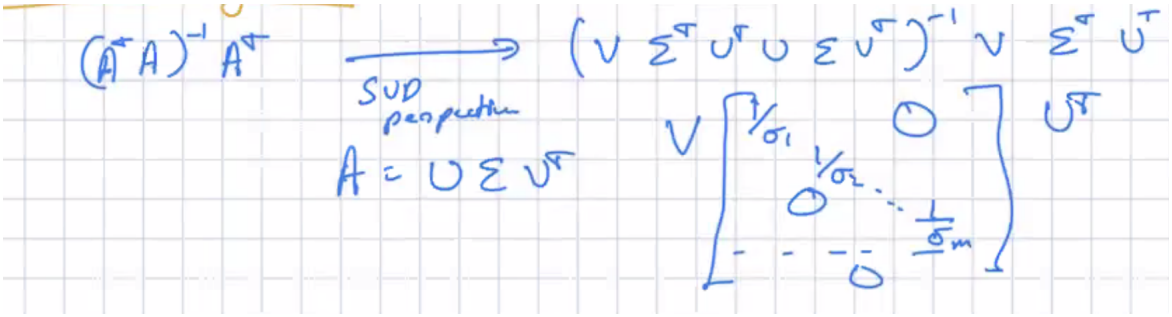
Where do features come from?

- Domain-specific knowledge
- Universal Function Approximation: any function on a compact set can be approximated to any chosen accuracy.

§1.4 Conditioning and Ridge Regularization

What can go wrong with squared-error approaches? The first clear problem is that A^TA might not be invertible.

We take an SVD approach to $(A^TA)^{-1}A^T$: let $A = U\Sigma V^T$.



Notice that we have reciprocals of singular values, which will blow up for small values. This is a problem because the noise in the data will be amplified by very large values.

We could take a dimensionality reduction(PCA) approach to kill off small singular values, or we take a hack: compute $(A^TA + \lambda I)^{-1}A^T$. We now have $\hat{w} = (A^TA + \lambda I)^{-1}A^TY$. We note that

$$\frac{\sigma_j}{\sigma_j^2 + \lambda} = \frac{1}{\sigma_j}\left(\frac{1}{1 + \frac{\lambda}{\sigma_j^2}}\right)$$

which acts as a high-pass filter. This is known as "ridge regularization".

§2 September 2nd, 2020

§2.1 Universal Function Approximation and Features

Recall, we have the goal of going from inputs, $\mathbf{x} \in \mathbb{R}^r$, to predictions, $\hat{y}$. We consider scalar y(we discussed the case of vector y in the previous lecture).

We use a linear parameterization of the patterns we want to learn:

$$y = \sum_{i=1}^{m-1} w[i]\phi_i(\mathbf{x})$$

where ϕ_i is the i-th feature and $\mathbf{w} \in \mathbb{R}^m$. The feature vector is denoted $\phi = \begin{bmatrix} \phi_0 \\ \phi_1 \\ \dots \\ \phi_{m-1} \end{bmatrix}$

The map

$$\mathbf{x} \mapsto \phi(\mathbf{x}),$$

is called a **lifting** if $m > r$ and a **distillation** if $m < r$.

Definition 2.1 (Universal Feature Family)**.** Families that allow (with enough features) arbitrarily good approximations of target patterns.

Some examples include:

- Polynomials,
- Fourier Features,
- Piecewise Linear functions.

Example 2.2 (Feature Growth)

We have the following classes of polynomials.

- **Univariate**: $1, x, x^2, x^3, x^4, \dots, x^d$ which requires $d+1$ features.
- **Bivariate**: $1, x_1, x_2, x_1^2, x_1x_2, x_2^2, \dots$
- **Trivariate**: $1, x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2x_3, x_2^2, x_3^2, \dots$

You can see that there are $\binom{r+d}{r} = \binom{r+d}{d}$ order at-most-d monomials in r variables(consider a stars-and-bars argument). This grows on the order d^r if r is fixed, r^d is d is fixed, or $2^{dH(\frac{r}{d+r})}$ if r is a proportion of d. We see that the number of polynomial features grow with exponential order.

We saw last time that we would like to turn problems into optimization, $\min \|A\mathbf{w} - \mathbf{b}\|^2 : (A^TA)^{-1}A\mathbf{b}$. We have

$$A = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_m(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_n) & \phi_2(x_n) & \dots & \phi_m(x_n) \end{bmatrix}.$$

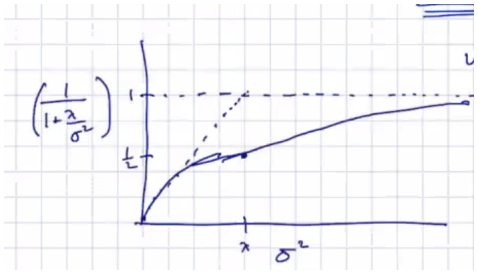
This has dimension n by m. When $m > n$, A^TA is m by m, so it is singular.

§2.2 Ridge Regression

Last time, we were considering $(A^TA)^{-1}A^T$ in the SVD sense, with $A = U\Sigma V^T$. We had a "hack": consider $(A^TA + \lambda I)^{-1}$ instead, which is well defined. On the diagonal, we had values

$$\frac{\sigma_j}{\sigma_j^2 + \lambda} = \frac{1}{\sigma_j}\left(\frac{1}{1 + \frac{\lambda}{\sigma_j^2}}\right).$$

We noticed that this behaves like a "highpass filter".



$$\frac{1}{1 + \frac{\lambda}{\sigma^2}} \approx \begin{cases} \frac{\sigma^2}{\lambda} & \text{if } \sigma^2 << \lambda \\ 1 & \text{if } \sigma^2 >> \lambda \end{cases}$$

When $\sigma_j^2 << k$, then

$$\frac{\sigma_j}{\sigma_j^2 + \lambda} \approx \frac{1}{\sigma_j}\left(\frac{\sigma_j^2}{\lambda}\right) = \frac{\sigma_j}{\lambda}.$$

Ridge regression is a soft way of expression trust is only those directions $\mathbf{v}_j$ in the parameter space that have large enough σ_j. Alternatively, ridge regression doesn't trust variations in $\mathbf{y}$ along $\mathbf{u}_j$(from SVD, associated with the output) with small σ_j.

We are left with the question: how to we choose λ?

Example 2.3

Suppose

$$\mathbf{y} = A\mathbf{w}_{true} = U\Sigma V^T\mathbf{w}_{true} = \sum_i \mathbf{u}_i\sigma_i\mathbf{v}_i^T\mathbf{w}_{true}.$$

Suppose we have $\hat{w} = (A^TA + \lambda I)^{-1}A^T\mathbf{y} = \sum_i v_i\frac{\sigma_i}{\sigma_i^2+\lambda}u_i^T\mathbf{y}$. Instead of getting $\mathbf{v}_i^T\mathbf{w}_{true}$, what do we get?

$$\hat{w} = \frac{\sigma_i}{\sigma_i^2 + \lambda}\mathbf{v}_i\sigma_i\mathbf{u}_i^T\mathbf{w}_{true} = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}\mathbf{v}_i\mathbf{u}_i^T\mathbf{w}_{true},$$

and note that

$$\frac{\sigma_i^2}{\sigma_i^2 + \lambda} = \begin{cases} 1 & \text{if } \sigma_i^2 >> \lambda \\ 0 & \text{if } \sigma_i^2 << \lambda \end{cases}.$$

We see that we pick λ so that it is smaller than σ_i^2 for directions we trust and larger than σ_i^2 for directions we don't trust.

To what optimization problem in $(A^TA + \lambda I)^{-1}A^Tb$ the answer? We know that $(A^TA)^{-1}A^Tb$ optimizes $\arg\min \|Aw - b\|^2$, so what does ridge regression optimize?

We will show that it optimizes

$$\arg\min \|Aw - b\|^2 + \lambda\|w\|^2.$$

Note that

$$\begin{aligned}(Aw-b)^T(Aw-b)+\lambda w^T w &= w^T A^T Aw - b^T Aw - w^T A^T b + b^T b + \lambda w^T w \\ &= w^T(A^T A + \lambda I)w - 2b^T Aw + b^T b.\end{aligned}$$

We take the derivative $\frac{d}{dw}$ to get

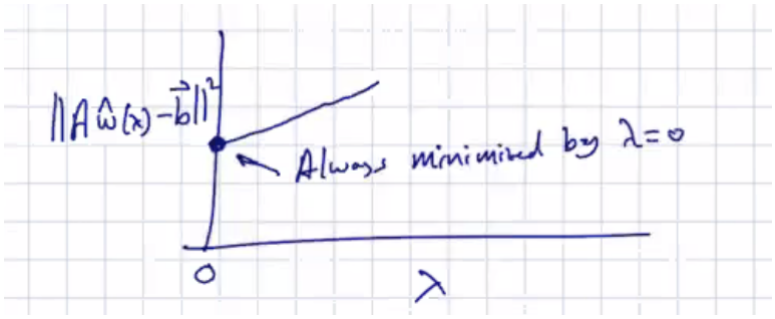
$$-2b^T A + w^T(A^T A + \lambda I) + w^T(A^T A + \lambda I)^T = 2w^T(A^T A + \lambda I) - 2b^T A = 0^T.$$

Hence,

$$w^T = b^T A(a^T A + \lambda I)^{-1} \Rightarrow w = (A^T A + \lambda I)^{-1} A^T b.$$

§2.3 Hyperparameters

Consider the plot of $\|A\hat{w} - b\|^2$ as a function of λ.



We would always want $\lambda = 0$ given the training data. λ is a hyperparameter, where if we tried to optimize it, we would always get 0.

§3 September 9th, 2020

§3.1 Test Sets

§3.2 Validation and Hyper-parameters

§3.3 K-fold Cross-Validation

§3.4 Probability and Toy Models