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 $\arg\min_{\mathbf{w}} \|[x_i^2, y_i^2, x_iy_i, x_i, y_i, 1]\mathbf{w} - [\mathbf{1}]\|^2$. We call the $x_i^2, y_i^2, x_iy_i, x_i, y_i$ 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.

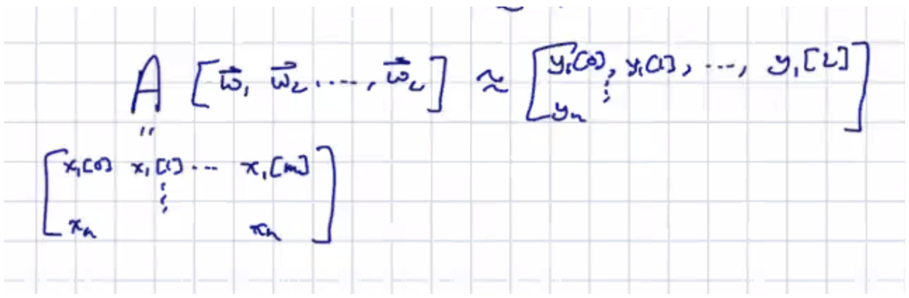
Definition 1.7 (Linear Regression)**.** The desired pattern, a linear combination of features.

§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 $\mathbf{y}$.


$A\,[\vec{w}_1, \vec{w}_2, \dots, \vec{w}_L] \approx \begin{bmatrix} y_1[0], y_1[1], \dots, y_1[L] \\ \vdots \\ y_n \end{bmatrix}$
$\begin{bmatrix} x_1[0] & x_1[1] \cdots & x_1[m] \\ & \vdots & \\ x_n & & x_n \end{bmatrix}$

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

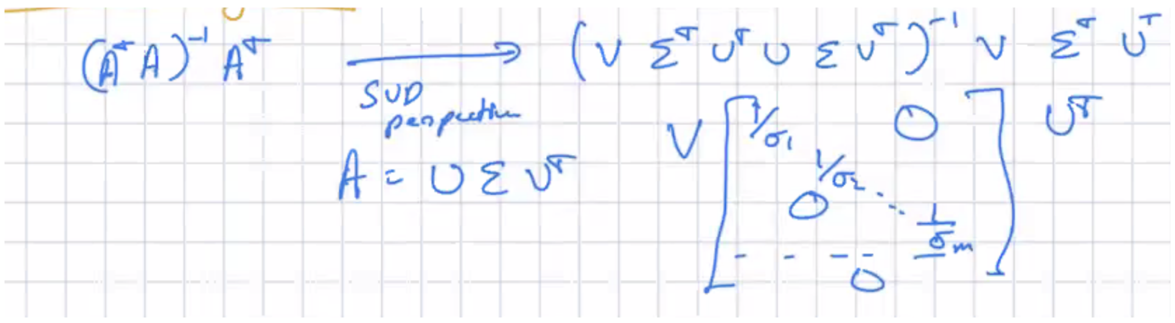
Where do features come from?

- 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 $A = U\Sigma V^T$.


$(A^TA)^{-1}A^T \xrightarrow[\text{SVD perspective}]{} (V\Sigma^TU^TU\Sigma V^T)^{-1}V\Sigma^TU^T$
$A = U\Sigma V^T$
$V\begin{bmatrix} 1/\sigma_1 & & 0 \\ & 1/\sigma_2 \ddots & \\ 0 & & \frac{1}{\sigma_m} \\ - - - & 0 & \end{bmatrix}U^T$

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

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

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