DATA SCIENCE II: Machine Learning MTH 9899 Baruch College

Lecture 1: Introduction to Machine Learning

Adrian Sisser

March 29, 2017

Outline

Supervised Learning

Supervised machine learning consists of learning a function from a set of labeled training examples.

- Generally, You are given input examples AND output values.
- Success can be easily measured through a variety of metrics on in and out-of-sample observations.
- Sometimes, we don't have exact output values, but instead, a notion of 'maximizing' a function (ie Reinforcement Learning).

Unsupervised Learning

In Unsupervised Learning, you're trying to learn a structure that you don't know at the beginning. There are 2 main categories of Unsupervised Learning:

- Clustering – Identify similar/related items based on their features.
    - Identify 'similar' stocks based on returns or other characteristics.
    - Group mortgages together based on geographic data to understand default correlations.
- Latent Variable Models – Identify underlying variables that drive the features you can observe.
    - Latent Variable - A variable who's value is never known, but instead is implied by it's state.

Classification

Identifying which category a variable belongs to. The categories can be:

- **Ordinal** Variables - which have an intrinsic order, ie. credit ratings
- **Categorical** Variables - No implicit ordering, such as what industry a stock belongs to.

Classification Metrics

Reciver Operating Characteristic (ROC) - A graph of the true positive vs false positive rate parameterized by the cutoff used to discriminate between outcomes in a true/false classification.
Confusion Matrix - A table of correct vs incorrect values across all categories.

Regression

Regression refers to prediction a continuous numerical variable - which is what we will focus on in this course. There are a wide variety of different metrics to measure the quality of fit of a regression, each with their own strengths and weaknesses.

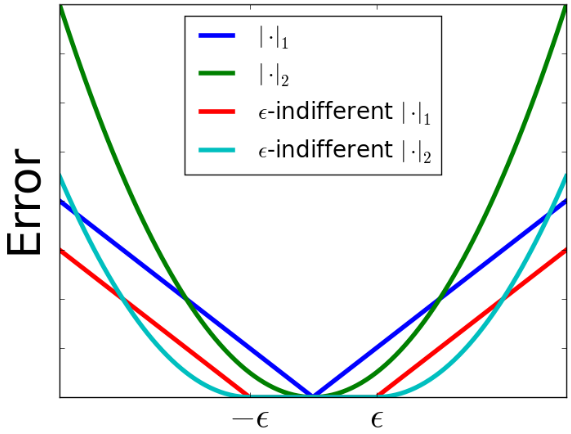
- Mean Squared Error (L2 Norm) - MSE is the most common metric for regression because it's intuitive and very easy to calculate. The problem is, it's not robust to outliers.
- Absolute Error (L1 Norm) - The L1 Norm is a very robust metric that can deal with outliers. Unfortunately, it's very costly to optimize since it's not a convex problem.

Regression

- ϵ Indifferent - This is a metric where we don't penalize for things within some constant, ϵ, of the training value, then apply another metric (ie L1 or L2 norm) to points outside of this area.

Ultimately, the best metric is a tradeoff of computational speed, robustness, and the underlying goal.

Regression


$|\cdot|_1$
$|\cdot|_2$
ϵ-indifferent $|\cdot|_1$
ϵ-indifferent $|\cdot|_2$
Error
$-\epsilon$
ϵ

Regression

You might need to do some cleaning/filtering if you're using an L2 Norm.

- Winsorization - Clip points to a given percentile or number of σ:

$$x_i^{'} = \text{clip}(x_i, \mu_X \pm n\sigma_x)$$

- Median Absolute Deviation (MAD) Filtering - Calculate the MAD - defined as the median of the absolute value of the deviation of every point in a series from the series' median. Then pull in all points to be within n (typically 3 to 5) MADs of the median:

$$\begin{aligned} \text{MAD}_X &= \text{med}\,|x_i - \text{med}\,X| \\ x_i^{'} &= \text{clip}(x_i, \text{med}_X \pm n\,\text{MAD}_X) \end{aligned}$$

Which of these techniques is more robust?

Overview

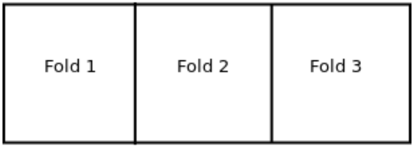
- First, divide your data up into F chunks, or ’folds’
- Cross Validation (CV) refers to fitting your model repeatedly on all but 1 one fold, and testing it on the out-of-sample fold repeatedly across all folds.
- At an extreme, we can perform ‘Leave One Out‘ CV, which is equivalent to N-Fold CV.

Example

Let F_i be the indexes of all rows in Fold i, and $\hat{y}_{i,j}$ be the prediction for row j under a model that was fit excluding rows in F_i.

$$Err = \sum_{i=1..k} \sum_{j \in F_i} (\hat{y}_{i,j} - y_j)^2$$

For example, for the 3-Fold CV shown below, we would fit a model to portions 1 & 2, 2 & 3, 1&3 and use those models to calculate the error on portions 3 & 1 & 2 respectively.



Motivation

- In ML, we face a risk of overfitting our model to the data. CV will help us avoid this, by measuring the performance on data that is not used to build the underlying model.
- Without CV, we will overestimate the accuracy of our models. This is what leads us to adjusted R^2.
- Regularization is a valuable technique we will use, and CV is well-suited to calibrating parameters.

Caveats

- With time series data, you have to be careful. If you divide your data up such that a single time is distributed across multiple folds, you might be using forward looking data!
- Without CV, we will overestimate the accuracy of our models. This is what leads us to adjusted R^2.
- Regularization is a valuable technique we will use, and CV is well-suited to calibrating it's parameters.
- Model Selection - Does adding a new variable really improve our model?

Bias & Variance

Normally, we try to calculate unbiased estimators:

$$E\left[\hat{x}\right] = x \tag{1}$$

Sometimes, we'd rather introduce a bias, if it can reduce the variance of our predictions. We have an inherent *variance* in our predictor, based on the input dataset, which is just a random sample.

Bias vs Variance in Linear Regression

In a traditional linear model (LM), of the form $Y = X\beta + \epsilon$, we assume $\epsilon \sim N(0, \sigma^2)$ and $\hat{\beta} = (X^TX)^{-1}X^TY$. Let's consider our expected squared prediction error for a new sample, x:

$$\begin{aligned}
\mathbb{E}_{\hat{\beta}}[(x\hat{\beta} - y)^2] &= \mathbb{E}[||(x\hat{\beta})^2 - 2x\hat{\beta}y + y^2) \\
&= \mathbb{E}[(x\hat{\beta})^2] - [\mathbb{E}(x\hat{\beta})]^2 + [\mathbb{E}(x\hat{\beta})]^2 + \mathbb{E}[-2(x\hat{\beta})y + y^2] \\
&= Var(x\hat{\beta}) + [\mathbb{E}(x\hat{\beta})]^2 - 2\,\mathbb{E}(x\hat{\beta})y + \mathbb{E}\,y^2 \\
&= Var(x\hat{\beta}) + [\mathbb{E}(x\hat{\beta})]^2 - 2\,\mathbb{E}(x\hat{\beta})(x\beta^* + \epsilon) + \mathbb{E}(x\beta^* + \epsilon)^2 \\
&= Var(x\hat{\beta}) + [\mathbb{E}(x\hat{\beta})]^2 - 2x\beta^*\,\mathbb{E}(x\hat{\beta}) + (x\beta^*)^2 + \mathbb{E}(\epsilon)^2 \\
&= Var(x\hat{\beta}) + (\mathbb{E}(x\hat{\beta}) - x\beta^*)^2 + \sigma^2 \\
&= Var(x\hat{\beta}) + (\mathbb{E}(x\hat{\beta}) - x\beta^*)^2 + \sigma^2 \\
&= \text{var} + \text{bias}^2 + \sigma^2
\end{aligned}$$

Bias vs Variance in Linear Regression

For normal linear regression, $\mathbb{E}\,\hat{\beta} = \beta$, so the bias term can be eliminated:

$$\begin{aligned} \mathbb{E}_{\hat{\beta}}[\|y - \hat{\beta}x\|_2] &= [\mathbb{E}_{\hat{\beta}}[(\hat{\beta}x)^2] - \mathbb{E}_{\hat{\beta}}[\hat{\beta}x]^2] + \epsilon^2 + (\mathbb{E}_{\hat{\beta}}[\hat{\beta}x] - \beta x)^2 \\ &= [\mathbb{E}_{\hat{\beta}}[(\hat{\beta}x)^2] - \mathbb{E}_{\hat{\beta}}[\hat{\beta}x]^2] + \epsilon^2 \end{aligned}$$

Now we cover Ridge Regression, where we'll see that allowing a biased estimator can improve the overall prediction quality.

Ridge Regression

Ridge Regression, also known as Tikhonov Regularization, is an extension of a normal least squares regression.

- We will add a penalty term to our normal linear regression - $\lambda||\beta||_2$
- This forces a tradeoff between the magnitude of the βs and the error terms.
- This is an example of **Regularization**, the notion of adding a penalty to shrink fitted parameters and reduce variance.

Ridge Regression

$$\min_{\beta^R} \|Y - X\hat{\beta}^R\| + \lambda\|\beta^R\|_2$$

$$\begin{aligned}
\mathcal{L} &= \|Y - X\hat{\beta}^R\| + \lambda\|\hat{\beta}^R\|_2 \\
\frac{\partial \mathcal{L}}{\partial \hat{\beta}^R} &= -2Y^TX + 2X^TX\hat{\beta}^R + 2\lambda\hat{\beta}^R \\
\hat{\beta}^R &= (X^TX + \lambda I)^{-1}X^TY
\end{aligned}$$

Regularization penalty

Ridge Regression

As we saw, in Ridge Regression, the estimate of β is no longer unbiased, in fact, we can show that:

$$\begin{aligned}\mathrm{Bias}(\hat{\beta}^R) &= -\lambda(X^TX+\lambda I)^{-1}\beta \\ \mathrm{Var}(\hat{\beta}^R) &= \sigma^2(X^TX+\lambda I)^{-1}X^TX(X^TX+\lambda I)^{-1}\end{aligned}$$

Compare this to OLS:

$$\begin{aligned}\mathrm{Bias}(\hat{\beta}^{LS}) &= 0 \\ \mathrm{Var}(\hat{\beta}^{LS}) &= \sigma^2(X^TX)^{-1}\end{aligned}$$

Ridge Regression

Does it work? Does allowing a biased estimator with a lower variance improve our regression results? Let's see in an IPython Notebook.

Ridge Regression - Choosing λ

So how do we pick a good λ?? Cross-Validation!!

- We test various values of λ and use the value that minimizes the out-of-sample MSE.
- For Ridge, we can do something called ’Generalized Cross Validation’, which is an estimate of leave-one-out validation

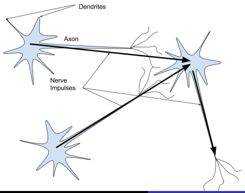
Ridge Regression

A few caveats:

- Ridge Regression is dependent on scale. Since we penalize all β values equally, we need to make sure that all variables are normalized, ie $\mu = 0, \sigma = 1$.
- The λ value scales linearly with the number of points, ie if your sample size doubles, your λ should too.

$$\hat{\beta}^R = (X^T X + \lambda I)^{-1} X^T Y$$

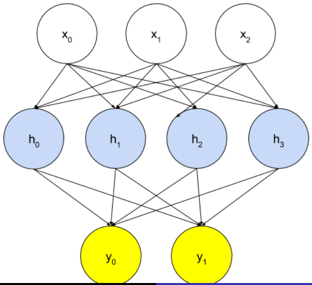
- Neurons are connected to each other in a network.
- They communicate by sending chemical signals between each other, which trigger electrical impulses, that are propagated further down the network.
- Neurons have thresholds of input from their neighbors that must be reached to trigger an output signal. They can be either excitatory or inhibatory.
- By varying the connections between neurons, functions can be learned.



- Neural Networks are a network of artificial neurons, the designs are simplistic versions of the brain.
- An artifical neuron is a node which collects the weighted output of it's neighboring neurons, and sends a transformed version to it's downstream connections. This transformation function is the key.
- Each neuron has an Activation Function, A, which will be applied to teh weighted some of the inputs to determine the output.
- Assume $w_{i,j}$ is the weight applied to the output of neuron i when it is used as the input to neuron j. We can define the output of a neuron as:

$$n_j = A(\sum_i w_{ij} n_i + b_j)$$

- The traditional topology for NNs is feed-forward, where the layers are cleanly separated (ie not interconnected within themselves), and feed data forward towards output nodes.
- There are many new and interesting new topologies that have been studied, which we'll talk about later.


x_0
x_1
x_2
h_0
h_1
h_2
h_3
y_0
y_1