A Machine Learning Refresher

Q & A on the main concepts and terminology

Huascar Sanchez
github.com/hsanchez

Home Research Lab

March 31, 2020

The views expressed do not necessarily reflect the position of my employer.

Q. What is Machine Learning?

Machine Learning (or **ML**) is fitting a function to examples[1] and using that function to generalize and make predictions about new examples.

Machine Learning, by large, falls into three categories:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

And solves two types of problems: Classification and Regression.

[1] An example, or a sample, is a data point that belongs to some data

Q. How do you represent data in ML?

In general, the given data is expressed in a form of a bunch of vectors $\vec{v}_j \in \mathbb{R}^d$ that belong to some high dimensional vector space.

For instance, in image recognition, the vector[2] of an each image is a set of pixels[3] (i.e., a pixelated version of the image).

If you have a notion of distance $\Delta(\vec{v}_i, \vec{v}_j)$, then you can compare which vectors are close to each other in this high dimensional vector space; e.g., the norm $\left\|\vec{v}_i - \vec{v}_j\right\|^2$. See slide 36 for example of distance metrics.

[2] a.k.a., feature vector; it could be numerical (e.g., height of tree) or descriptive (e.g., eye color)

[3] each entry in the vector (e.g., pixel) represents a feature.

Q. What is Supervised Learning?

In **Supervised Learning** (or **SL**), you are given a bunch of examples and their labels (e.g., A or B) and the goal is to classify (or assign), when you are given a new example, to which label we assign the new example.

You could think of these labels the name of the **class or cluster** to which certain portions of the data belong.

Q. Can you give an example of Supervised Learning?

Support vector machines or **SVM**, which goal is to construct the optimal separating *hyperplane* between pieces of data; e.g.,

- Say a hyperplane between clusters of data represented by labels A and B.
- These clusters sit in some high dimensional space, and the idea is to construct a plane that maximizes the margins between the plane and the data.
- If a new datum sits closer to one area of the data, say A, then we assign this new datum to A.

(For historical reasons) This algorithm is called **support vector machines** because the vectors that lie on the margin of the plan are called the *support* vectors.

This is a method for constructing a device to discriminate. If we're having a supervised learning problem then this method gives me an optimal form of discrimination.

Q. What is Unsupervised Learning?

In **Unsupervised Learning** (or **UL**), you are given a bunch of data and you are not told it falls naturally into clusters, but you are not told what the clusters are.

The goal is identify the clusters of data, how many clusters there are, and then be able to assign new things to these different clusters.

Q. Can you give an example of Unsupervised Learning?

Principal Component Analysis (or **PCA**) is a classical UL algorithm.

In **PCA**, the way this works, we construct a *covariance matrix*, and my covariance matrix is just the following object: $C = \sum_j \vec{v}_j \vec{v}_j^{\dagger}$, where $\vec{v}_j^{T}$ is the transpose of $\vec{v}_j$.

- In other words, we construct C from the data by taking these vectors $\vec{v}_j$ and multiply them by their transpose $\vec{v}_j^{\dagger}$.

  E.g., Financial forecasting: the vectors could be, for example, can represent the changes in stock prices over a 24 hr period, and the covariance matrix C would give the correlations (or covariances in the data) between the prices of the different stocks in different times within the 24 hrs period.

In **PCA**, you diagonalize C and say $C = \sum_k P_k \vec{\omega}_k \vec{\omega}_k^{\dagger}$, P_k is piece of the data with size k, and $\vec{\omega}_k$ are the set of vectors you need to find.

If and only if a small set of $P_k >> 0$, then C is effectively low-rank, and the corresponding $\vec{\omega}_k$ are the principal components.

Q. What is Reinforcement Learning?

Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.

In reinforcement learning an algorithm learns to do something by being rewarded for successful behavior and/or being punished for unsuccessful behavior.

Machine learning function fitting

Q. What is Machine learning function fitting?

Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained.

A model that is well-fitted produces more accurate outcomes. A model that is overfitted matches the data too closely. A model that is underfitted doesn't match closely enough.

Q. How do you measure a ML model performance for given data?

We do that by using a cost function.

In ML, a cost function (or loss function) is used to represent how far away a mathematical model is from the real data.

A common way to do this is via the quadratic cost function[4]:

$J(\theta) = \frac{1}{2N} \sum_{n=1}^{N} (h_\theta(x^{(n)}) - y^{(n)})^2$.

We are interested in the parameters that minimize this quadratic cost function. This is called ordinary least squares (OLS).

One adjusts the mathematical model usually by varying parameters within the model, so as to *minimize the cost function*. This is interpreted as given the best model, of its type, that fits the data.

[4] This is just the sum of the squares of the vertical distances between the points and the straight line

Q. What is Regularization?

Regularization is a form of regression, which constrains/regularizes or shrinks the (learned) coefficient estimates towards zero. In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.

Q. What is Gradient Descent?

Gradient descent is an optimization algorithm used to minimize some *convex* function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient.

In ML, we use gradient descent to update the parameters of a ML model.

Start with an initial guess for each parameter θ_k. Then move θ_k in the direction of the slope.

Update all θ_k simultaneously (known as batch gradient descent), and repeat until convergence.

Q. What is Stochastic Gradient Descent?

Similar to batch gradient descent except that you only update using one of the data points each time.

And that data point is chosen randomly. Hence the stochastic.

In other words, stochastic gradient descent pick an n at random and then update using one of the data points. Repeat, picking another data point at random, etc.

Training, Testing and Validation

Q. What is training?

Most ML algorithms need to be trained. That is, you give them data and they look for patterns, or best fits, etc.

They know they are doing well when perhaps a loss function has been minimized, or the rewards have been maximized.

Q. What is an epoch?

In some ML methods, one uses the same training data many times, as the algorithm gradually converges, for example, in stochastic gradient descent. Each time the whole training set of data is used in the training that is called an **epoch or iteration**[5].

One might see a decreasing error as the number of epochs increases. But that doesn't mean your algorithm is getting better, it could easily mean that you are overfitting[6]. To test for this you introduce a test data set, the data that you've held back.

[5] Typically you won't get decent results until convergence after many epochs

[6] This could happen if the learning algorithm has seen the training data so many times or epochs

Q. Any caveats on training and testing?

Over many epochs, if the test error begins to rise (and it's much bigger than the training error) then you have overfitted.

To help avoid overfitting sometimes we divide up our original data into three sets. The third data set is the validation data set. (One can use cross validation – see slide 25 – to evaluate performance of model using training, test, and validation data sets.)

Q. What is Bias and Variance?

Bias is how far away (or error) the trained model is from the correct result *on average*. Where "on average" means over many goes at training the model, using different data:

$$Bias(\hat{f}(x')) = \underbrace{\mathbb{E}[\hat{f}(x)]}_{\text{Average error}} - f(x')$$

Variance is a measure of the magnitude of that error.

$$Var(\hat{f}(x')) = \mathbb{E}[\hat{f}(x)^2] - \mathbb{E}[\hat{f}(x')]^2$$

We often find there is a trade-off between bias and variance. As one is reduced, the other is increased[7].

[7] This the matter of over-and-underfitting

Q. What is Overfitting and Underfitting?

Overfitting is

Underfitting is

Algorithms questions, please!

Q. What is linear regression?

Linear regression attempts to model the relationship between two continuous variables – a scalar response (or dependent variable) and and one or more explanatory variables (or independent variables) – by fitting a linear equation to observed data.

The governing equation for linear regression is also quite simple. It takes the form: $y = Bx + A$ where y is the dependent variable, x is the independent variable, and A and B are coefficients dictating the equation

e.g. a model that assumes a linear relationship between the input variables x and the single output variable y.

Q. What is the general form of multiple regression?

The general form of the equation for linear regression.

Evaluating algorithms!

Q. What is cross-validation?

Cross-validation is a technique for assessing how well a model performs on new independent data. (how robust the model is.)

The simplest example of cross-validation is when you split your data into two groups[8]: (1) training data, and (2) testing data.

One uses training data to build the model and testing data to test the model.

[8] e.g., a ˜60%-˜40% split

Q. How to define/select metrics?

There isn't a one-size-fits-all metric.

The metric(s) chosen to evaluate a ML model depends on various factors:

Is it a regression or classification task? E.g., MSE vs Accuracy.

What is the business objective? E.g., precision vs recall.

What is the distribution of the target variable?

Examples of other metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared, Confusion Matrix and related metrics (Precision, Recall, Accuracy).

Q. Can you explain what precision and recall are?

Recall is a measure of completeness or quantity, whereas

Precision is a measure of exactness or quality:

$$\underbrace{Recall = \frac{TP}{TP + FN}}_{\text{What proportion of actual positives was identified correctly?}} \qquad \underbrace{Precision = \frac{TP}{TP + FP}}_{\text{What proportion of positive identifications was actually correct?}}$$

In simple terms, **high** precision means your algorithm has returned substantially *more relevant results than irrelevant ones*, while **high** recall means your algorithm has returned *most of the relevant results*.

Q. What is a confusion matrix?

A **Confusion Matrix** is a simple way of understanding **how well an algorithm is doing at classifying data**. It is just the idea of **false positives** and **false negatives**

		Predicted	
		Yes	No
Actual	Yes	TP	FN
	No	FP	TN

Q. Can you explain what are false positives and false negatives?

There are two errors that often rear their head when you are learning about hypothesis testing —- false positives and false negatives, technically referred to as type I error and type II error respectively.

False positives are incorrect classifications of the presence of a condition when it is actually absent. A **false positive** is when you reject a true null hypothesis.

False negatives are incorrect classifications of the absence of a condition when it is actually present. A **false negative** is when you accept a false null hypothesis.

Q. Provide examples when false positives are more important than false negatives, false negatives are more important than false positives and when these two types of errors are equally important

E.g., cancer screening. It is much worse to say that someone doesn't have cancer when they do (false negative) rather than saying that someone has it when in fact they don't (false positive).

E.g., (from a psychological perspective) pregnancy test. Given the null hypothesis: "I am not pregnant." A false negative for someone who really does not want a child, is not ready for one and when assuring themselves with a negative result.

E.g., forecasting (health weather). During the covid-19 outbreak. Tracking rate of false positives and false negatives in March 2020 in the United States of America.

Q. Can you list some of metrics derived from Confusion Matrix?

Confusion matrix related metrics

- Accuracy rate: $\frac{TP+TN}{Total}$ where Total is $TP + TN + FP + FN$,
- Error rate: $1 - \frac{TP+TN}{Total}$,
- False positive rate: $\frac{FP}{TN+FP}$,
- Recall: $\frac{TP}{TP+TN}$, Precision: $\frac{TP}{TP+FP}$,
- Specificity: $1 - \frac{FP}{TN+FP}$,
- Prevalence: $\frac{TP+FN}{Total}$

Q. How do you measure how good your classification algorithm is? You have unbalanced numbers, with our class being much larger or smaller than others

You use the Matthews correlation coefficient. The number it yields is between plus or minus one. Plus one means perfect prediction, zero means no better than random.

$$\frac{TPxTN - FPxFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Q. Is there another way to measure how well an algorithm is classifying data? Also given an example

Another way of looking at how well an algorithm is classifying is the **receiver operating characteristic** or **ROC** curve.

E.g., suppose there is a threshold (or parameter) in your classification algorithm that determines whether you have an apple or a non apple object.

- Plot the *true positive* rate against the *false positive* rate as the threshold (or parameter) varies. Points above the 45° diagonal are good, and the further into the top left of the plot the better.

- The area under the ROC curve (the **AUC**[9]) is then a measure of how good different algorithms are. The closer to one (the max possible) the better.

[9] AUC is used to rank kaggle competitions

Miscellaneous questions, please!

Q. What is the curse of dimensionality?

This phenomena involves data in high dimensions.

Suppose you are working in M dimensions, that is you data has M features. And suppose you have N data points. Having a large number of data points is good, the more the merrier. BUT what about number of features?

Think how these N data points might be distributed in M dimensions. Suppose that the numerical data for each feature is 0 or 1. There will therefore be 2^M possible combinations.

If N is less than 2^M then you run the risk of having every data point being on a different corner of the M-dimensional hypercube.

Q. Why is it important about measuring distances?

Because the shorter the distance between two feature vectors the closer in character are the two samples they represent.

There are many ways to measure distance:

- Manhattan distance (L^2 norm).The sum of the absolute values of the different between entries in the vector. (Preferred distance when dealing with data in high dimensions.)
- Euclidean distance (L^2 norm). Square the distances between vector entries, sum these and square root.
- Cosine similarity. Cosine of the angle between two vectors[10]. Just take the *dot* product of two vectors and divide by the two lengths.

[10] Two vectors might be similar if they are pointing in the same direction even if they are of different lengths.