Shams S
Research Assistant @ Purdue University
Apr 10•18 min read

Image Source: http://cfss.uchicago.edu/persp009_svm_files/figure-html/hyperplane-1.png

Support Vector Machine

A SVM is a discriminative classifier formally defined by a separating hyperplane. Given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples.

· · ·

Basics of Machine Learning Series

> ***Index***

· · ·

Optimization Objective

The support vector machine objective can seen as a modification to the cost of logistic regression. Consider the sigmoid function, given as,

where

The cost function of logistic regression as in the post **Logistic Regression Model**, is given by,

Each training instance contributes to the cost function the following term,

So when , the contributed term is , which can be seen in the plot below. The cost function of SVM, denoted as , is a modification the former and a close approximation.

Fig-1. SVM Cost function at y = 1

Fig-1. SVM Cost function at y = 1

Fig-1. SVM Cost function at y = 1

```
import numpy as npdef svm_cost_1(x):    return np.array([0 if _ >= 1 else -0
```

Similarly, when , the contributed term is , which can be seen in the plot below. The cost function of SVM, denoted as , is a modification the former and a close approximation.

Fig-2. SVM Cost function at y = 0

Fig-2. SVM Cost function at y = 0

Fig-2. SVM Cost function at y = 0

```
def svm_cost_0(x):    return np.array([0 if _ <= -1 else 0.26*(_ + 1) for _
```

> ***While the slope the straight line is not of as much importance, it is the linear approximation that gives SVMs computational advantages that helps in formulating an easier optimization problem.***

Regularized version of can from the post **Regularized Logistic Regression** can rewritten as,

In order to come up with the cost function for the SVM, is modified by replacing the corresponding cost terms, which gives,

Following the conventions of SVM the following modifications are made to the cost in , which effectively is a change in notation but not the underlying logic,

- removing does not affect the minimization logic at all as the minima of a function is not changed by the linear scaling.
- change the form of parameterization from to where it can be intuitively thought that .

After applying the above changes, gives,

The SVM hypothesis does not predict probability, instead gives hard class labels,

Large Margin Intuition

Fig-3. SVM Cost function plots

Fig-3. SVM Cost function plots

Fig-3. SVM Cost function plots

According to and the plots of the cost function as shown in the image above, the following are two desirable states for SVM,

- if , then (not just)
- if , then (not just)

Let C in be a large value. Consequently, in order to minimize the cost, the corresponding term must be close to 0.

Hence, in order to minimize the cost function, when , should be 0, and similarly, when , should be 0. And thus, from the plots in Fig.3, it is clear that it can only fulfilled by the two states listed above.

Following the above intuition, the cost function can we written as,

subject to contraints,

What this basically leads to is the selection of a decision boundary that tries to maximize the margin from the support vectors as shown in the plot below. This maximization of the margin as seen for decision boundary A increases the robustness over decision boundaries with lesser margins like B. And it is this property of the SVMs that attributes the name **large margin classifier** to it.

Fig-4. Large Margin Decision Boundary

Fig-4. Large Margin Decision Boundary

Fig-4. Large Margin Decision Boundary

Effect of Parameter C

Fig-5. Effect of Parameter C

Fig-5. Effect of Parameter C

Fig-5. Effect of Parameter C

As discussed in the section above, the effect of C can be considered as reciprocal of regularization parameter, . This is more clear from Fig-5. A single outlier, can make the model choose the decision boundary with smaller margin if the value of C is large. A small value of C ensures that the outliers are overlooked and best approximation of large margin boundary is determined.

Mathematical Background

Vector Inner Product: Consider two vectors, and , given by,

Then, the **inner product** or the **dot product** is defined as .

Norm of a vector, , denoted as is the euclidean length of the vector given by the pythagoras theorem as,

The inner product can also be defined as,

where can be described as the projection of vector onto vector which can be either positive or negative signed based on the angle between the vectors as shown in the image below.

Fig-6. Dot Product

Fig-6. Dot Product

Fig-6. Dot Product

SVM Decision Boundary: From , the optimization statement can be written as,

subject to contraints,

Let and , i.e. number of features is 2 for simplicity, then can be written as,

Using , in can be written as,

The plot of can be seen below,

Fig-7. Dot Product in SVM

Fig-7. Dot Product in SVM

Fig-7. Dot Product in SVM

Hence, using and , the optimization objective in and the constraints in are written as,

subject to contraints,

where is the projection of onto vector .

Consider two decision boundaries, A and B, and their respective perpendicular parameters, and as shown in the plot below. As a consequence of choosing for simplification, all the corresponding decision boundaries pass through the origin.

Fig-8. Choosing Large Margin Classifier

Fig-8. Choosing Large Margin Classifier

Fig-8. Choosing Large Margin Classifier

Based on the two training examples of either class chosen, close to the boundaries, it can be seen that the magnitude of projection is more in case of than . This basically tells that it would be possible to choose smaller values of and satisfy and if the value of projection is bigger and hence, the decision boundary, B is more favourable to the optimization objective.

Why is decision boundary perpendicular to the ?

Consider two points and on the decision boundary given by,

Since the two points are on the line, they must satisfy . Substitution leads to the following,

Subtracting from ,

Since and lie on the line, the vector is on the line too. Following the property of orthogonal vectors, is possible only if is orthogonal or perpendicular to , and hence perpendicular to the decision boundary.

Kernels

When dealing with non-linear decision boundaries, a learning method like logistic regression relies on high order polynomial features to find a complex decision boundary and fit the dataset, i.e. predict if,

where .

A natural question that arises is if there are choices of better/different features than in ? A SVM does this by picking points in the space called **landmarks** and defining functions called **similarity** corresponding to the landmarks.

Fig-9. SVM Landmarks

Fig-9. SVM Landmarks

Fig-9. SVM Landmarks

Say, there are three landmarks defined, , and as shown in the plot above, the for any given x, , and are defined as follows,

Here, the similarity function is mathematically termed a **kernel**. The specific kernel used in is called the . Kernels are sometimes also denoted as .

Consider from . If there exists close to landmark , then and hence, . Similarly for a far from the landmark, will be a larger value and hence exponential fall will cause . So effectively the choice of landmarks has helped in increasing the number of features had from 2 to 3. which can be helpful in discrimination.

For a gaussian kernel, the value of defines the spread of the normal distribution. If is small, the spread will be narrower and when its large the spread will be wider.

Also, the intuition is clear about how landmarks help in generating the new features. Along with the values of parameter, and , various different decision boundaries can be achieved.

How to choose optimal landmarks?

In a complex machine learning problem it would be advantageous to choose a lot more landmarks. This is generally acheived by choosing landmarks at the point of the training examples, i.e. landmarks equal to the number of training examples are chosen, ending up in if there are training examples. This translates to the fact that each feature is a measure of how close is an instance to the existing points of the class, leading to generation of new feature vectors.

> ***For SVM training, given training examples, , features are computed, and , if***

The training objective from is modified as follows,

In this case, in by the virtue of procedure used to choose .

> ***The regularization term in can be written as . But in practice most SVM libraries, instead , which can be considered a scaled version is used as it gives certain optimization benefits and scaling to bigger training sets, which will be taken up at a later point in maybe another post.***

While the kernels idea can be applied to other algorithms like logistic regression, the computational tricks that apply to SVMs do not generalize as well to other algorithms.

> ***Hence, SVMs and Kernels tend to go particularly well together.***

Bias/Variance

Since ,

- Large C: Low bias, High Variance
- Small C: High bias, Low Variance

Regarding ,

- Large : High Bias, Low Variance (Features vary more smoothly)
- Small : Low Bias, High Variance (Features vary less smoothly)

Choice of Kernels

- **Linear Kernel:** is equivalent to a no kernel setting giving a standard linear classifier given by,

Linear kernels are used when the number of training data is less but the number of features in the training data is huge.

- **Gaussian Kernel:** Make a choice of to adjust the bias/variance trade-off.

Gaussian kernels are generally used when the number of training data is huge and the number of features are small.

> ***Feature scaling is important when using SVM, especially Gaussian Kernels, because if the ranges vary a lot then the similarity feature would be dominated by features with higher range of values.***

> ***All the kernels used for SVM, must satisfy Mercer's Theorem, to make sure that SVM optimizations do not diverge.***

Some other kernels known to be used with SVMs are:

- Polynomial kernels,
- Esoteric kernels, like string kernel, chi-square kernel, histogram intersection kernel, ..

Multi-Class Classification

- Most SVM libraries have multi-class classification.
- Alternatively, one may use one-vs-all technique to train different SVMs and pick class with largest

Logistic Regression vs SVM

- If is large relative to , use logistic regression or SVM with linear kernel, like if
- If is small and is intermediate, use SVM with gaussian kernel, like if
- If is small and is large, create/add more features, then use logistic regression or SVM with no kernel, as with huge datasets SVMs struggle with gaussian kernels, like if

> ***Logistic Regression and SVM without a kernel (with linear kernel) generally give very similar. A neural network would work well on these training data too, but would be slower to train.***

Also, the optimization problem of SVM is a convex problem, so the issue of getting stuck in local minima is non-existent for SVMs.

REFERENCES:

Machine Learning: Coursera - Optimization Objective
Machine Learning: Coursera - Large Margin Intuition
Machine Learning: Coursera - Mathematics of Large Margin Classification
Machine Learning: Coursera - Kernel I
Machine Learning: Coursera - Kernel II
Machine Learning: Coursera - Using An SVM
Quora - Why is theta perpendicular to the decision boundary?
Introduction to support vector machines

machine-learning andrew-ng

· · ·

Error Metrics for Skewed Classes and Using Large Datasets

K-Means Clustering