Machine Learning

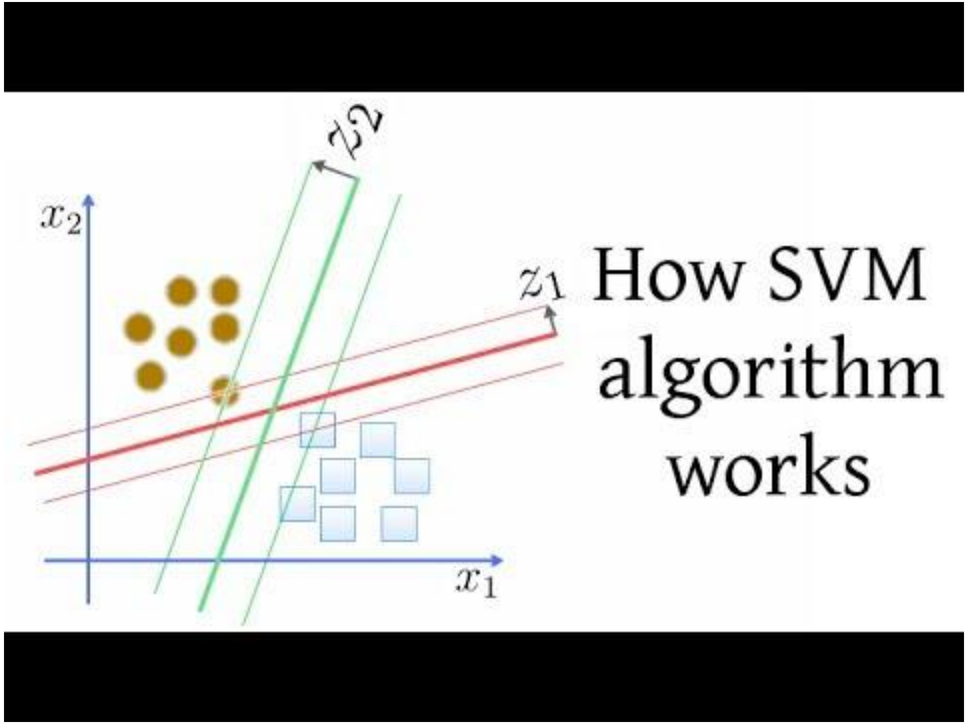
SVMs
Clustering

Dom Huh

What is Support Vector Machine?

“In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. A SVM constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively , a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class.”

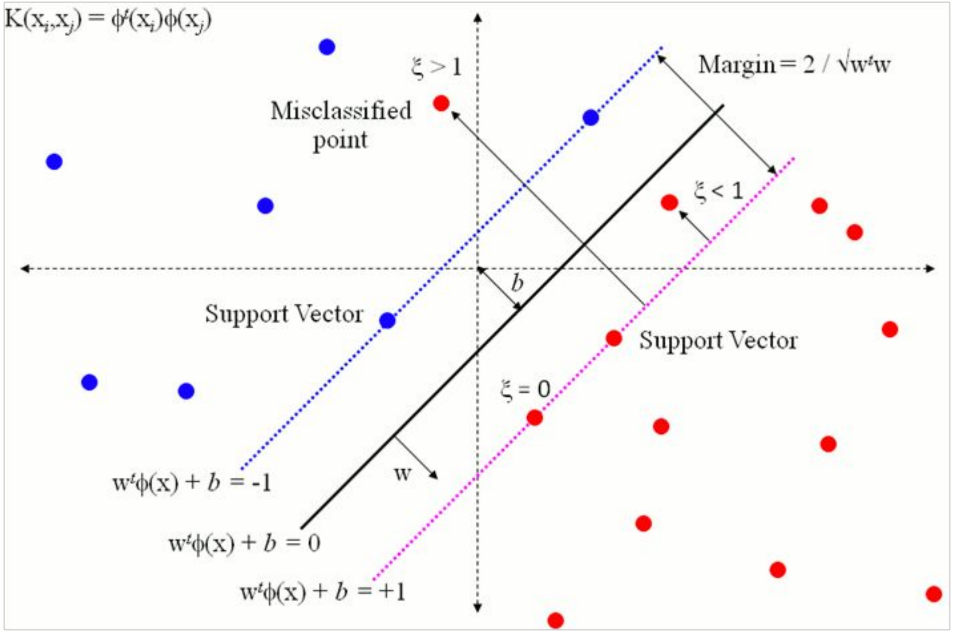
Video



Linear SVM in R2

Classification by separation using hyperplanes with maximized width/margin through constrained optimization of hinge loss (Optimization Objective)

Boundaries, Decision Boundary



Concepts of SVM

What is it?

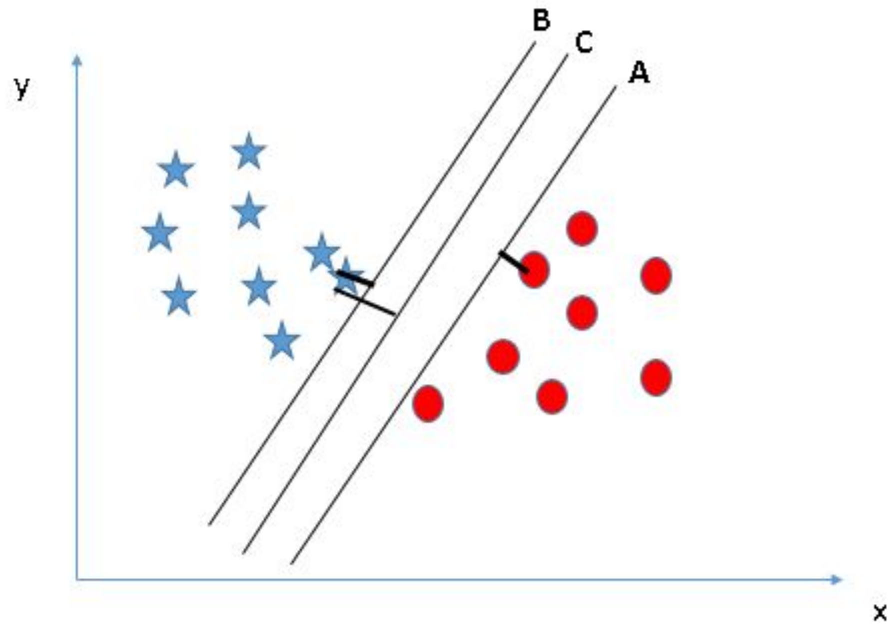
Mathematical Understanding

- Vectors
- Hinge Loss
- Convex optimization

Kernels

Patrick Winston’s lecture on SVMs

https://www.youtube.com/watch?v=_PwhiWxHK8o



SVMs in Higher Dimensions

Use kernels

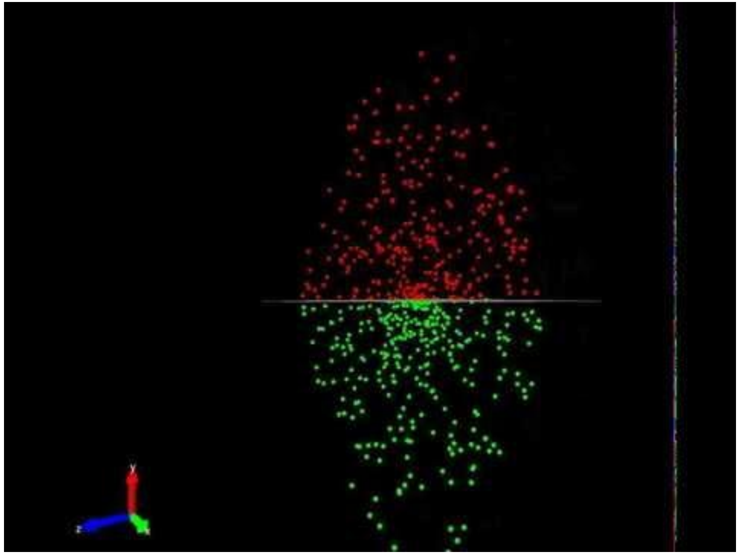
Time complexity is increased(not much)

clf = sklearn.svm.SVC(kernel=”k_type”)

clf.predict(data)

SVC - Support Vector Classifier

*SVR is not commonly used (for regression)



Kernels Operation

Linear

RBF

Radial basis function

Poly

Sigmoid

Gaussian, String, Path, Tree, Graph...

Code to visualize:

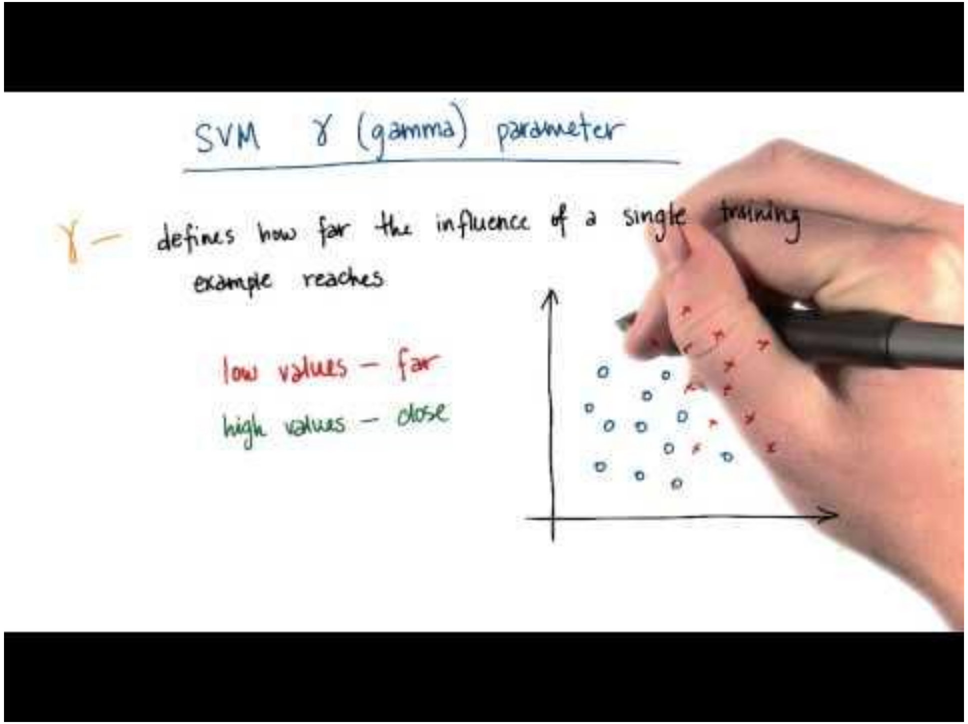
http://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html

RBF

Gamma parameter (similar to C parameter)

Lower = Less complexity

Larger = More complexity



Soft vs Hard Margins

How much to penalize wrong classification?

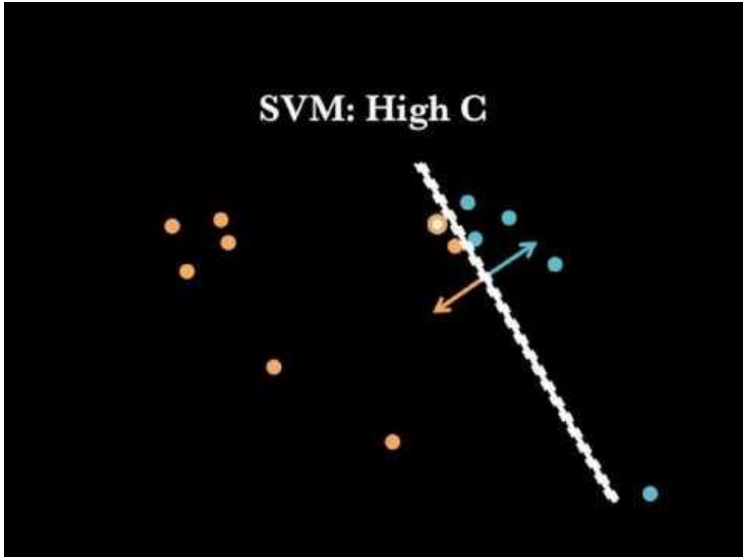
Soft margin prioritizes simplicity

Hard margin prioritizes accuracy

Overfitting

C parameter:

sklearn.svm.SVC(C=0.0001) vs sklearn.svm.SVC(C=1000)



Multiclass SVM

Decision Boundaries:

OVR:

One vs Rest - Fewer classifications, imbalanced classes

OVO:

One vs One - More classification, lower sensitivity to imbalances, more computationally expensive

SVM implementation

http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

Create SVM class

https://github.com/scikit-learn/scikit-learn/tree/master/sklearn/svm
https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/svm/classes.py
https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/svm/base.py
https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/svm/bounds.py

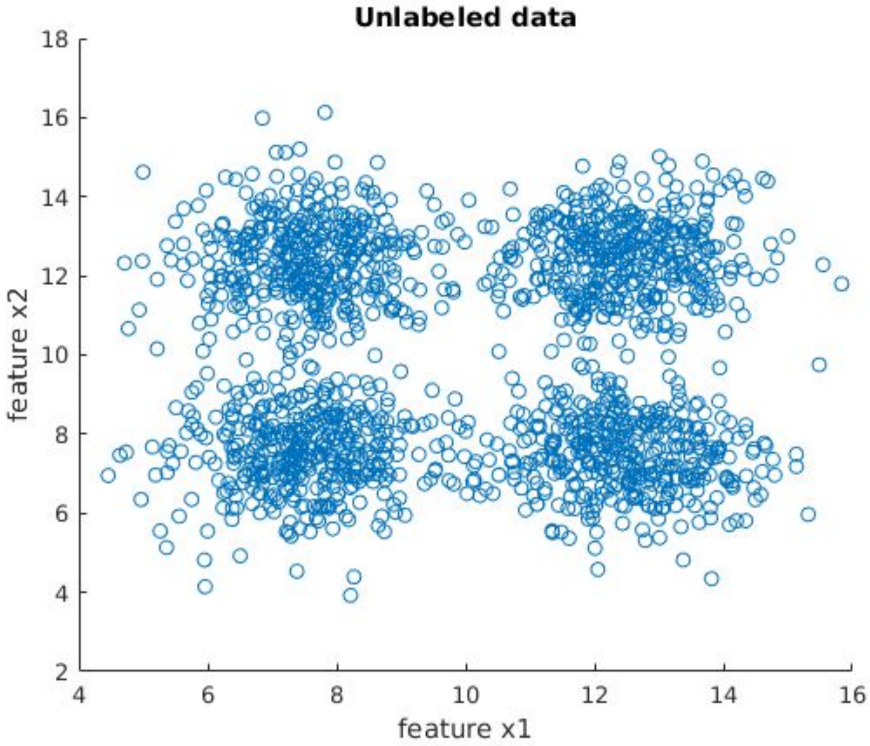
Clustering

Minimize dissimilarity function, making the constraint of how many clusters or minimum distance between each data point.

Dissimilarity- Non-normalized variances

Hierarchical clustering

Flat clustering



Agglomerative hierarchical clustering

1) Make each data point as a cluster
2) Find two most similar clusters and cluster them
3) Then keep going until the all items are cluster together

Use stopping criteria before clustering all of them together

Dendrogram allows us to visualize the process

What is our metric of distance?

Linkages: really slow, what is the runtime?

Single linkage- distance between the closest points from one cluster to the other

Complete linkage - distance of farthest points

Average linkage - distance of average points

Flat clustering with K-Mean

Grouping the data by

First randomly placing k centroids in the data.

Group the closest data points together. Then, move the centroids to the average of the joined group. Repeat this section until the location of the centroids doesn’t change.

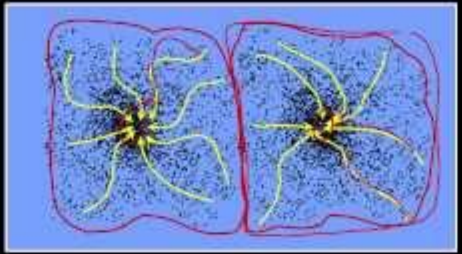
K: How many clusters (should be less than size of dataset)

Visualization:
https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

More detail: https://nlp.stanford.edu/IR-book/pdf/16flat.pdf

Mean shift clustering

- Cluster: all data points in the *attraction basin* of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



Optimization Objective of K-Mean

Minimize the squared distances of the average and centroid location, known as distortion cost function.

More in-depth look into optimization

http://kldavenport.com/the-cost-function-of-k-means/

https://cseweb.ucsd.edu/~dasgupta/291-geom/kmeans.pdf

https://pdfs.semanticscholar.org/8276/580b77bac4575e12186ceaf510a8b4f4bbf6.pdf

Random initialization cluster centroid

Avoid local optima of distortion cost function

Choice 1:

Pick two random points of training set

Choice 2:

Run K-Means multiple times with random initializations (50-100), and compute distortion function. Find the lowest cost.

Choosing K

Choice 1: Visualization/Manually/Apriori

Choice 2: Elbow method- Find distortion of varying k, and use the least k to cost ratio. Shortcomings: Impractical

<u>Choice 3: Based on what the problem you are trying to solve</u>

More in-depth look into Elbow Method:

https://en.wikipedia.org/wiki/Elbow_method_(clustering)

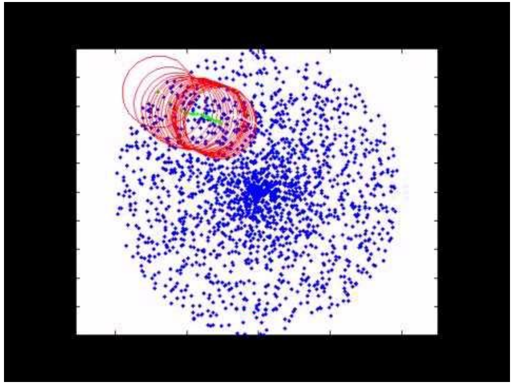
K-Mean implementation

Create K-Mean class

https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/cluster/k_means_.py

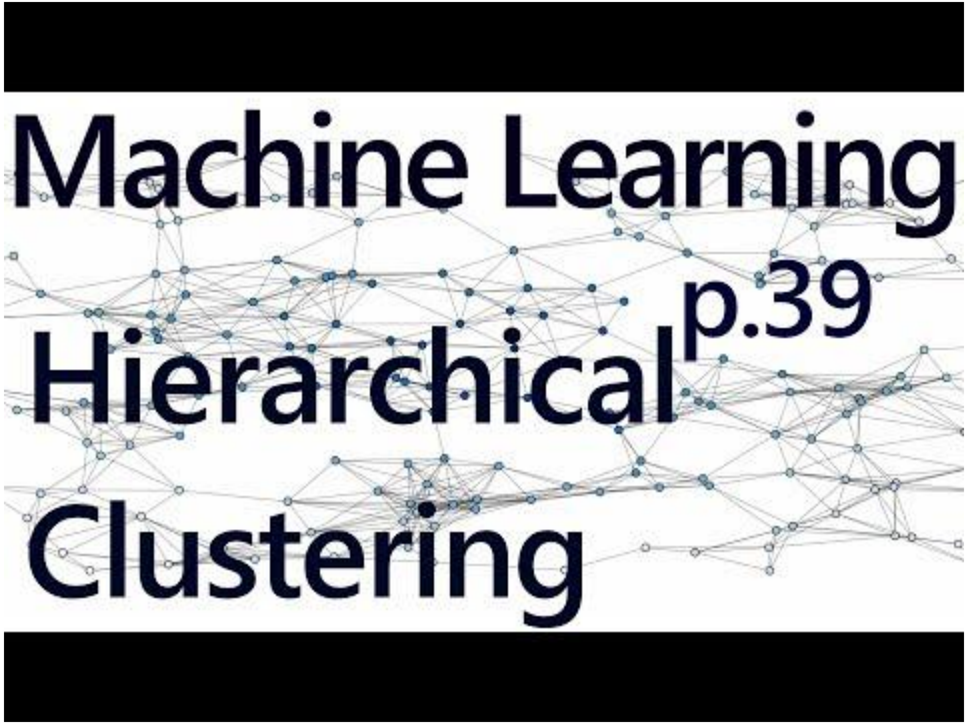
Mean Shift

Finding modes, or local maxima, via density estimation, and segment the data into clusters based on attraction basin, or the region with highest concentration



More in-depth look into Mean Shift:
http://vision.stanford.edu/teaching/cs131_fall1314_nope/lectures/lecture13_kmeans_cs131.pdf

Mean Shift



Mean-Shift implementation

Create Mean-Shift Class

https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/cluster/mean_shift_.py

Choice:

More regression/classification/clustering algorithms