Introduction to Support Vector Machines

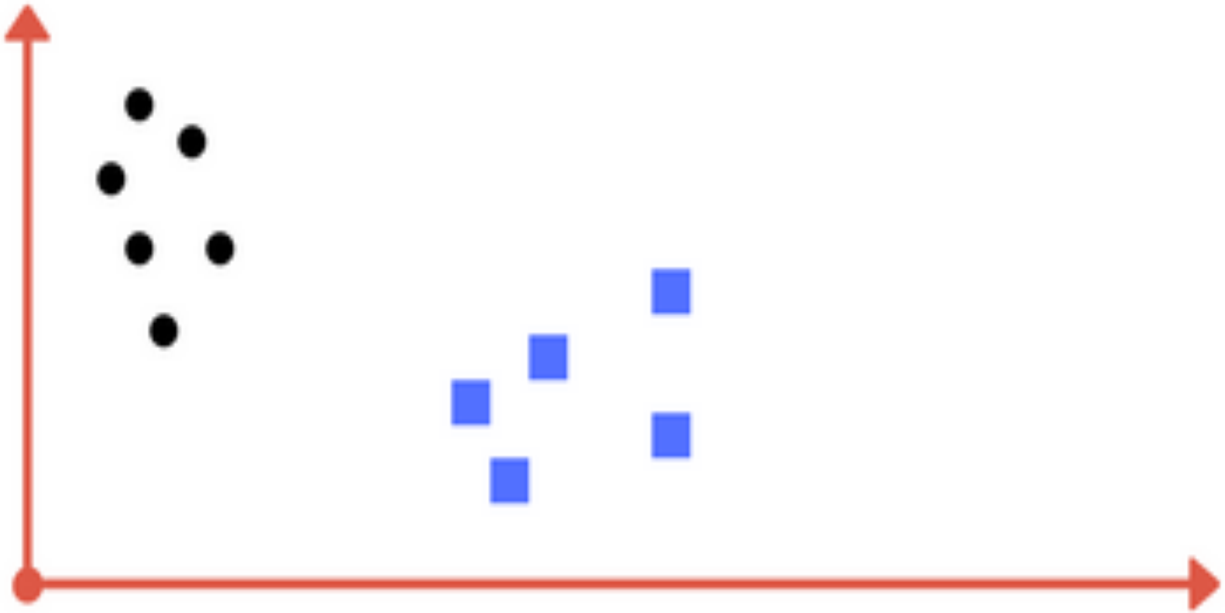
Kazi Aminul Islam

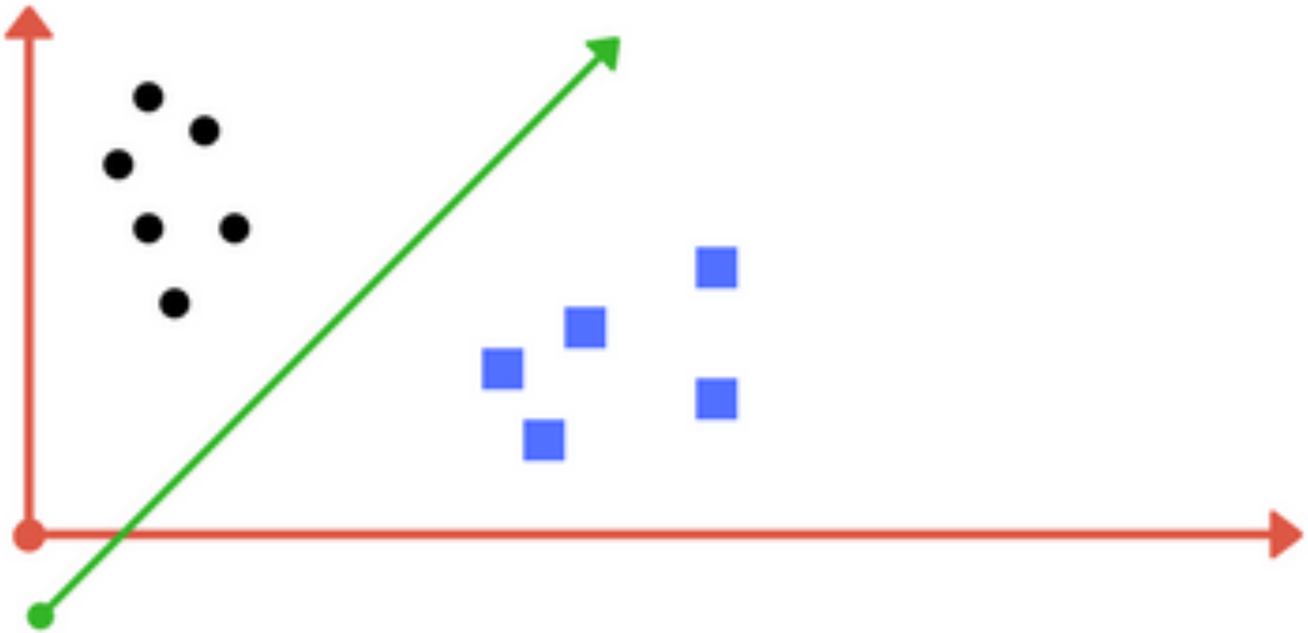
Department of Computer Science

Kennesaw State University

A Linear Classifier

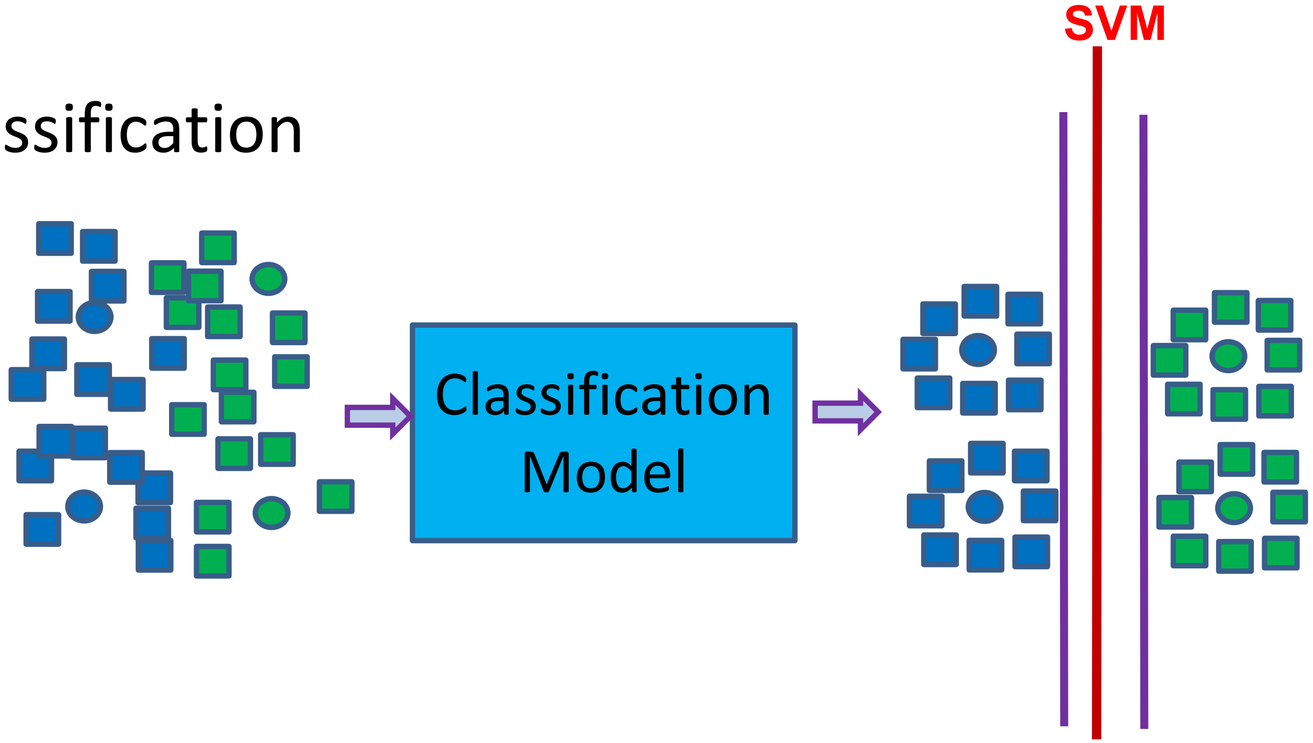
Draw a line (hyperplane) that separates black circles and blue squares





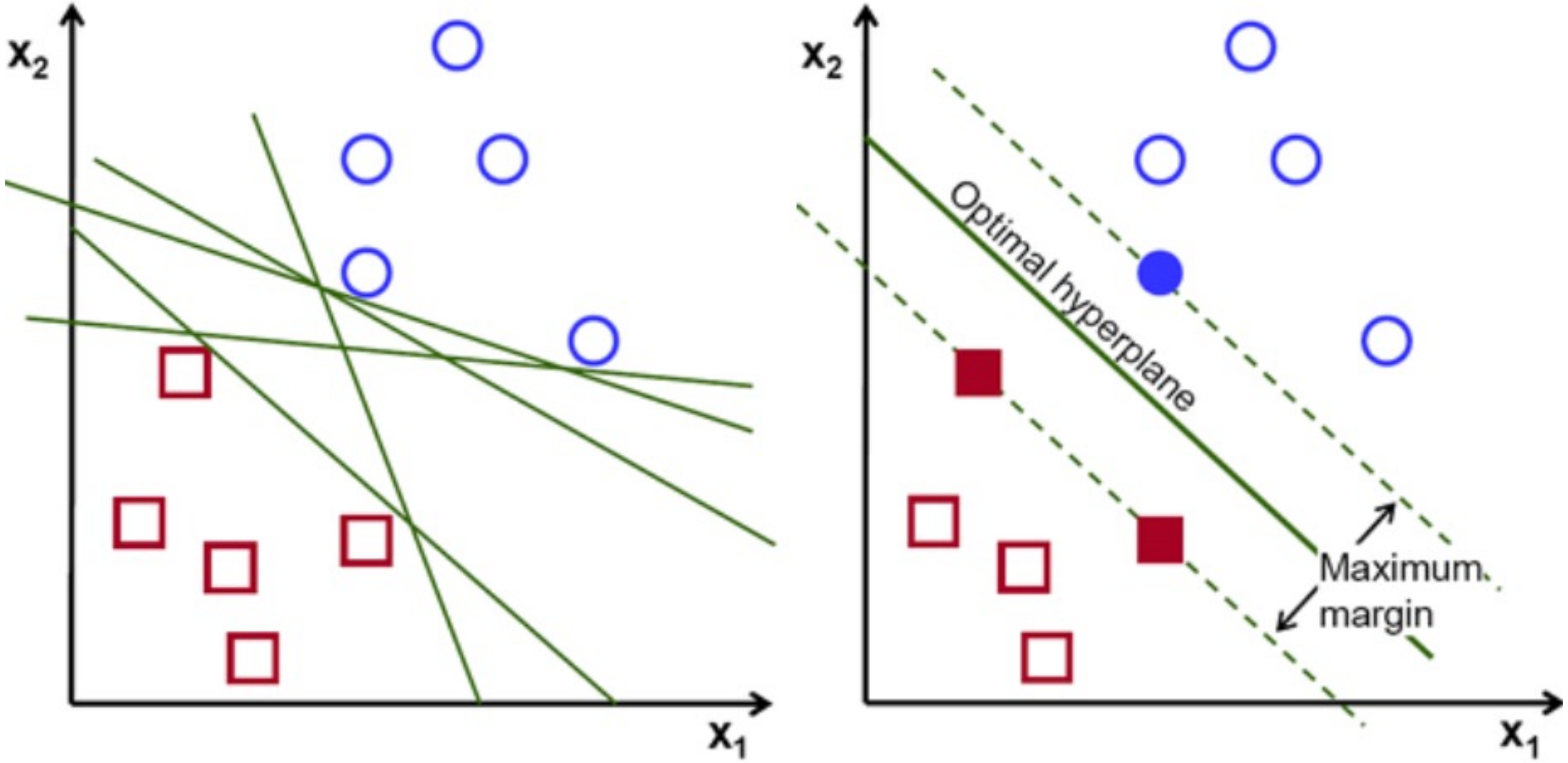
Why Support Vector Machine (SVM)

- Support Vector Machine (SVM) is a widely-used supervised machine learning algorithm.
- Support vector machine gives a maximum margin to separate classes
- State-of-the-art model for Classification



Why Support Vector Machine (SVM)

- Support Vector Machine (SVM) finds optimum hyperplane with maximum separations.

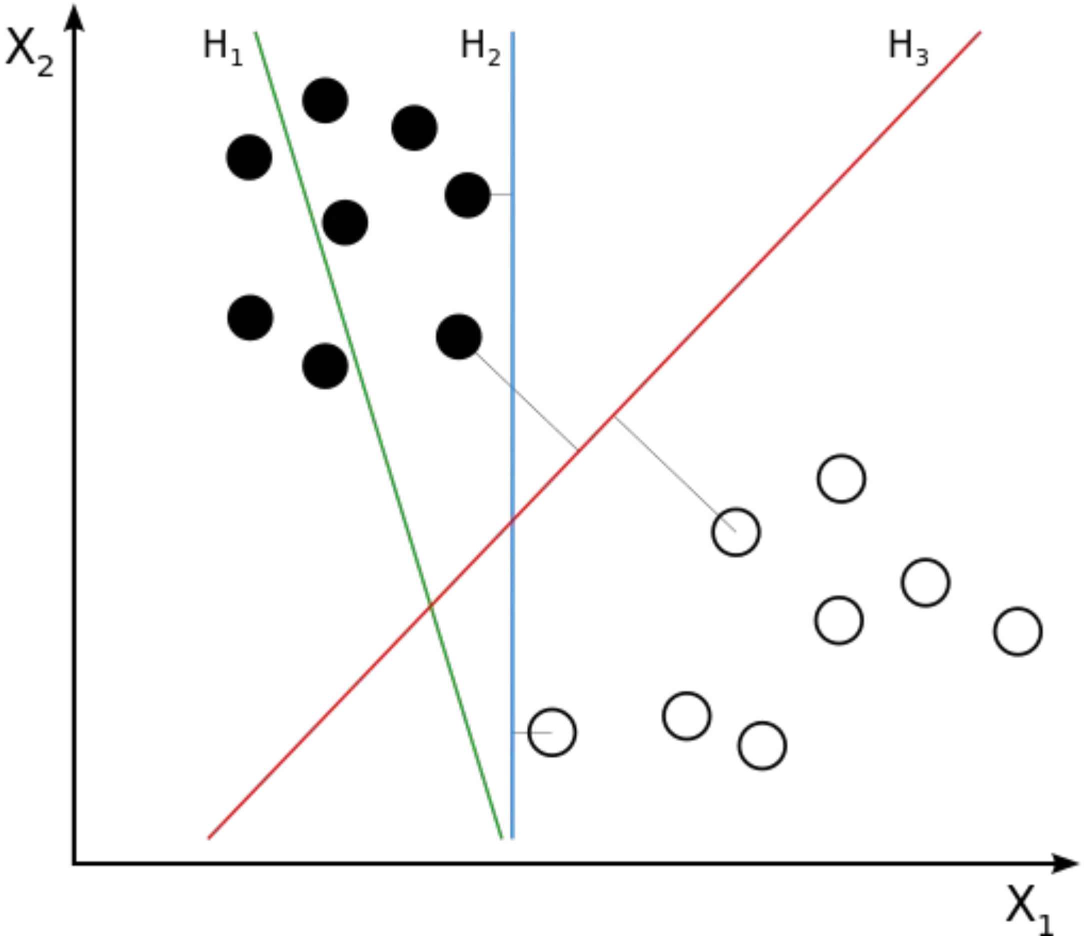


https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c

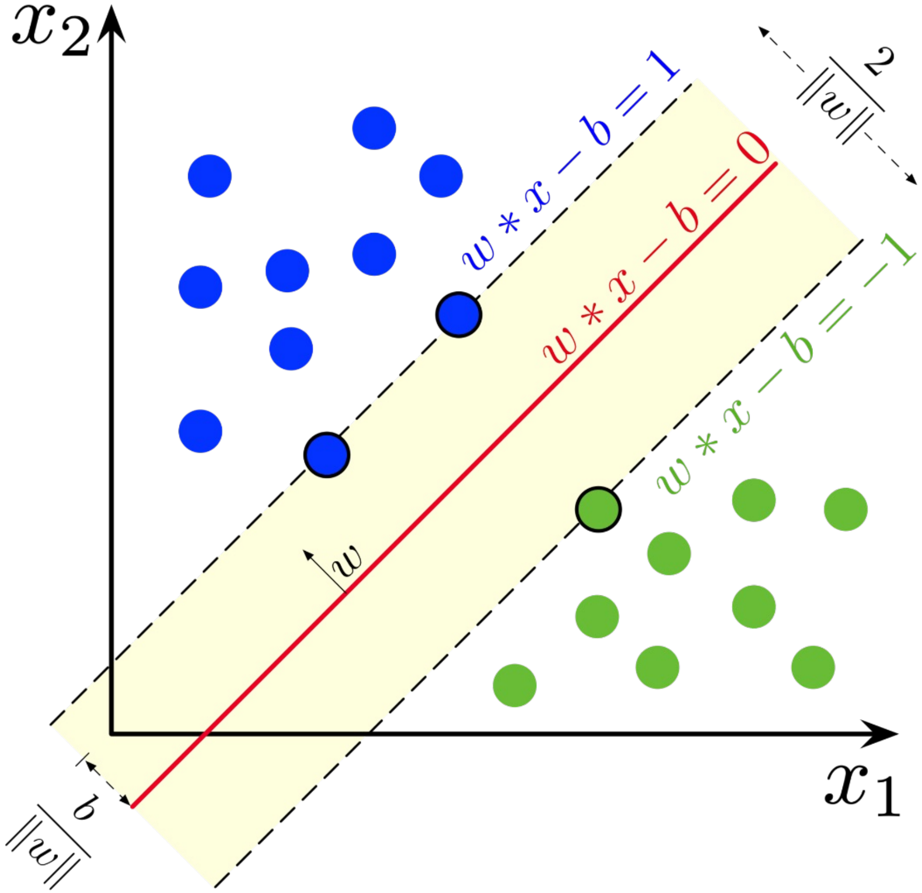
Introduction

- Support Vector Machines (SVMs) were created in the mid 1990’s by Vladimir Vapnik and his team at AT&T Bell Labs
- Named as “support-vector networks” that is binary classification algorithm and implements the idea of mapping non-linearly vectors to a very high-dimension feature space to construct a linear decision surface (hyperplane) in this feature space
- This hyperplane is optimal in the sense of being a maximal margin classier with respect to the training data
- SVM extends the two-dimensional linear separable problem to multidimensional, and aims to seed the optimal classification surface, also called as optimal hyperplane.

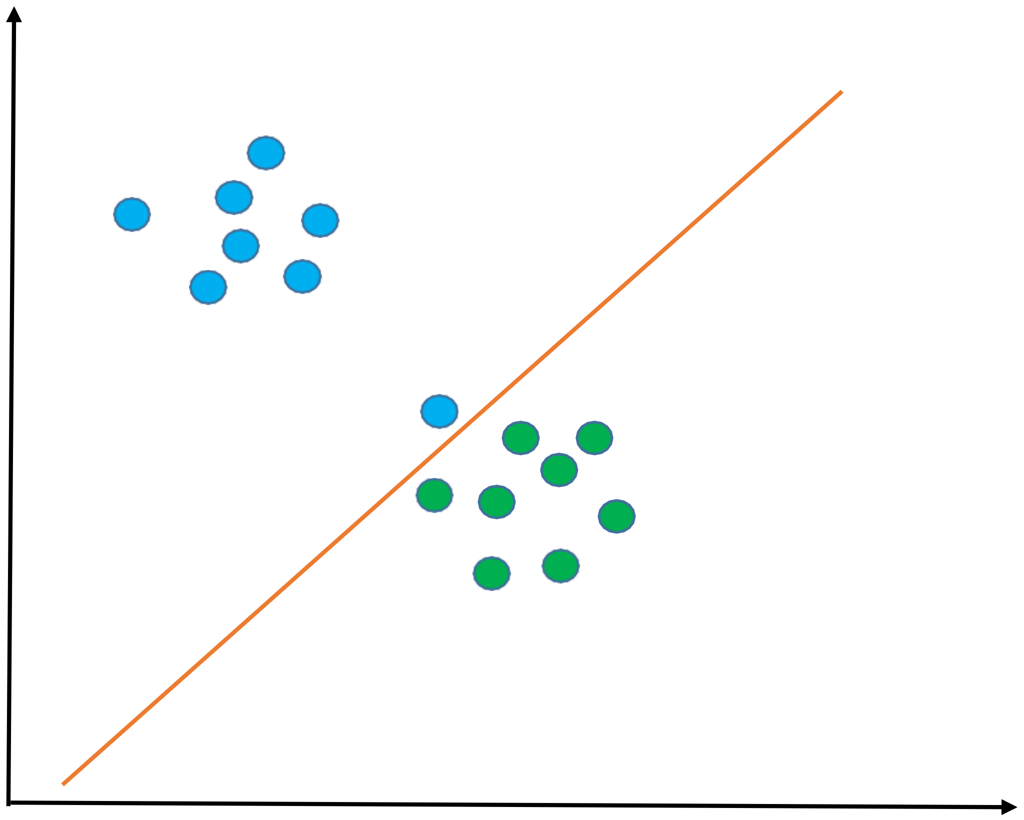
Maximum-Margin Hyperplane



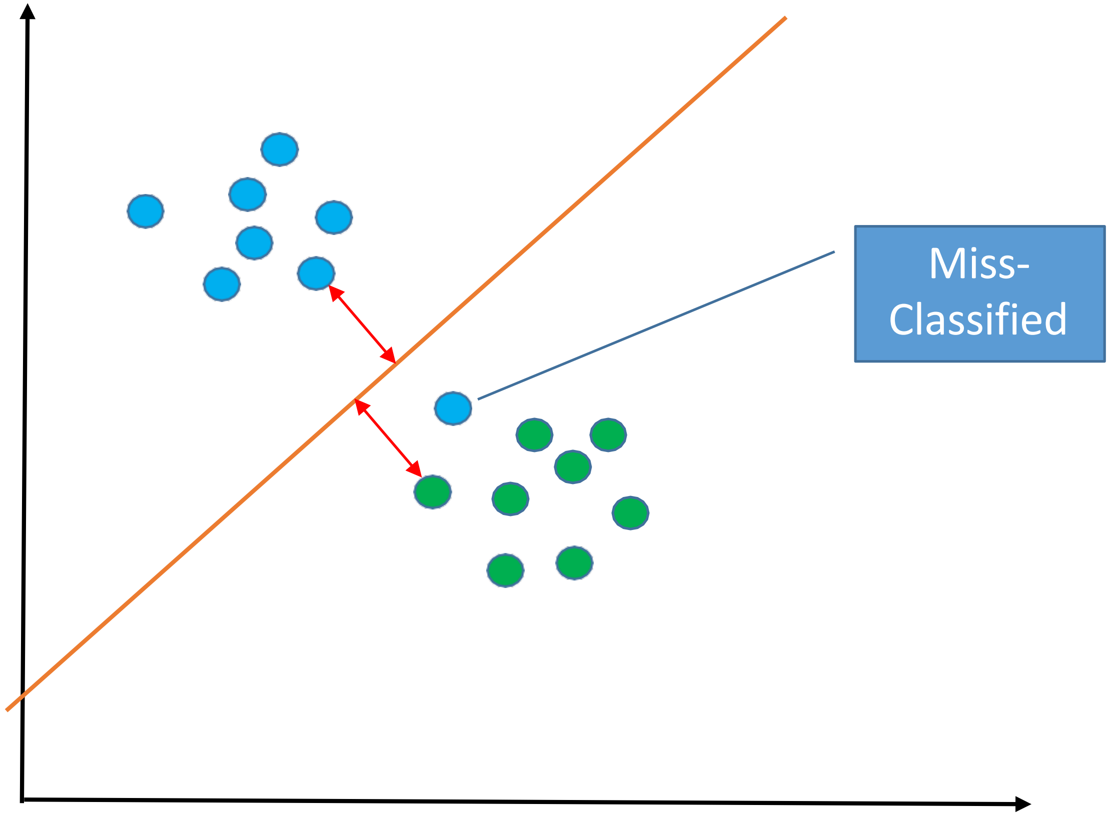
Maximum-Margin Hyperplane/Classifier



Too Sensitive to Outliers

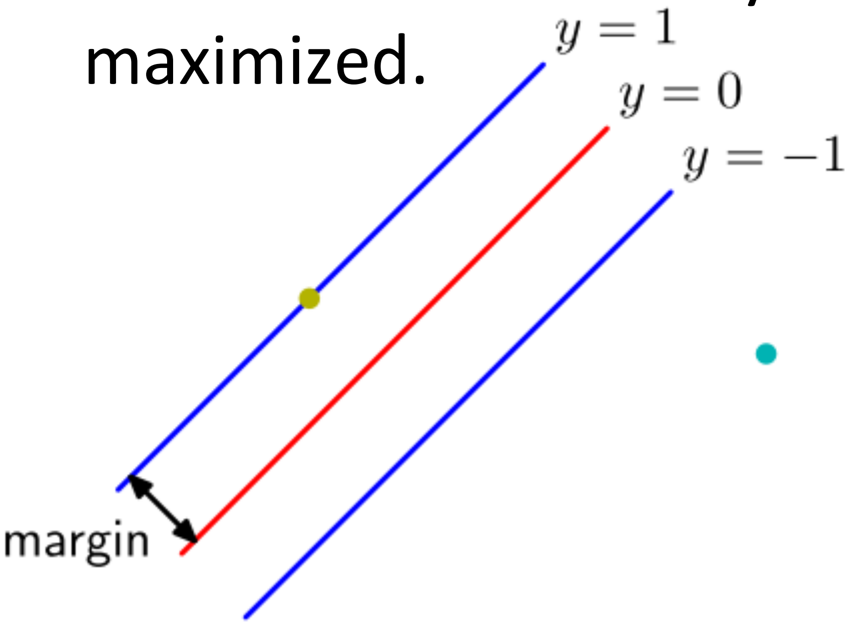


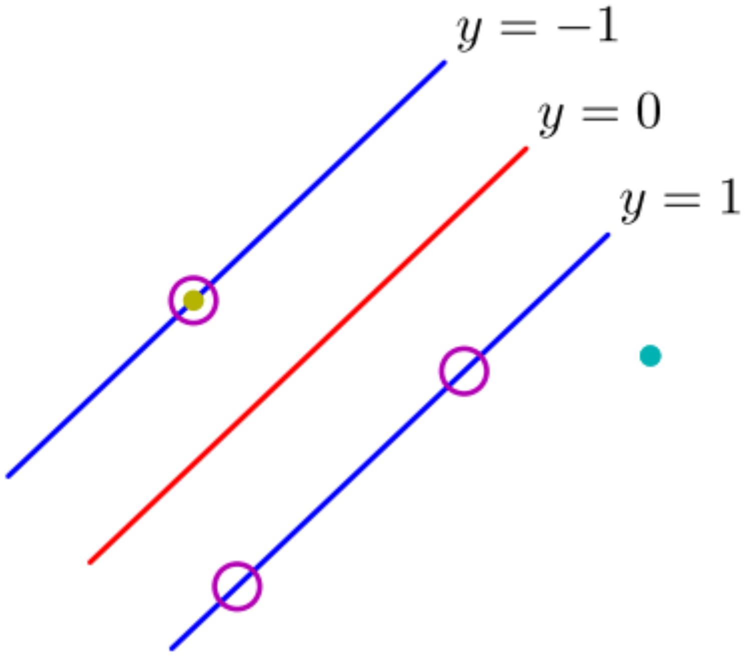
Allow Misclassification



Support Vector Machines (SVM)

Support Vector Machines (SVM) try to find the one that will give the smallest generalization error.
The decision boundary is chosen to be the one for which the margin is maximized.





Support Vector Machines (SVM)

SVM

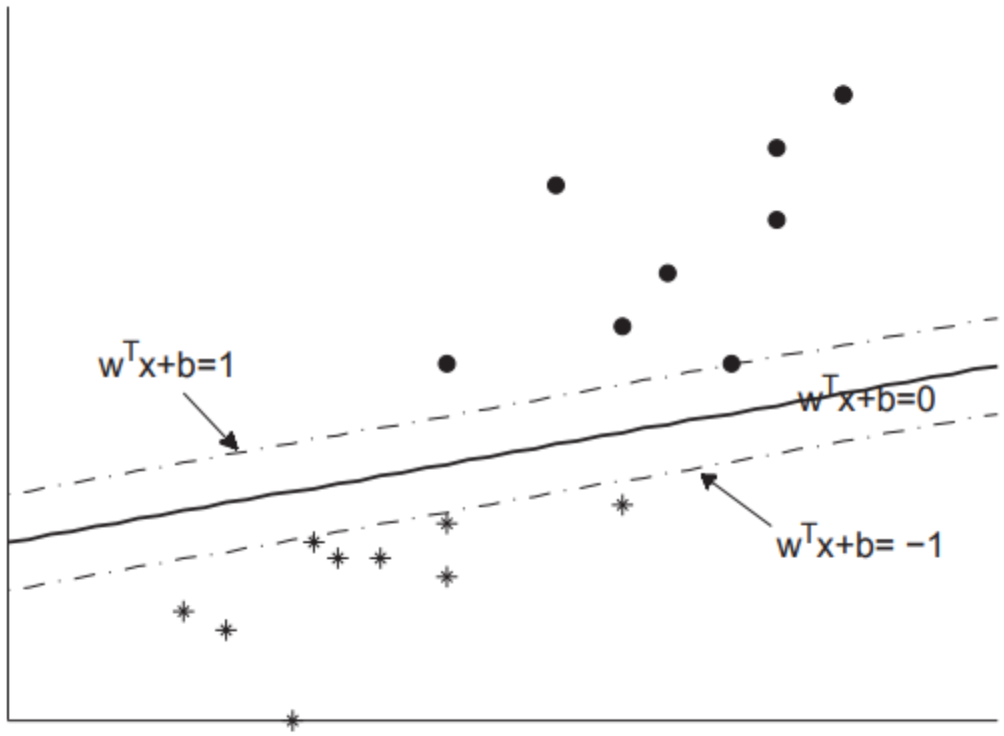


Figure 3: Optimal Classification Line.

Soft Margin Classifier/Support Vector Classifier

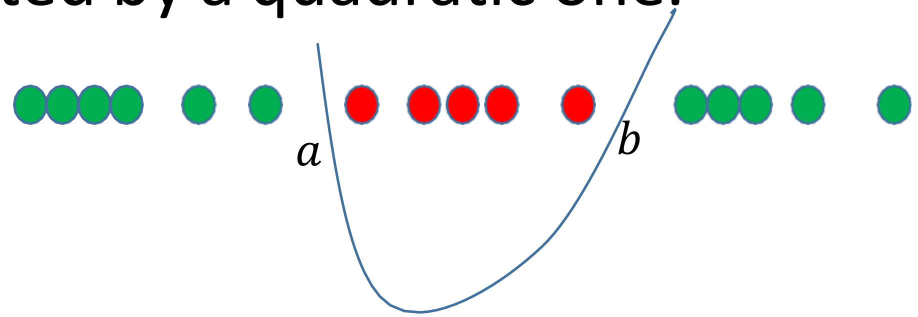
- If we allow misclassification, the distance between the “edge” is called a soft margin.
- How do find a line with the best soft margin?
- Use cross validation to determine how many misclassifications and observations to allow inside of the soft margin to get the best classification.
- Using soft margin to determine the line is called soft margin classifier, aka, support vector classifier.
- Data points on the edge and with the soft margin are called support vectors.

High Dimensional Feature Space

- For one-dimensional feature space, the support vector classifier is a point.
- For two-dimensional feature space, the support vector classifier is a line.
- For three-dimensional feature space, the support vector classifier is a plane.
- The support vector classifier is a hyperplane with n-1 dimensions for a n-dimensional feature space.
- The hyperplane is called flat affine subspace.

Non-Linear SVMs

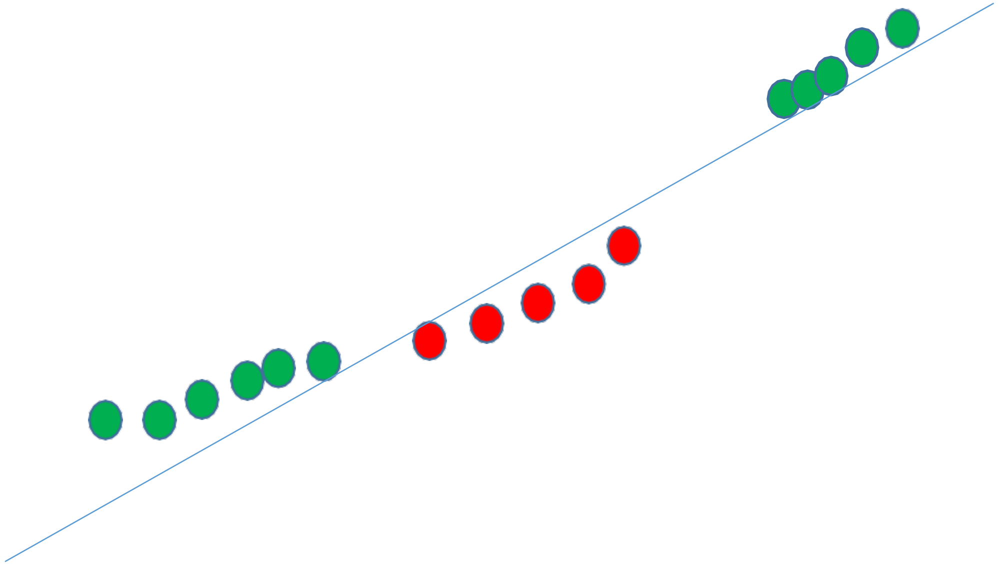
- For non-linearly separable data sets, we may map the data to a higher dimensional space.
- The following set cannot be separated by a linear function, but can be separated by a quadratic one.



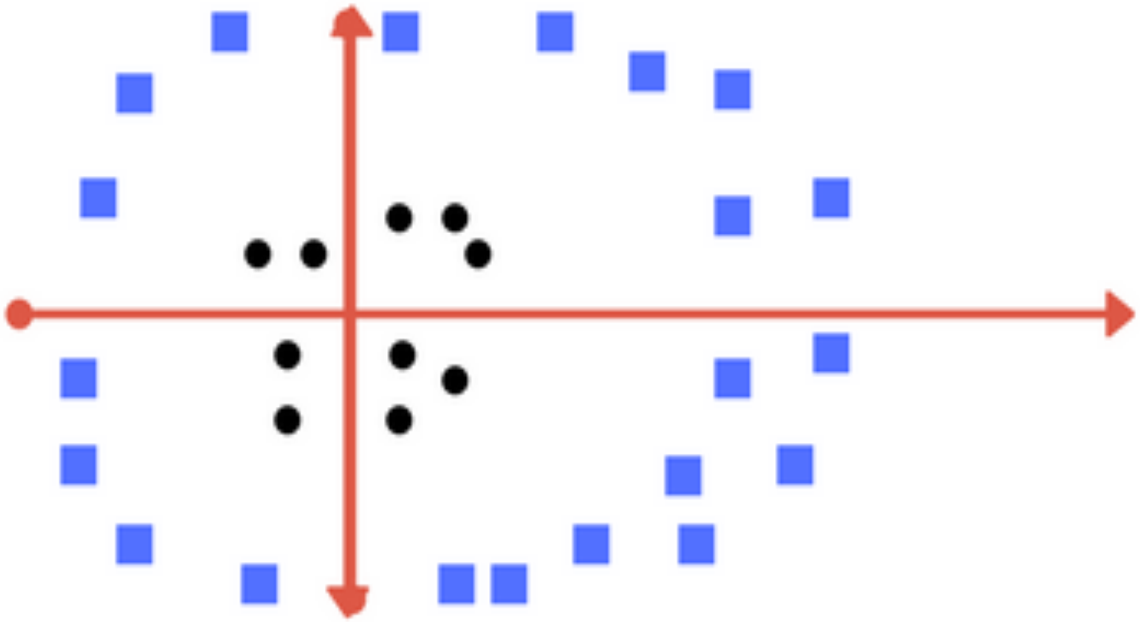
- $y = (x - a)(x - b) = x^2 - (a + b)x + ab$

Mapping to y-axis

- If we map $x \rightarrow x^2$, we can separate them by a line.



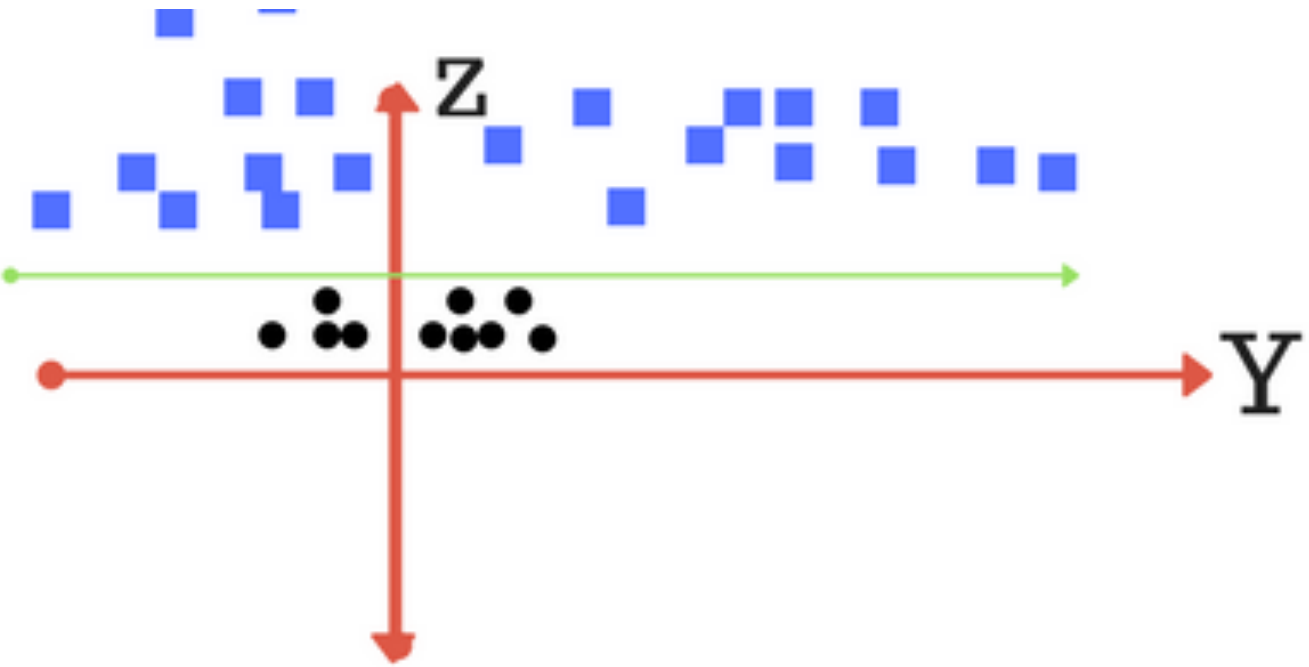
How about this?



Both Maximum Margin Classifier and Support Vector Classifier cannot do this!

Adding another Dimension

We add z-axis that represents the distance from the origin to a data point, computed by $\sqrt{x^2 + y^2}$.



Support Vector Machines

- Start with data is relatively low dimension
- Move the data to a higher dimension
- Find a support vector classifier

How to Determine Data Transformation?

- In the previous example, we compute $\sqrt{x^2 + y^2}$. Why not others?
- SVM uses kernel functions to systematically find support vector classifiers in higher dimensions.
- What are kernel functions?

Early Development in Support Vector Machines

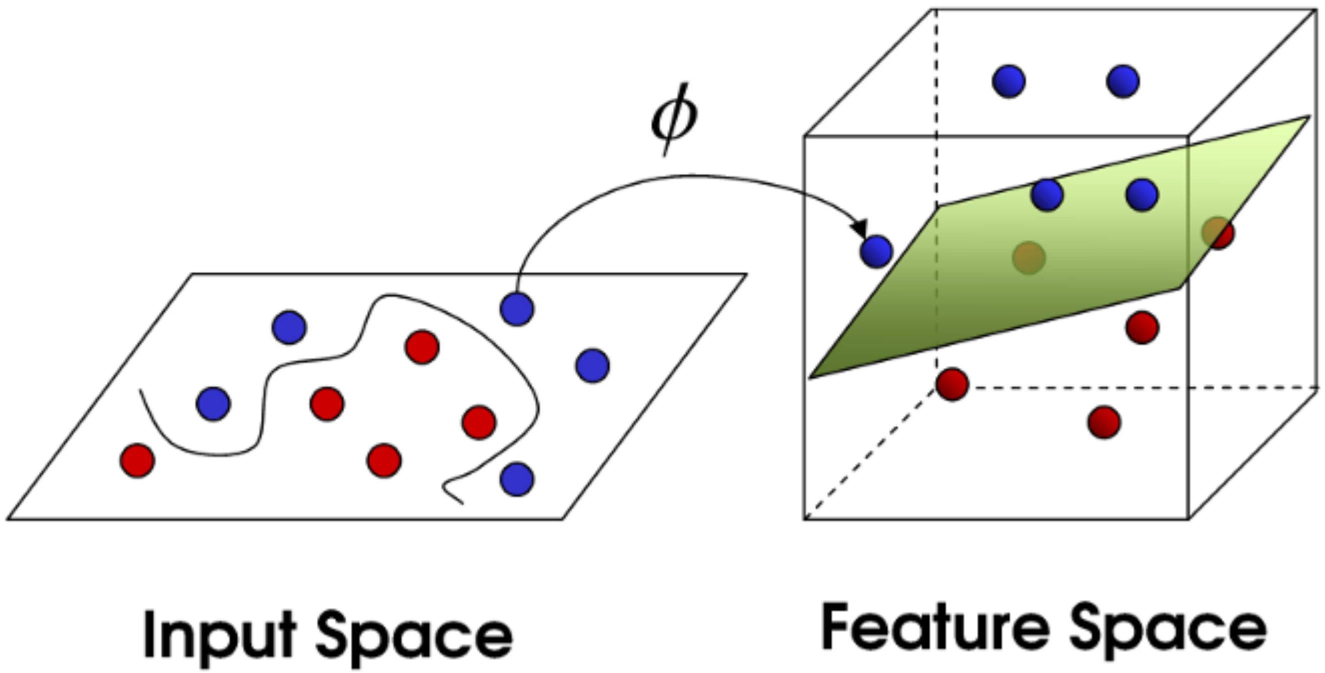
- Most basic form attempts to predict membership of a data point, into one of two classes (e.g., yes vs no, male vs female).
- SVMs are very robust and not prone to over fitting, so they are a good choice for less experienced aspiring Data Scientists

Cont.

- They attempt to identify a mathematical function that “maps” the data to a new “space” where the data for a given class/group are far apart from those of the other class/group.
- They employ a mathematical “similarity” metric using a mathematical operation called an “inner product” along with the mathematical mapping, to form a “kernel function” that provides the separation of classes/groups.
- Examples of mathematical mappings that are commonly used are: polynomials and exponential functions

Kernel Trick

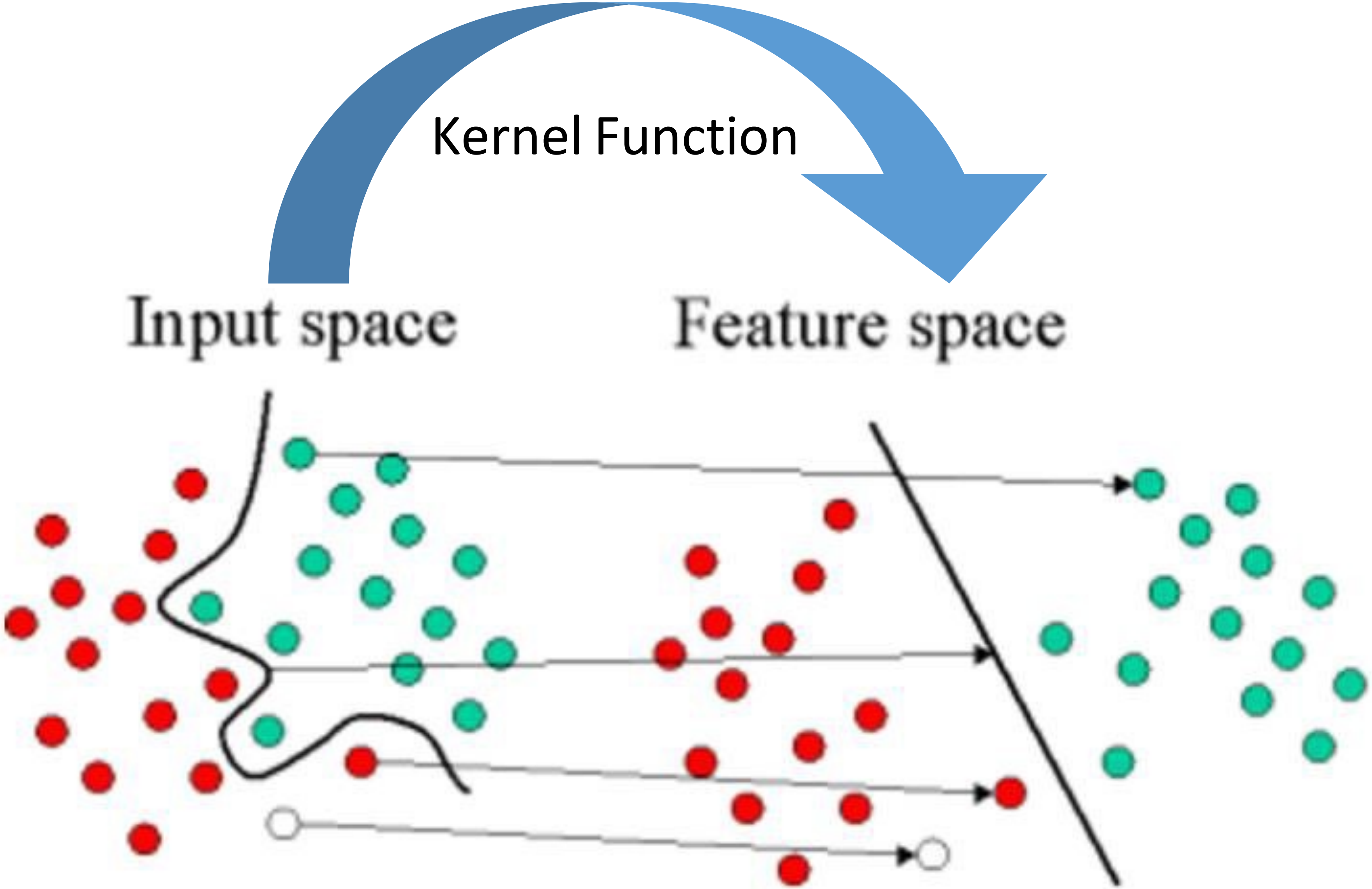
- Kernels:
  - Linear
  - Gaussian
  - Polynomial



https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c

Linear SVM

- Given a training dataset of n points (support vectors) of the form $(X_i,$ $y_i)$, where the y_i are either 1 or −1, each indicating the class to which the point X_i belongs.
- We want to find the "maximum-margin hyperplane" that divides the group of points X_i for which y_i=-1 from the group of points for which y_i=1, which is defined so that the distance between the hyperplane and the nearest point from either group is maximized.
- Any hyperplane can be described as the set of points X_i where w is the (not necessarily normalized) normal vector to the hyperplane. The parameter $wx - b = 0,$ b determines the offset of the hyperplane from the origin along the normal vector.



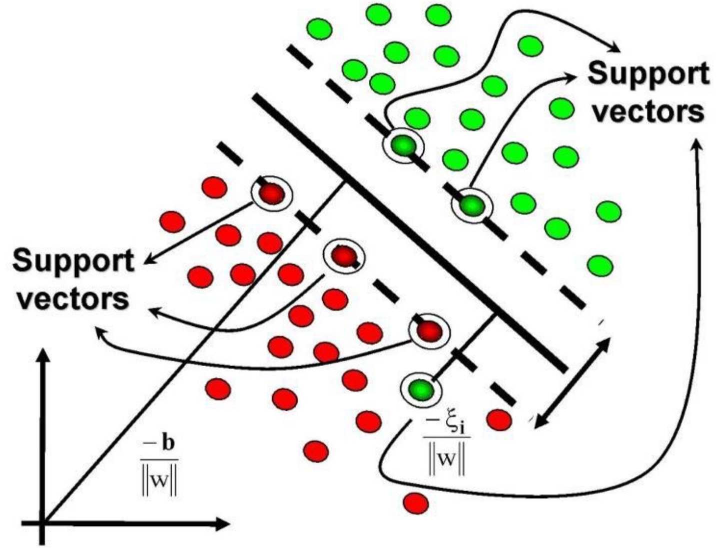
Kernel Function

Graphic Source: http://www.statsoft.com/Textbook/Support-Vector-Machines

Why SVM?

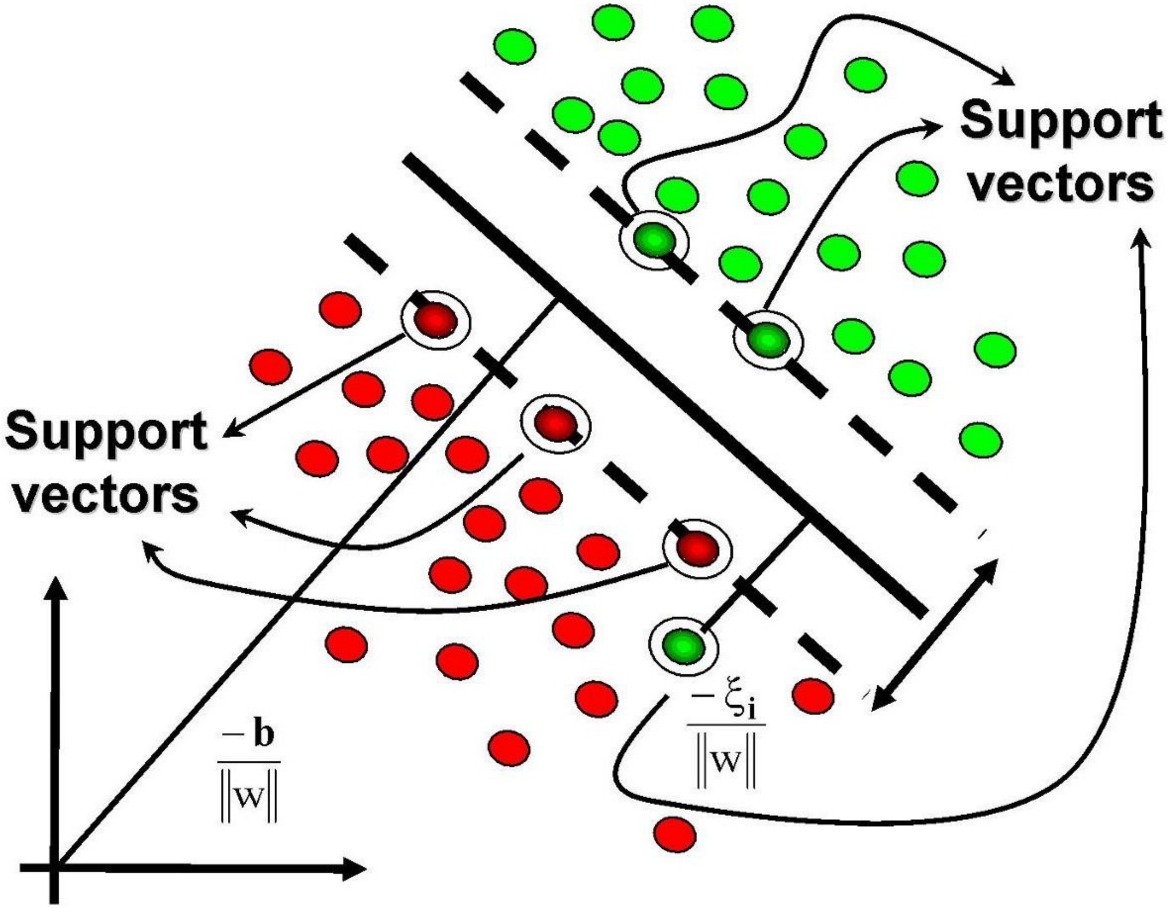
- “Machine” because it sounds cool (or some equally trivial reason, I’m sure).
- “Support Vectors” are essentially the points that define the separation of the classes.
- The points are called “vectors” because mathematicians think of points like these as representing vectors from the origin.

- In the image below, they reside on the dashed line or are points that aren’t on the same side of the dashed lines as the rest of the class.
- The dotted lines represent the “margin boundaries”.
- Algorithm attempts to map all data points to be on or outside the boundary (on the appropriate side)



- **wx** + b = 0 : equation of the solid line.
- **wx** + b = 1 : equation of upper dashed line
- **wx** + b = -1 : equation of lower dashed line

- The minimum ”size” of **w** that satisfies the
- equation is the best solution


Support vectors
Support vectors
$\frac{-\mathbf{b}}{\|w\|}$
$\frac{-\xi_{\mathbf{i}}}{\|w\|}$

Notes

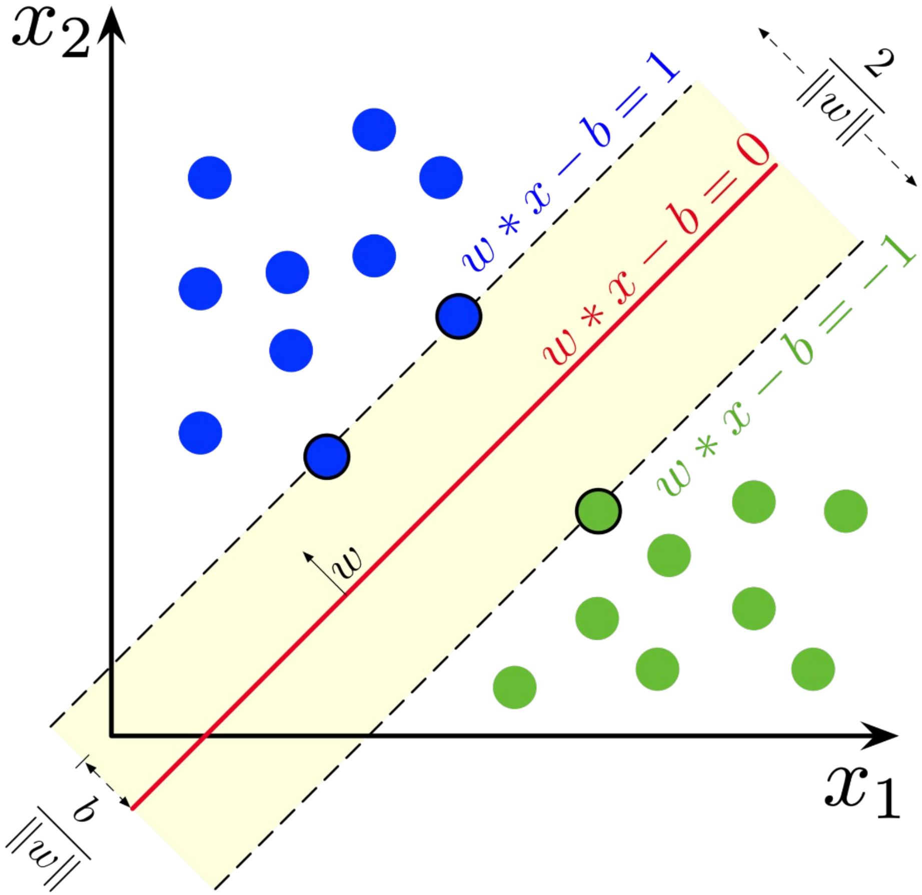
- SVM originates the concept of “maximum margin classifiers,” which require a fixed minimum separation of points.
- The concept is successfully applied to deep neural network algorithms.
- Maximum margin based models tend to “generalize” better than those that don’t use a fixed separation margin.

Maximizing the Margin

- We want a classifier with as big a margin as possible.
- The margin is $\frac{2}{||w||}$.
- To maximize the margin, we need to minimize $||w||$ with the condition that there are no data points in the margin.

$$w^T x_i - b \geq 1, \forall y_i = 1$$
$$w^T x_i - b \leq -1, \forall y_i = -1$$

- The condition can be merged to $y_i(w^T x_i - b) \geq 1$



Bias/Variance Tradeoff

- This plagues all of machine learning.
- If we pick a line that is very sensitive to the training data (low bias), it performs poorly when we predict new data (high variance).
- On the other hand, if we pick a line that is less sensitive to the training data and allow misclassifications (higher bias), it performs better when we test new data (low variance).

Parameter Optimization for SVM

- Goal is now to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary. We therefore minimize

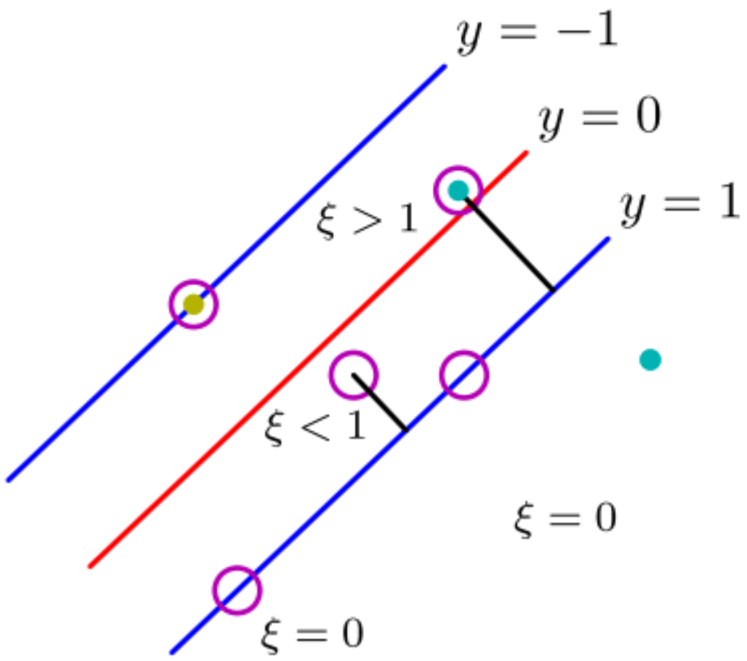
$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\mathbf{w}\|^2$$

where the parameter C> 0 controls the trade-off between the slack variable penalty and the margin

- C is a regularization coefficient because it controls the trade-off between minimizing training errors and controlling model complexity

Slack Variables

- To do this, we introduce slack variables, $\xi_n \geq 0$ where n =1,...,N, with one slack variable for each training data point,
- ξ_n =0 for data points that are on or inside the correct margin boundary
- Thus, a data point that is on the decision boundary $y(x_n)=0$ will have ξn =1,
- Points for which 0 <ξn ≤ 1 lie inside the margin, but on the correct side of the decision boundary,
- Those data points for which ξn > 1 lie on the wrong side of the decision boundary and are misclassified



Hyperparameters for SVM (C and Gamma)

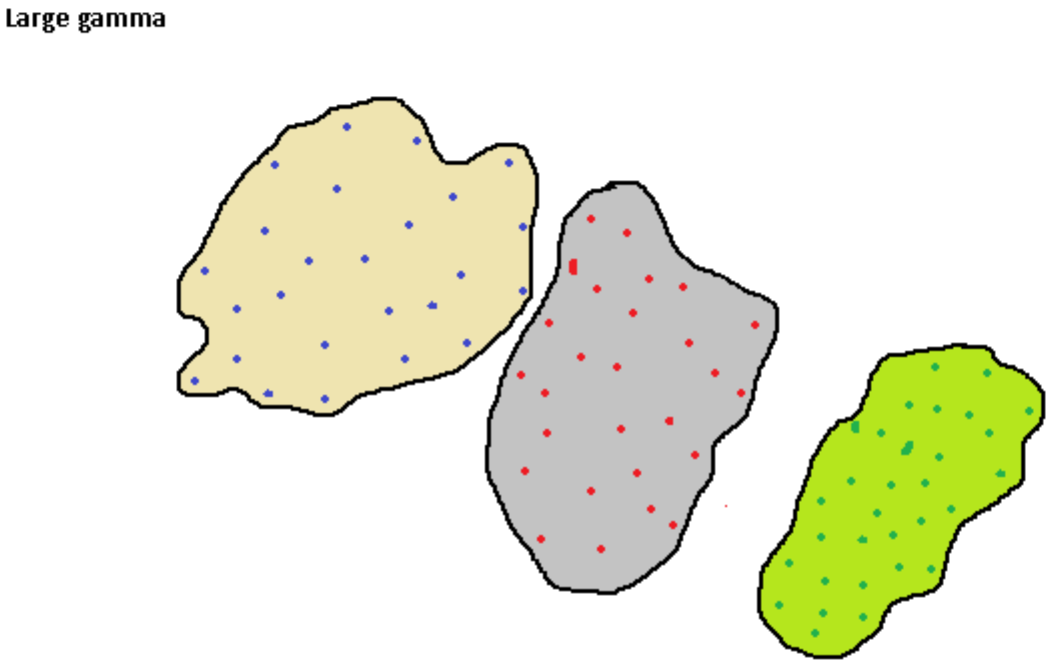
- Most of the machine learning and deep learning algorithms have some parameters that can be adjusted which are called hyperparameters.
- **C parameter** adds a penalty for each misclassified data point. If c is small, the penalty for misclassified points is low so a decision boundary with a large margin is chosen at the expense of a greater number of misclassifications [High Bias]
- If c is large, SVM tries to minimize the number of misclassified examples due to high penalty which results in a decision boundary with a smaller margin. Penalty is not same for all misclassified examples. It is directly proportional to the distance to decision boundary.[Low Bias]

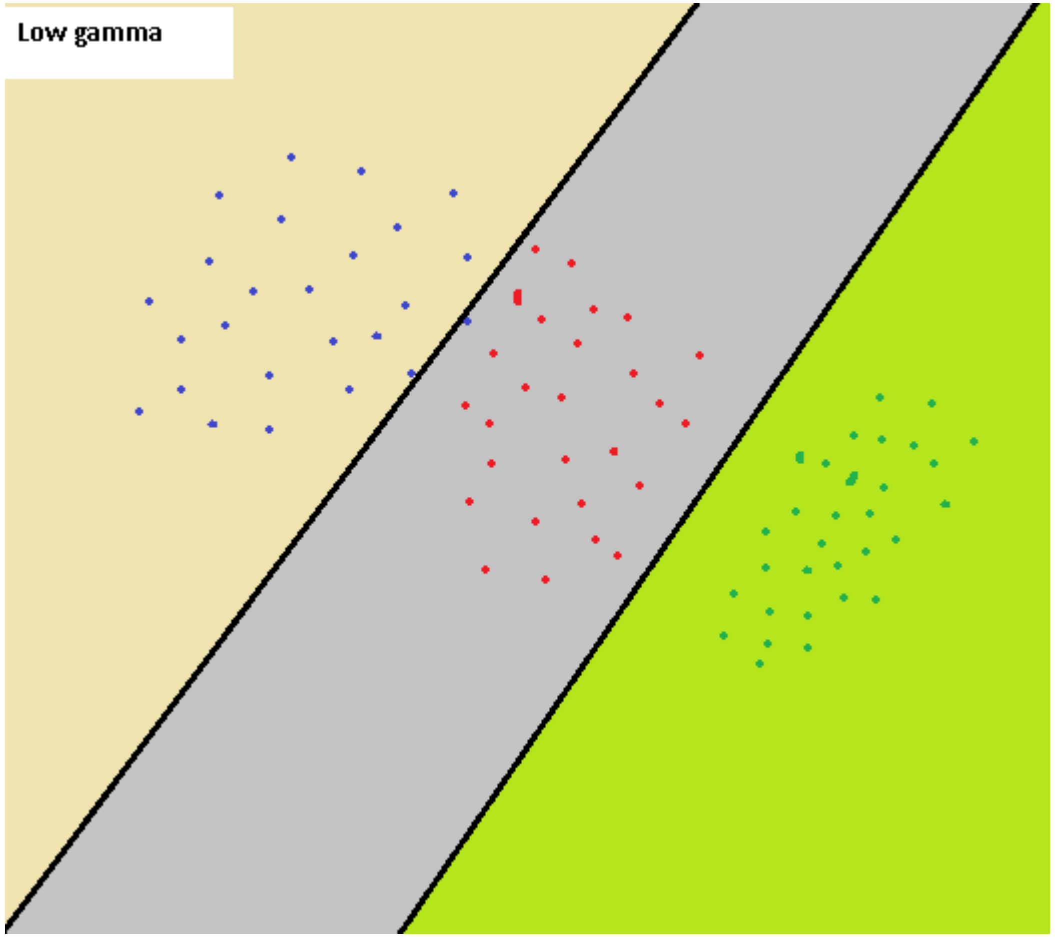
https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167

Gamma

- Gamma parameter of RBF controls the distance of influence of a single training point.
- Low values of gamma indicates a large similarity radius which results in more points being grouped together.
- For high values of gamma, the points need to be very close to each other in order to be considered in the same group (or class).
- Therefore, models with very large gamma values tend to overfit. [High Variance]

Gamma





https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a50974

Gamma vs C parameter

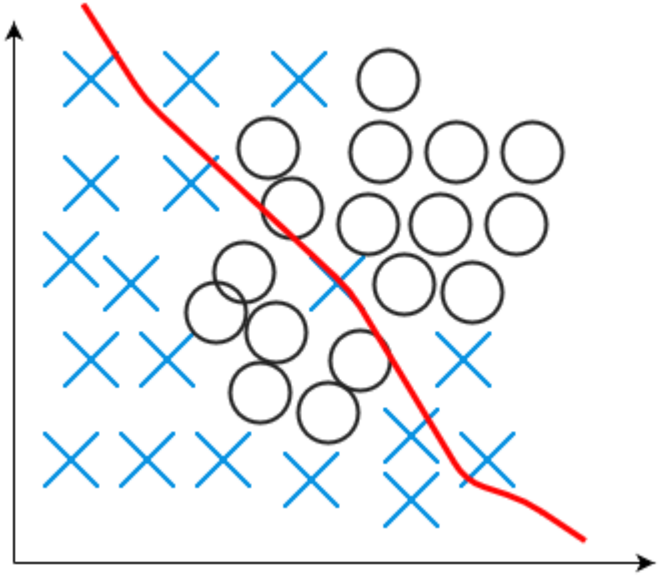
- For a linear kernel, we just need to optimize the c parameter.
- However, if we want to use an RBF/Gaussian kernel, both c and gamma parameter need to optimized simultaneously.
- If gamma is large, the effect of c becomes negligible.
- If gamma is small, c affects the model just like how it affects a linear model.

Standard Values

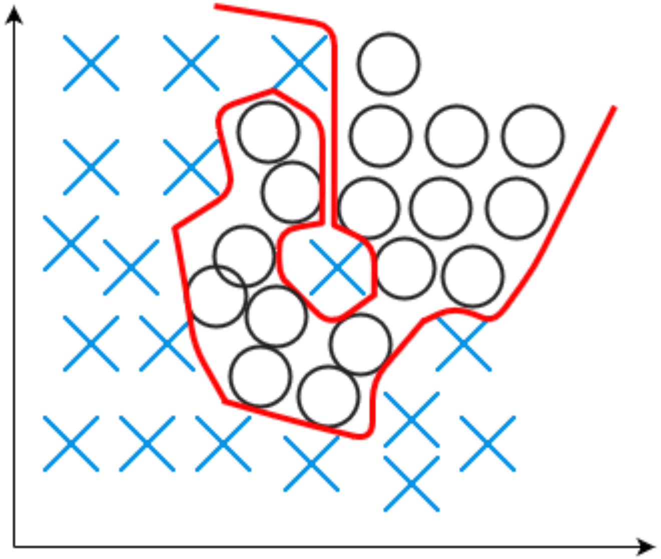
- Typical values for c and gamma are as follows:
  - 0.0001 < gamma < 10
  - 0.1 < c < 100
- However, specific optimal values may exist depending on the application
- Bias vs Variance Tradeoff

Bias vs Variance Tradeoff

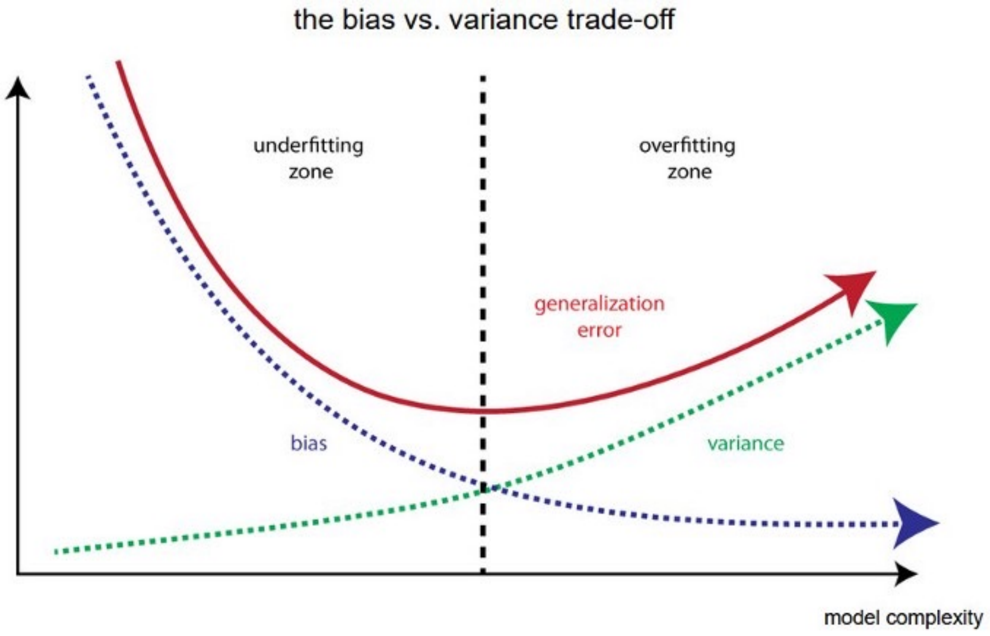
Build accurate model and avoid the mistake of overfitting and underfitting.



Underfitting/Low Variance



Overfitting/High Variance



https://towardsdatascience.com/bias-and-variance-but-what-are-they-really-ac539817e171