Support Vector Machines

Yulia Newton, Ph.D.

CS156, Introduction to Artificial Intelligence

San Jose State University

Spring 2021

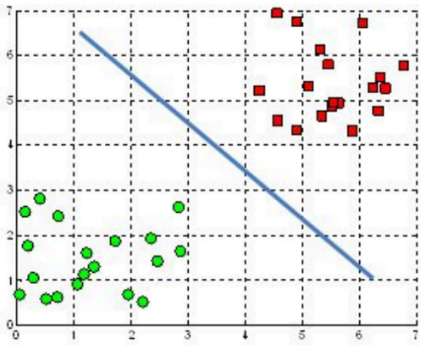
What is SVM?

- Support Vector Machines (SVM) are used for supervised ML problems
  - Can be used for both classification and regression problems but most often used for classification
    - The simplest SVM model is a linear binary classifier
  - Can do multi-class classification by creating multiple SVM models
    - Wrapper libraries are available
    - One vs. all classification
      - E.g. breast cancer vs. not, colon cancer vs. not, brain cancer vs. not
    - One vs. one classification
      - E.g. breast cancer vs. colon cancer, breast cancer vs. brain cancer, colon cancer vs. brain cancer

Decision hyperplane

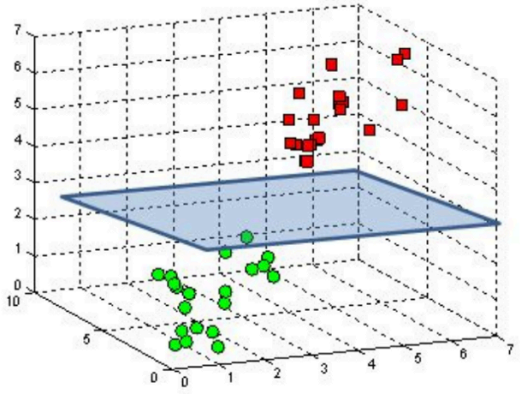
- Decision hyperplane separates classes
  - Decision hyperplane in 2-D is just a line
  - Decision hyperplane in 3-D is a plane
  - In > 3-D is called a hyperplane

A hyperplane in $\mathbb{R}^2$ is a line



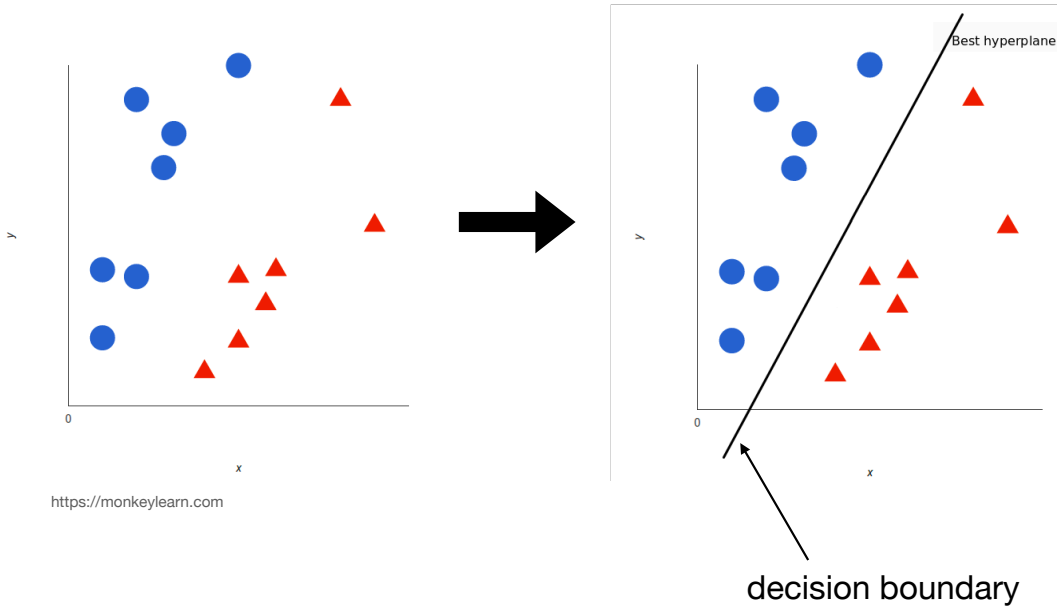
https://towardsdatascience.com/

A hyperplane in $\mathbb{R}^3$ is a plane



How does SVM model work?

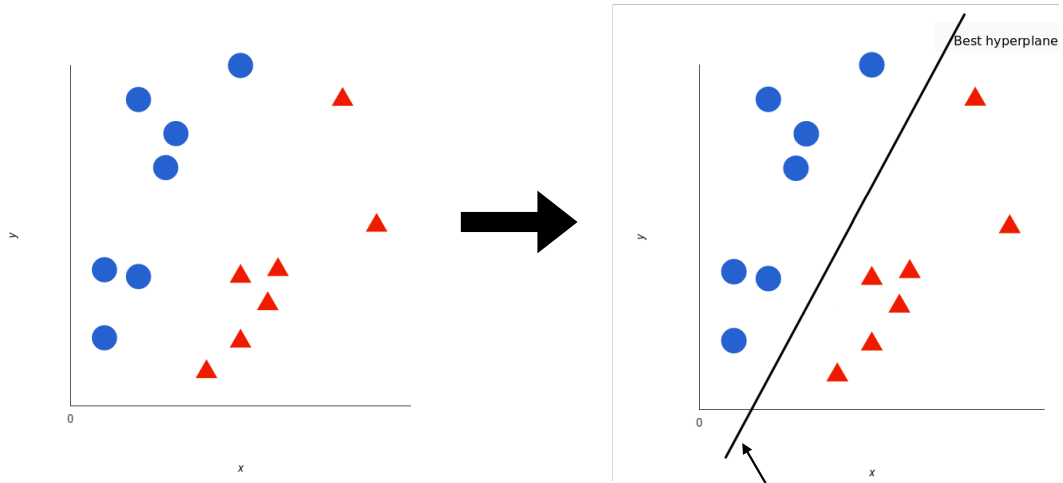
Linear SVM is the simplest SVM model



https://monkeylearn.com

How does SVM model work?

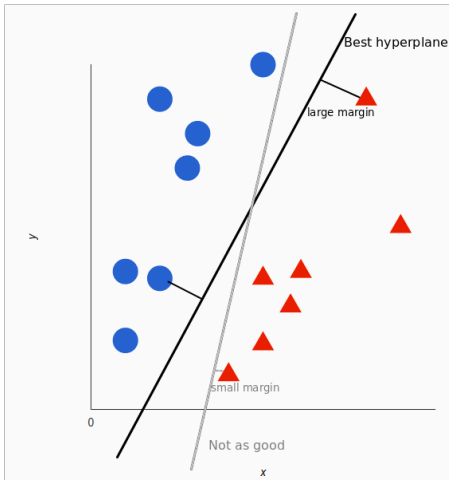
Linear SVM is the simplest SVM model



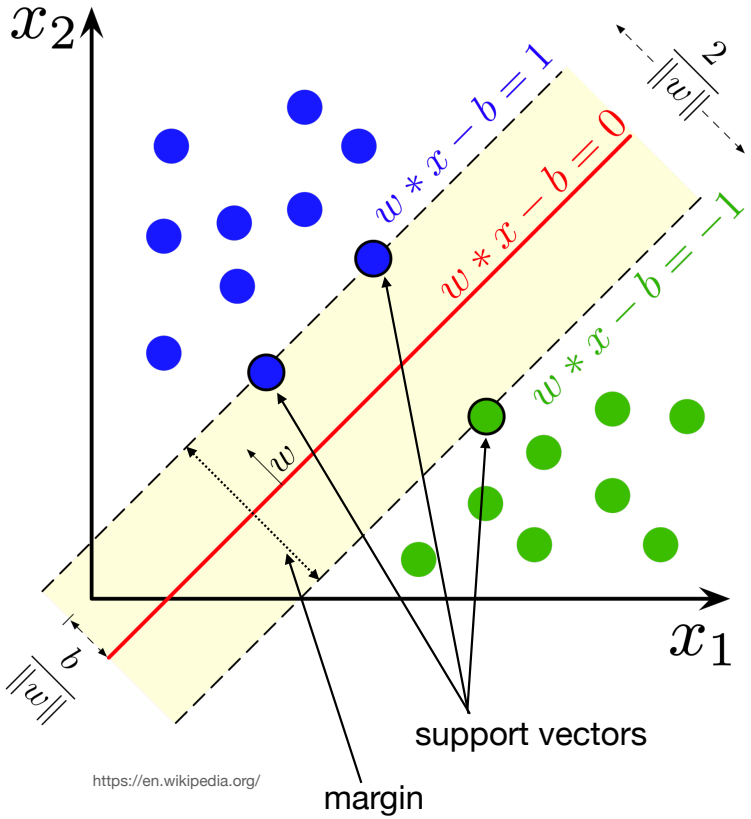
https://monkeylearn.com

decision boundary

Not all hyperplanes are equally good:

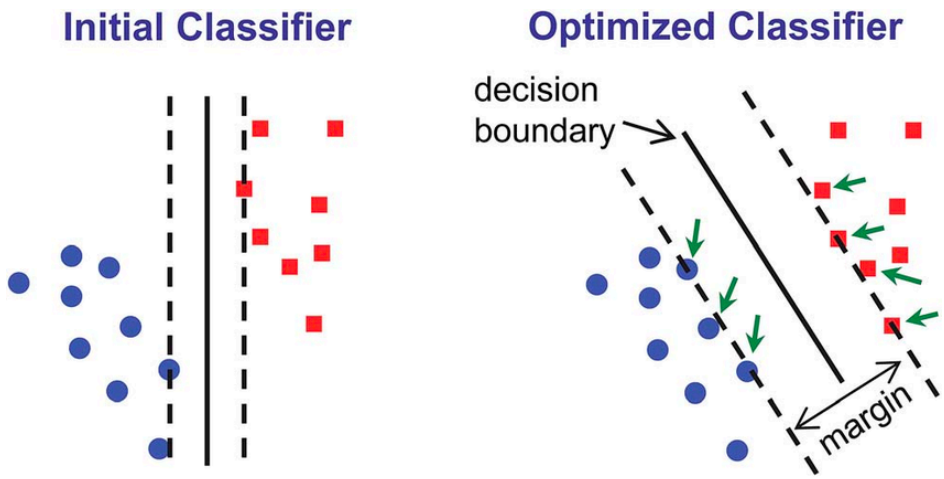


Maximum margin hyperplane



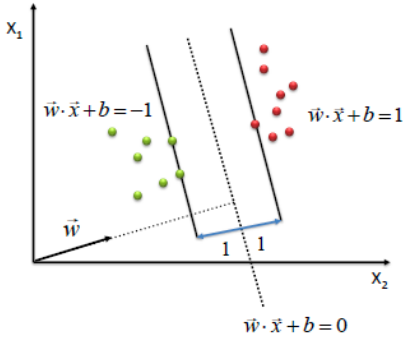
- SVM classifier is a non-probabilistic classifier
  - Each samples is assigned to one class or another based on the estimated weights (no probabilities involved)
    - Platt scaling allows transforming outputs of an SVM model to a probability distribution
  - A hyperplane with the biggest margin from the plane to support vectors is fitted

SVM optimization



Askim et al., Chemical Science 2015

- The optimal hyperplane is found by maximizing the margin between the support vectors of two classes



$$\max \frac{2}{\|w\|}$$

$s.t.$
$(w\cdot x+b)\geq 1, \forall x \text{ of class 1}$
$(w\cdot x+b)\leq -1, \forall x \text{ of class 2}$

www.saedsayad.com

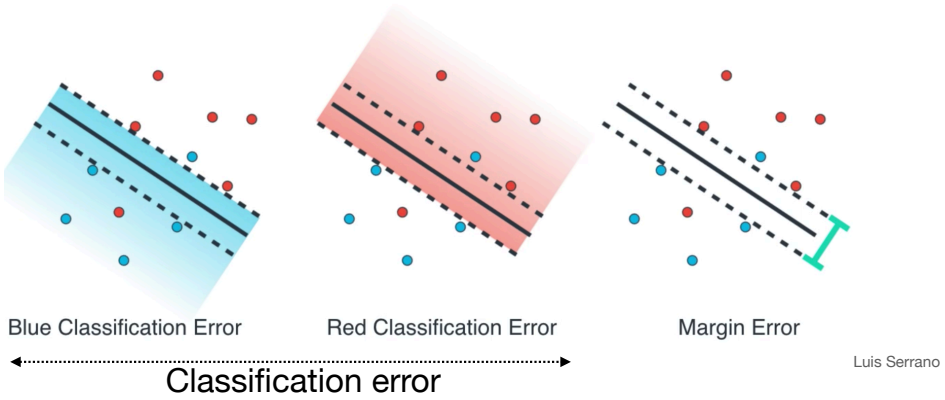
SVM optimization (cont'd)

- Start with a random hyperplane (random weights)
- For a pre-determined number of repetitions
  - Pick a random data point and classify
  - If the classification is correct, do nothing
  - If the classification is incorrect, move hyperplane towards the point
  - Increase the margin as we go

SVM error components

- Classification error - how many data points are misclassified
  - Misclassifications are based on the margin lines, not on the decision line
- Margin error - how far part the margin lines are
  - Want to keep the margin as wide as possible

SVM error is made up of two components:



Minimizing SVM loss

- SVM uses hinge loss

$$Cost(h_\theta(x), y) = \begin{cases} max(0, 1 - \theta^T x) & \text{if } y = 1 \\ max(0, 1 + \theta^T x) & \text{if } y = 0 \end{cases}$$

  - Hinge loss is a convex function often used as a cost function in classifier optimization
- The SVM error is minimized using gradient descent optimization
- Allows for soft margin - allow SVM a small number of misclassifications in order to keep the margin as wide as possible

$$J(\theta) = \sum_{i=1}^{m} y^{(i)} max(0, 1 - \theta^T x) + (1 - y^{(i)}) max(0, 1 + \theta^T x)$$

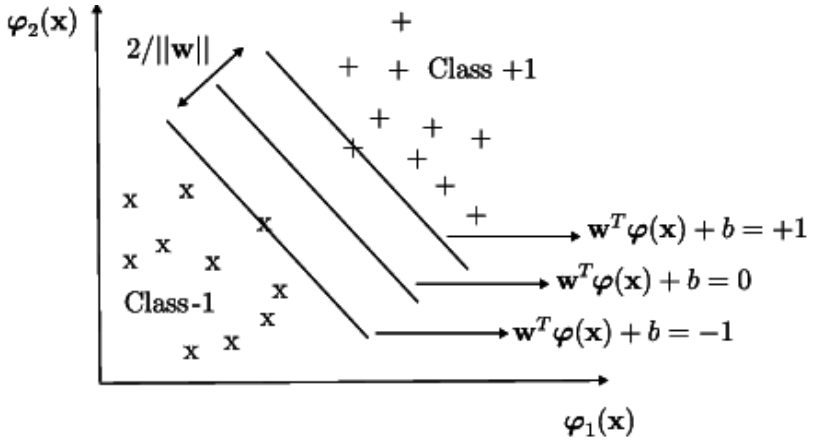
$$\downarrow$$

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})]$$

https://towardsdatascience.com

How class is predicted once we optimize the model weights (model parameters)

- After we estimate the weights and the intercept we can apply them to the new samples in the same way as we would for other prediction tasks
  - If the predicted value is positive then the sample is predicted as class 1, otherwise class 0



Askim et al., Chemical Science 2015

Apply weights and intercept:

$\mathbf{w}^T\mathbf{x} + b \geq 0$

$\mathbf{w}^T\mathbf{x} + b < 0$

Regularized SVMs

- We can add a regularization term to the loss function and make our model a regularized model

L2 regression (ridge)

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

https://towardsdatascience.com

Kernelized SVMs

- We can add a kernel function to an SVM model
  - Often just called a kernel
  - Transforms data into a new space prior to learning a decision hyperplane

Cost functions:

Linear SVM: $$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})]$$

Linear SVM with regularization: $$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

Non-linear SVM: $$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(f^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(f^{(i)})]$$

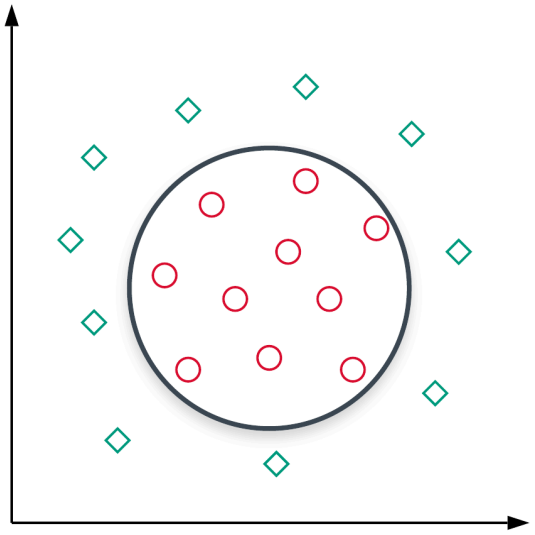
Non-linear SVM with regularization: $$J(\theta) = C[\sum_{i=1}^{m} [y^{(i)} Cost_1(\theta^T(f^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(f^{(i)})] + \frac{1}{2}\sum_{j=1}^{m} \theta_j^2$$

kernel function

https://towardsdatascience.com

Why use a kernel?

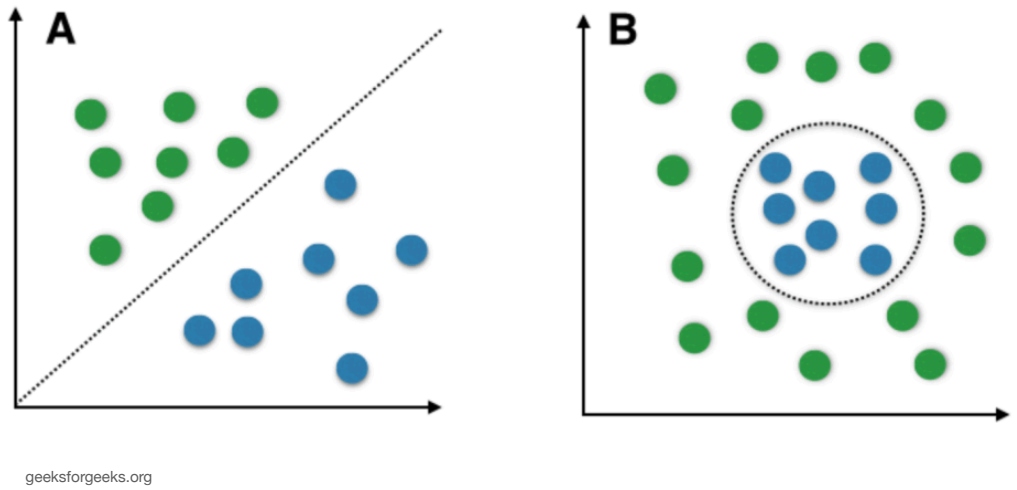
- Sometimes data is not separable by a linear line/hyperplane but it is separable if transformed by a non-linear transformation



https://towardsdatascience.com

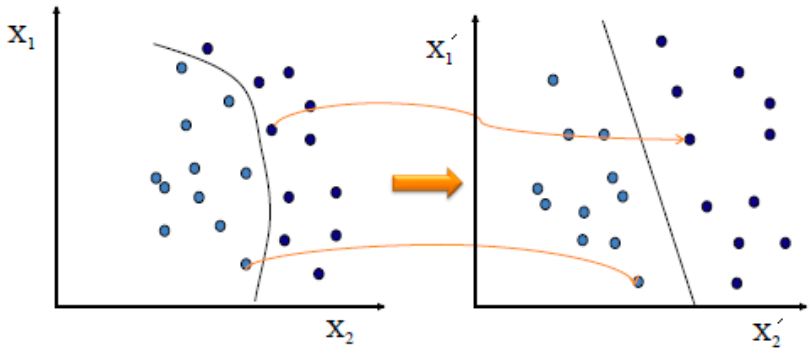
Hmmm … what do we do now?

Separable vs. not

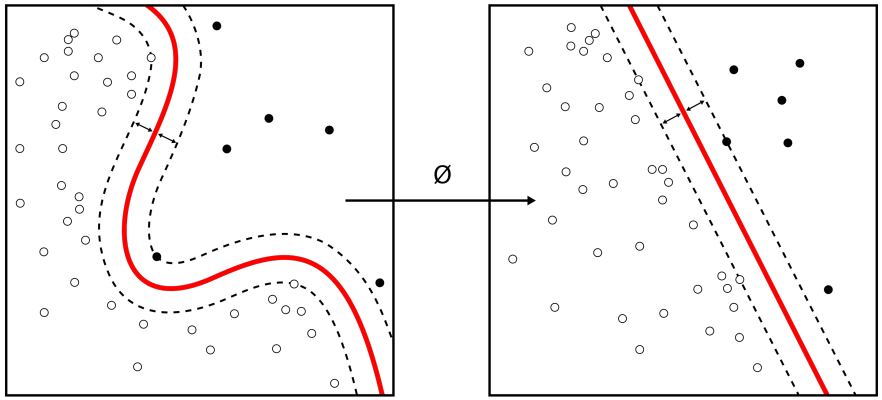


geeksforgeeks.org

Kernel trick allows transforming data into a new space, in which it's separable

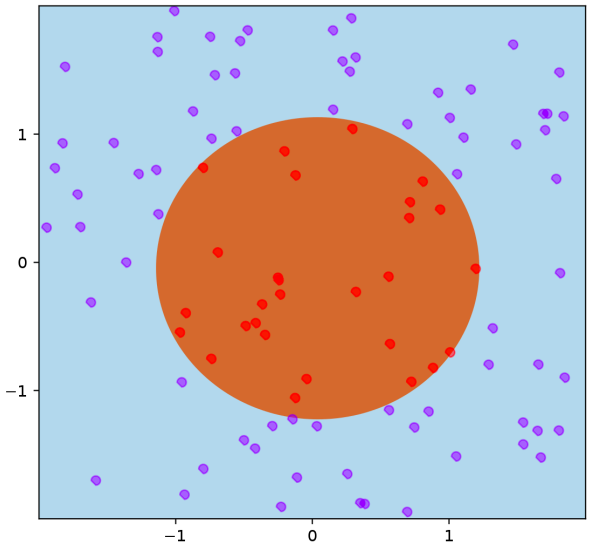


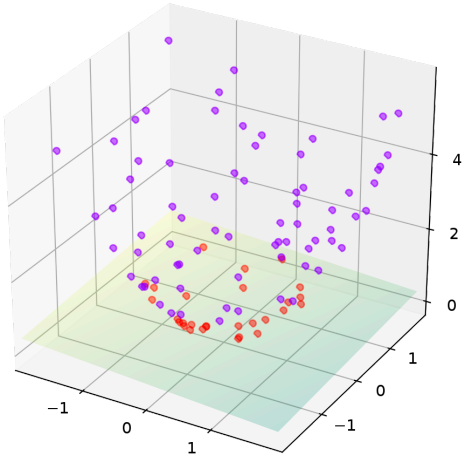
https://towardsdatascience.com



en.wikipedia.org

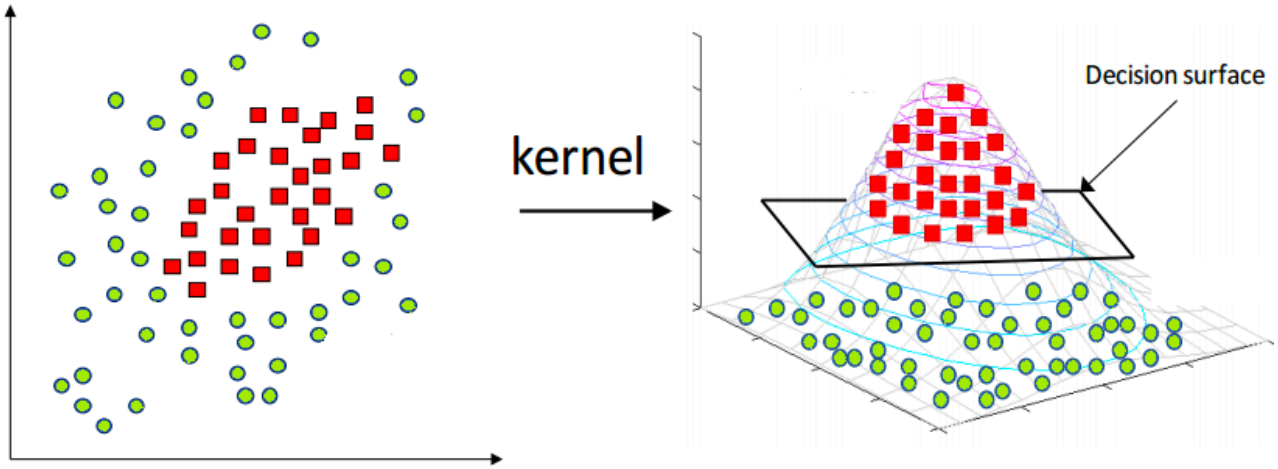
Example: quadratic polynomial kernel





en.wikipedia.org

Example 2: quadratic polynomial kernel



kdnuggets.com

Formulation of linear SVM with a kernel function

- Linear SVM can be formulated as a kernelized SVM model where the kernel function is an identity function

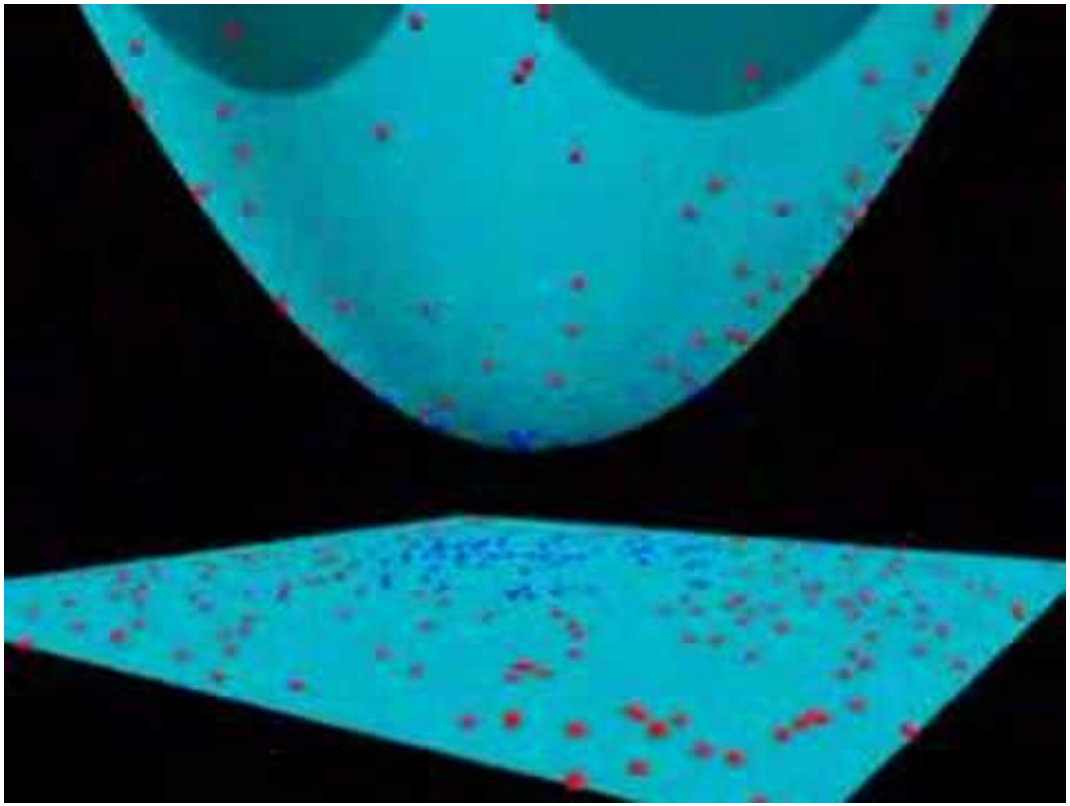
$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(x^{(i)})]$$

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(f^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(f^{(i)})]$$

This cost function becomes the cost function on the left if $f(x) = I(x)$

Visualizing kernel trick

- https://youtu.be/3liCbRZPrZA



Types of kernel functions most commonly used

- Linear
- Polynomial
- Radial basis function (RBF)
- Sigmoid
- Tanh

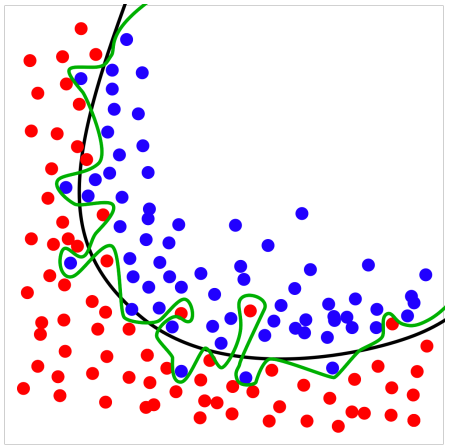
Type of Support Vector Machine	Inner Product Kernel $K(x,x_i)$, $I = 1, 2, \ldots, N$	Usual inner product
Polynomial learning machine	$(x^T x_i + 1)^p$	Power *p* is specified *a priori* by the user
Radial-basis function (RBF)	$\exp(1/(2\sigma^2)\|\|x-x_i\|\|^2)$	The width σ^2 is specified *a priori*
Two layer neural net	$\tanh(\beta_0 x^T x_i + \beta_1)$	Actually works only for some values of β_0 and β_1

Pros vs. cons of using a kernel function

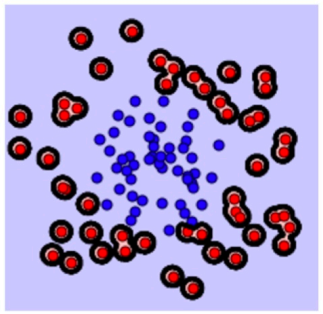
- Pros
  - Increased classifier performance on most types of data
  - Versatility - many kernel function options
    - Custom kernel functions are an option
  - Kernelized SVMs do not depend on dimensionality of the data (high vs. low dimensions)

- Cons
  - Resource usage and efficiency decreases as the training set size increases
  - Depended on data normalization/scaling
  - Requires additional hyper-parameter tuning
  - No direct probability estimation (e.g. Platt scaling cannot be applied)
  - Predictions are less interpretable than in the case of linear SVM (why was the prediction made?)
  - Easy to overfit the model

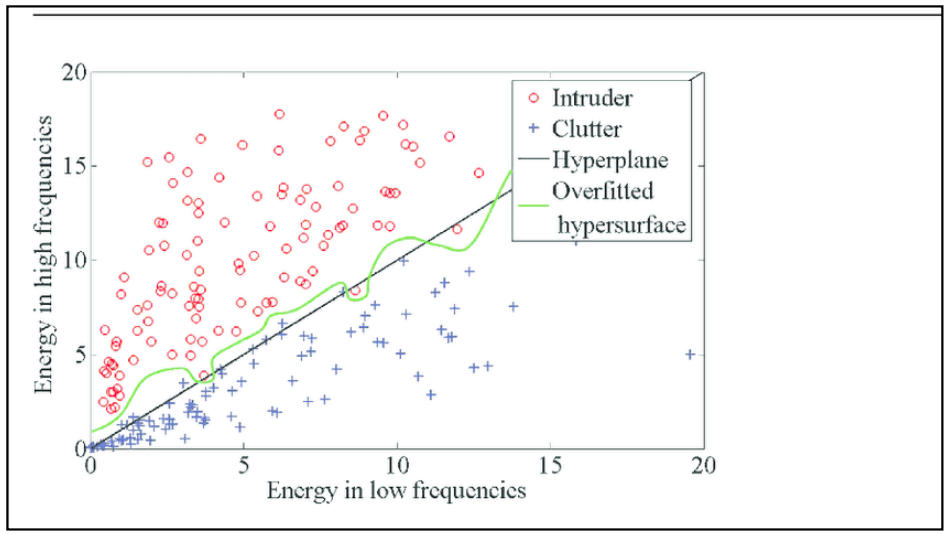
Be careful of overfitting your SVM model



en.wikipedia.org



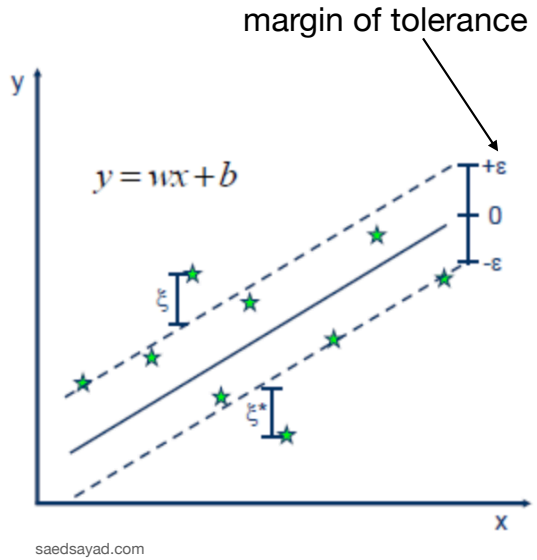
https://github.com/mainkoon81/Study-09-MachineLearning-A



"Wireless Sensor Networks for Human Intruder Detection", Journal of the Indian Institute of Science 2010

How to use SVM for regression problems?

- Support Vector Regression (SVR)
  - Uses the same principles as the SVM
    - Minimize error while maximizing the margin



- Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

- Constraints:

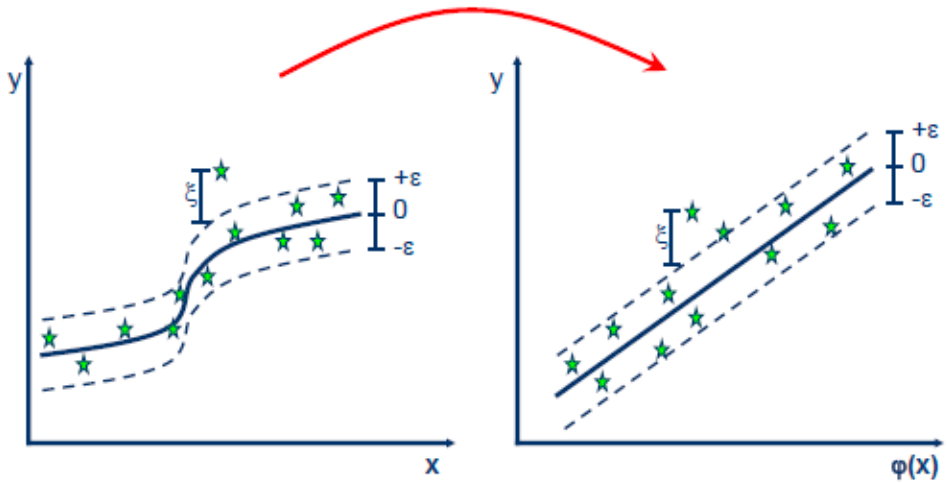
$$y_i - wx_i - b \le \varepsilon + \xi_i$$

$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$

Kernelized SVR

- Just like with SVMs we can use the kernel trick with SVRs



saedsayad.com

Some concluding remarks

- Let’s look at some python code examples in Jupyter notebooks
  - Iris dataset
    - *SVM.Iris.ipynb*
  - Breast cancer dataset
    - *SVM.Breast.ipynb*