Binary Classification, Non-linear SVM, and Kernel Trick

Lesson 8 – Section 2

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

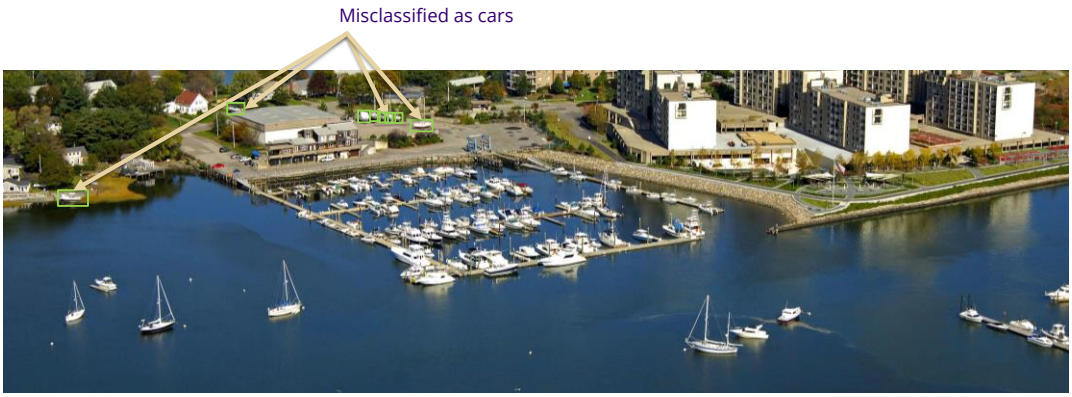
Binary Classification Example

Importance of Slack Variables

Binary Classification: Cars vs. Boats

Binary Classification: Building and Boats

Binary Classification: Cars vs. Boats

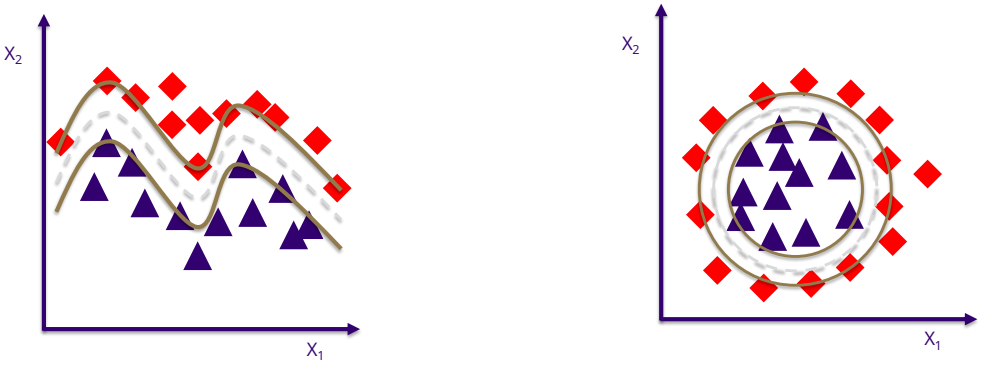


PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Other Types of Data Sets

And the Kernel Trick

Other Non-linearly Separability Data Sets

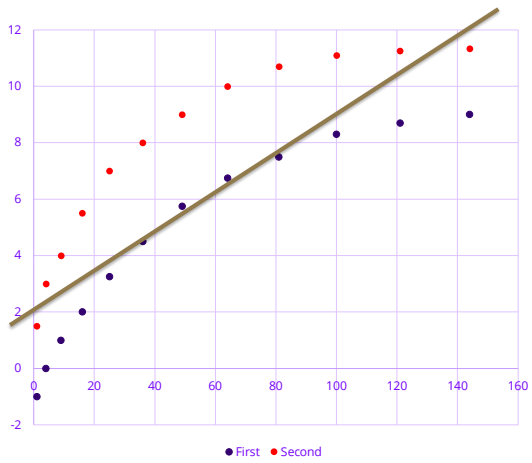


Separation in non-linear data sets is accomplished using a kernel function, of which there are several

The "Kernel Trick"

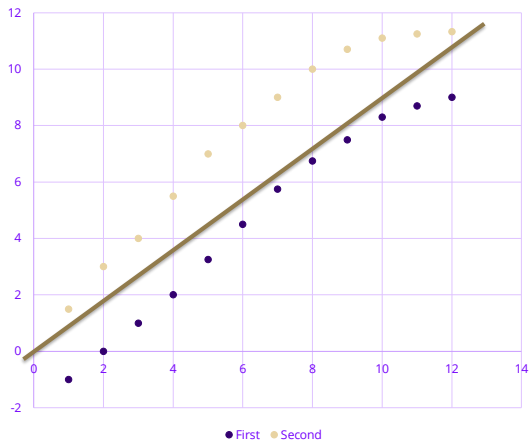
- You have an N-dimensional dataset but for computational reasons you'd rather use a linear machine learning technique to it, but it doesn't work because the dataset is too non-linear
- You could try finding a non-linear separator to determine higher-order surfaces
- Instead you transform your input variables so that the shape of the dataset becomes more linear (e.g., square one of the variables). You do not need to preserve the dimensionality of the original dataset.

Simple Non-linear Example



$f(x)$	First	Second
1	-1	1.5
4	0	3
9	1	4
16	2	5.5
25	3.3	7
36	4.5	8
49	5.8	9
64	6.8	10
81	7.5	10.7
100	8.3	11.1
121	8.7	11.25
144	9	11.33

Simple Non-Linear Example



$f(\sqrt{x})$	First	Second
1	-1	1.5
2	0	3
3	1	4
4	2	5.5
5	3.3	7
6	4.5	8
7	5.8	9
8	6.8	10
9	7.5	10.7
10	8.3	11.1
11	8.7	11.25
12	9	11.33

Nonlinear Support Vector Machines

- Non-parametric
- Good for smaller data sets
- Converting 2D Data to Multidimensional Data
- Common methods:
  - Polynomial kernel
  - Radial Basis Function
  - Sigmoid kernel
  - Gaussian kernel
  - Exponential kernel
  - Among others… *
- Choosing the correct kernel is a non-trivial task

*Piyush Rai, Kernel Methods and Nonlinear Classification. https://www.cs.utah.edu/~piyush/teaching/15-9-slides.pdf

Advantages and Caveats

- Useful in high-dimensional spaces – can work even when the number of dimensions > number of samples
  - Caveat: predictive capability might be poor
- Features are non-parametric
  - Not constricted to a "distribution"
  - In theory, infinite, thus are "assumption free" model
  - No curse of dimensionality
  - Caveat: Computational cost
- Kernel functions can be added together to create even more complex hyperplanes
  - caveat: increases computational complexity
- They give a highly optimal hyperplane
  - caveat: Unlike other methods (e.g., boosted trees) they do not produce probability functions

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Parameters in Support Vector Classification *sklearn.svm.svc*

- Important Hyperparameters:
  - **Kernel** – can be linear, **rbf**, poly, sigmoid,
  - **C** (cost) hyperparameter – higher value adds a higher cost for misclassifications (hard margin) and lower value allows for more leeway (soft margin) – softer margin allow for more generalizability and lower sensitivity to noise. Default is **1.0**
  - **Gamma** – hyperparameter for rbf, poly and sigmoid kernels to configure model sensitivity to feature differences. It defines the distance of influence for a single training example. Low values meaning 'far' and high values meaning 'close'. Default is **1/n** (each input vector has a 1/n influence)
  - **Degree** – hyperparameter for polynomial/exponential kernels, specifies the largest possible exponent. Default is x^3

In general, cost and gamma are way to tune the model for softer or harder margins

Impacts of Kernels on an SVM

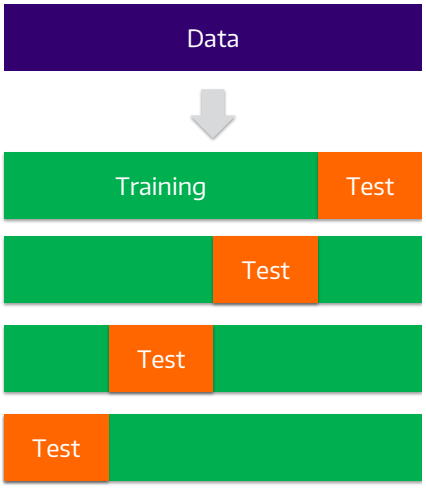
Python Notebook Demo

Choosing a Kernel Function

- Probably the most tricky part of using SVM : Radial Basis Function kernel (RBF) is a good first option…
- Which is best is dependent on the dataset—try several
- It may help to use a combination of several kernels.
- Keep the same training-testing sets when you try different kernels and parameters.

Cross Validation

- Assessing if result will generalize to an independent data set in practice.
- Involves partitioning a sample of data into complementary subsets, performing the analysis on one subset, and validating the analysis on the other subset.
- Often multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.
- Cross-validation is important when you are short on data and it is hazardous, costly or impossible to collect



Fine Tuning a Model

- Grid Search enables you to avoid 'twiddling' hyperparameters
- sklearn.model_selection.GridSearchCV –
  –Where CV is cross validation
- Most important functions are fit, predict
- Performs an exhaustive search over the *specified* parameters and the values
- Parameters are optimized by cross-validated grid-search over a parameter grid

Caveat: Even *more* computationally expensive

Sparse Data

- SVM algorithms speed up dramatically if the data is sparse (i.e. many values are 0)
- Why? Because they compute lots and lots of dot products
- Sparse data compute dot products very efficiently
- SVMs can process sparse datasets with 10,000s of attributes

Building a SVM Model

- Start with linear regression before you move onto LSVM or try both and compare
  - –If results are good enough for one or the other, stop
  - –Else try SVM with a multidimensional kernel (e.g., RBF)
- SVMs require vector of real numbers
  - –Categorical variables numeric data {R,G,B} {0,0,1},…{1,0,0}
  - –May require scaling to the range [-1, +1] or [0,1]
- Use of kernel functions:
  - –RBF is a reasonable first choice
  - –Grid search to identify best values for parameters
  - –Use cross validation to ensure good performance on test data

References

http://scikit-learn.org/stable/auto_examples/applications/plot_face_recognition.html
Hofmann, T, Schölkopf, B, and Smola, AJ (2008) Kernel Methods in Machine Learning, Annals of Statistics, 36:1171-1220.
Ben-Hur, A, Ong, C, Sonnenburg, S, Schölkopf, B, and Rätsch, G (2008) Support Vector Machines and Kernels for Computational Biology, PLoS Computational Biology, 4.
Chen, P, Lin, C, and Schölkopf, B (2003) : http://www.csie.ntu.edu.tw/~cjlin/papers/nusvmtutorial.pdf
Schölkopf, B (2000) The Kernel Trick for Distances, Microsoft Research, TR MSR(2000-51), Redmond, WA.
Schölkopf, B (2000) Statistical Learning and Kernel Methods, Microsoft Research, MSR-TR(2000-23).
Burges, CJ (1998) A Tutorial on Support Vector Machines for Pattern Recognition, Knowledge Discovery and Data Mining, 2(2).

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Summary

- SVMs balance between correctness and generalization
  - Decision boundaries
  - Margins
  - Support vector
- Two key concepts of SVM: maximize the margin and the kernel trick
- Many SVM implementations are available on the web for you to try on your data set!