Support Vector Machines

Ross Mitchell

CSCI-446 Artificial Intelligence

12-2-2015

Overview

- What are Support Vector Machines (SVMs)?
- SVMs uses
- History of SVMs
- SVM Concept
- Practical Guide to SVMs

What are Support Vector Machines (SVMs)?

- SVMs are supervised learning models that analyze data and recognize patterns, used for classification and regression analysis.
- An SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.
- SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

SVMs uses

- Support Vector and Kernel Machines are part of a class of algorithms that detect and exploit complex patterns in data (e.g. by clustering, classifying, ranking, cleaning, etc. the data).
- Typical problems include how to represent complex patterns (a computational problem) and how to exclude spurious, unstable patterns which is overfitting (statistical problem).

History of SVMs

- 1963 - The original SVM algorithm was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis
- 1992 - Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick to maximum-margin hyperplanes
- 1993 - The current standard incarnation (soft margin) was proposed by Corinna Cortes and Vapnik and published in 1995

SVM Concept

- SVM framework is currently the most popular approach for "off-the-shelf" supervised learning: if you don't have any specialized prior knowledge about a domain, then the SVM is an excellent method to try first.
- SVMs build off of Linear Learning Machines and Neural Network findings.

3 Properties that make SVMs Attractive

- SVMs construct a maximum margin separator—a decision boundary with the largest possible distance to example points. This helps them generalize well.
- SVMs create a linear separating hyperplane, but they have the ability to embed the data into a higher-dimensional space, using the so-called kernel trick. The high-dimensional linear separator is actually nonlinear in the original space. This means the hypothesis space is greatly expanded over methods that use strictly linear representations.
- SVMs are a nonparametric method—they retain training examples and potentially need to store them all. On the other hand, in practice they often end up retaining only a small fraction of the number of examples—sometimes as few as a small constant times the number of dimensions. Thus SVMs combine the advantages of nonparametric and parametric models: they have the flexibility to represent complex functions, but they are resistant to overfitting."

SVMs Basic Notation

- Input space $x \in X$
- Output space $y \in Y = \{-1, +1\}$
- Hypothesis $h \in H$
- Real-valued: $f : X \rightarrow R$
- Training Set $S = \{(x1, y1), \ldots, (xi, yi), \ldots\}$
- Test error ε
- Dot product $\langle x, z \rangle$

Perceptron algorithm – discussed in Artificial Neural Networks

Linear separation of the input space using the following:

- $f(x) = \langle w, x \rangle + b$
- $h(x) = sign(f(x))$

The sign function extracts the sign of a real number.

The decision function can be written:

- $f(x) = \langle w, x \rangle + b = \Sigma \alpha_i y_i \langle x_i, x \rangle + b$
- $w = \Sigma \alpha_i y_i x_i$

SVM represented in a dual fashion where the data only appears within the dot products

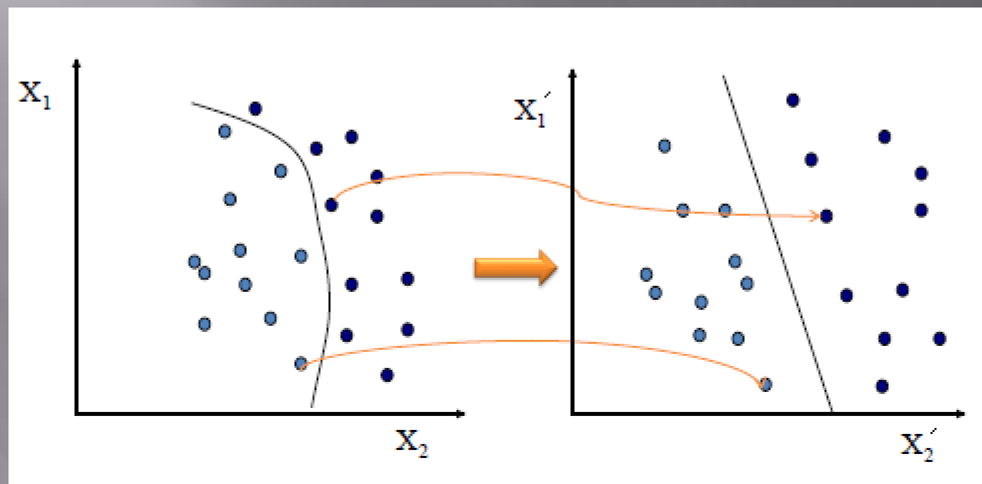
Separating the Data

- The simplest way to separate two groups of data is with a straight line (1 dimension), flat plane (2 dimensions) or an N-dimensional hyperplane.
- Situations where a nonlinear region can separate the groups more efficiently.
- SVM handles this by using a kernel function (nonlinear) to map the data into a different space where a hyperplane (linear) cannot be used to do the separation.
- a non-linear function is learned by a linear learning machine in a high-dimensional feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space.
- This is called kernel trick which means the kernel function transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

The Kernel

- In the dual representation, the data points only appear inside the dot products:
- $f(x) = \Sigma \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b$
- A kernel is a function that returns the value of the dot product between the images of the two arguments:
- $K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$
- Linear Learning Machines can be used in a feature space by rewriting it in dual representation and replacing dot products with kernels:
- $\langle x_1, x_2 \rangle \leftarrow K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$

Map Data into a Feature Space



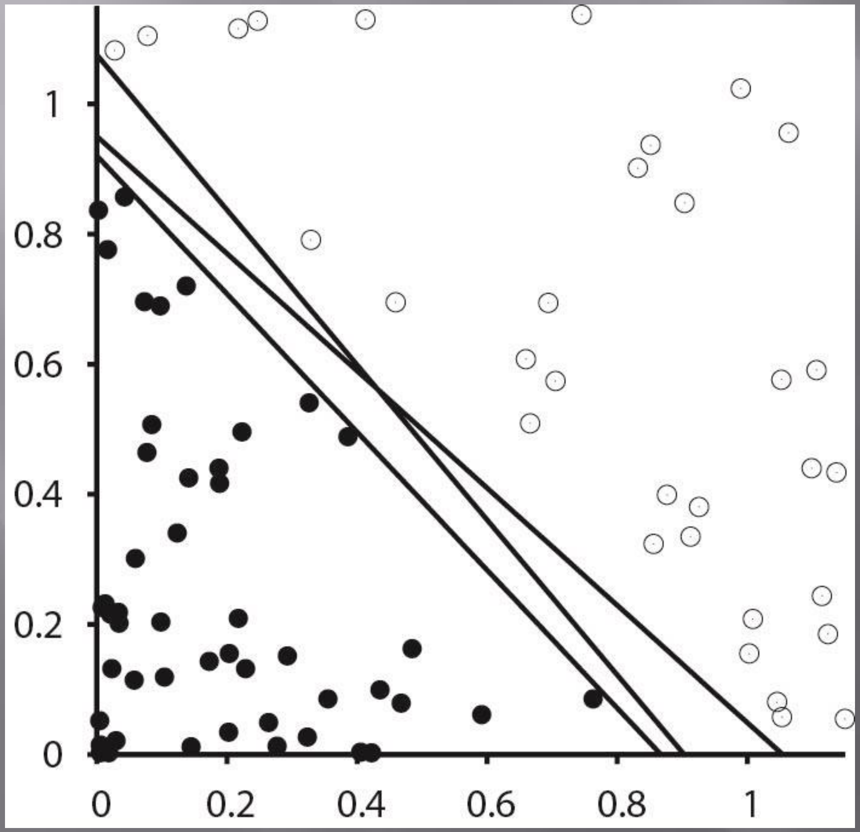
Linear SVM: $x_i \cdot x_j$

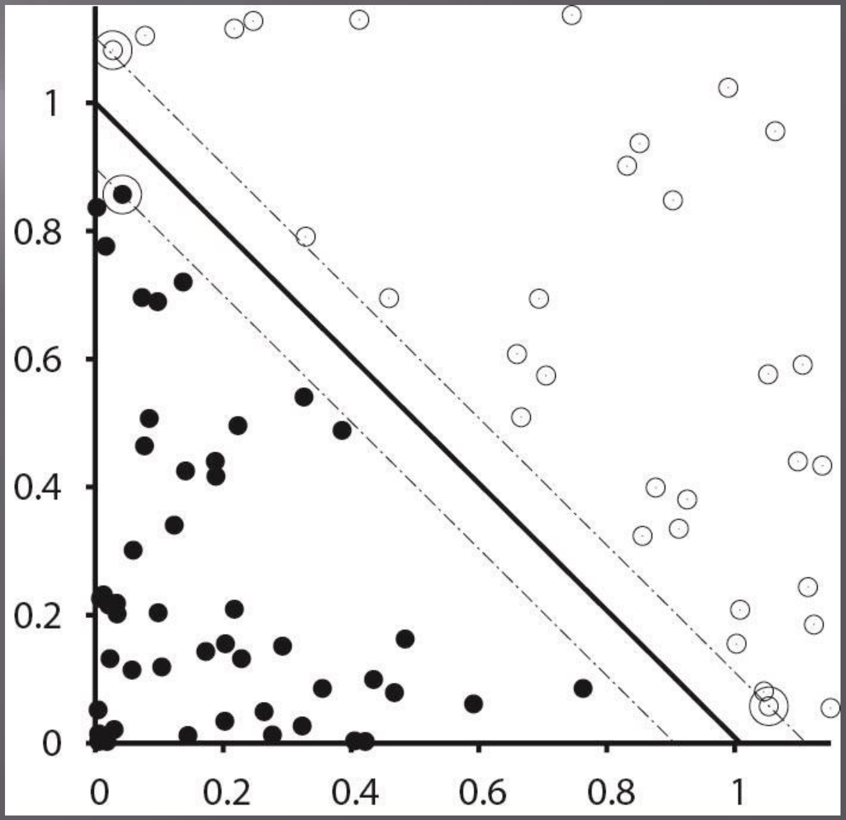
Non-linear SVM: $\phi(x_i) \cdot \phi(x_j)$

Kernel function: $k(x_i \cdot x_j)$

Maximum margin separator

- Support vectors (points with large circles) are the examples closest to the separator.





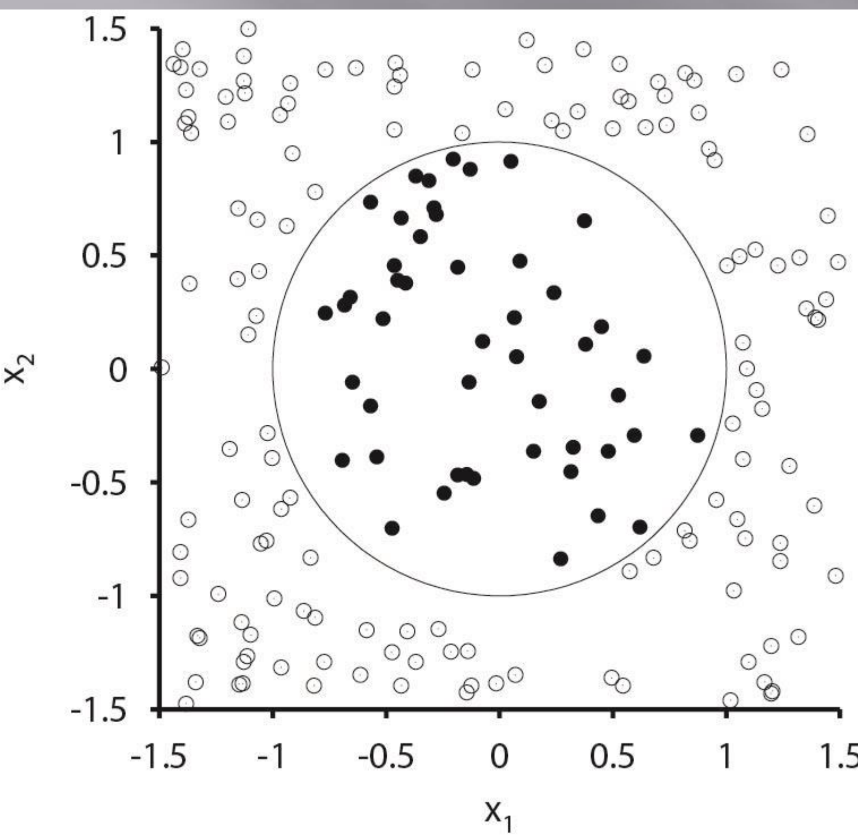
Kernel Matrix

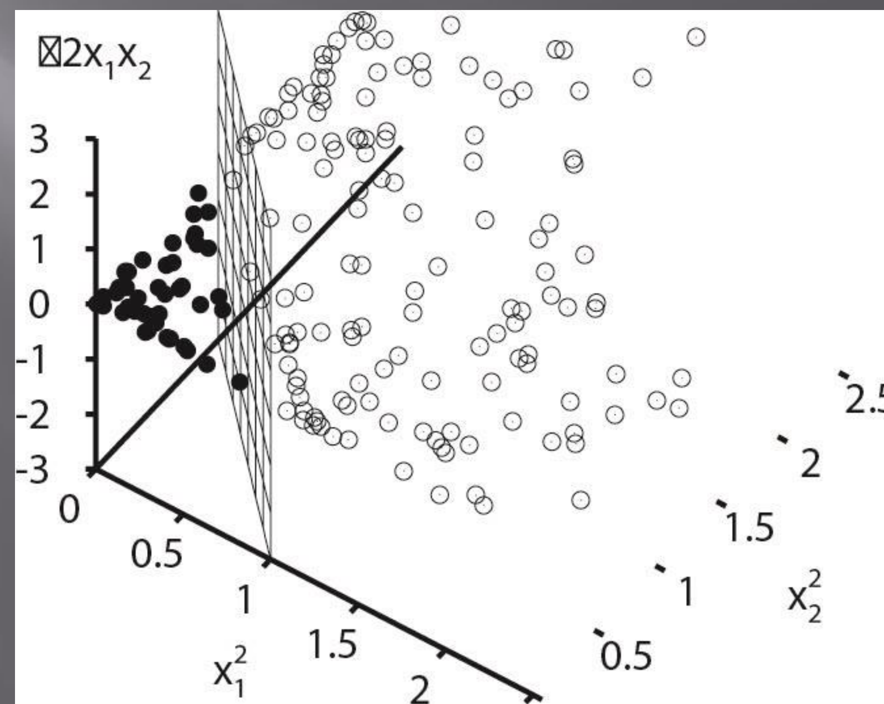
- contains all necessary information for the learning algorithm. The Kernel Matrix fuses information about the data and the Kernel

K(1,1)	K(1,2)	K(1,3)	…	K(1,m)
K(2,1)	K(2,2)	K(2,3)	…	K(2,m)
…	…	…	…	…
K(m,1)	K(m,2)	K(m,3)		K(m,m)

Kernel Function to Evaluate Dot products in Feature Space

- A two-dimensional training set with positive examples as black circles and negative examples as white circles.
- The same data after mapping into a three-dimensional input space





Practical Guide to SVMs

Recommended trying this procedure to get started:

- Transform data to the format of an SVM package
- Conduct simple scaling on the data
- Consider the RBF kernel,
- Use cross-validation to find the best parameter C and γ
- Use the best parameter C and γ to train the whole training set
- Test

Works Cited

- Cristianini, N. (2001). support-vector.net. Retrieved from icml-tutorial: http://www.support-vector.net/icml-tutorial.pdf
- Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2015). A Practical Guide to Support Vector Classification. Taipei, Taiwan: National Taiwan University.
- Russell, S., & Norvig, P. (2015). Artificial Intelligence: A Modern Approach, 3/e, 3rd Edition. Upper Saddle River, New Jersey: Prentice Hall.
- Sayad, S. (2015). Support Vector Machine - Classification (SVM). Retrieved from Dr. Saed Sayad: http://www.saedsayad.com/support_vector_machine.htm
- Support vector machine. (2015, November 17). Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Support_vector_machine