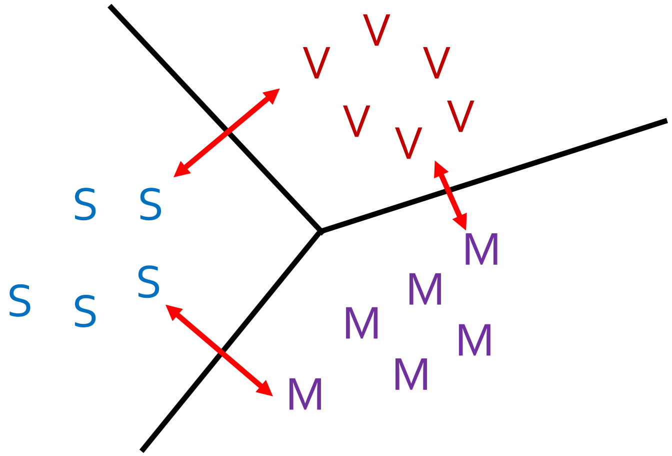
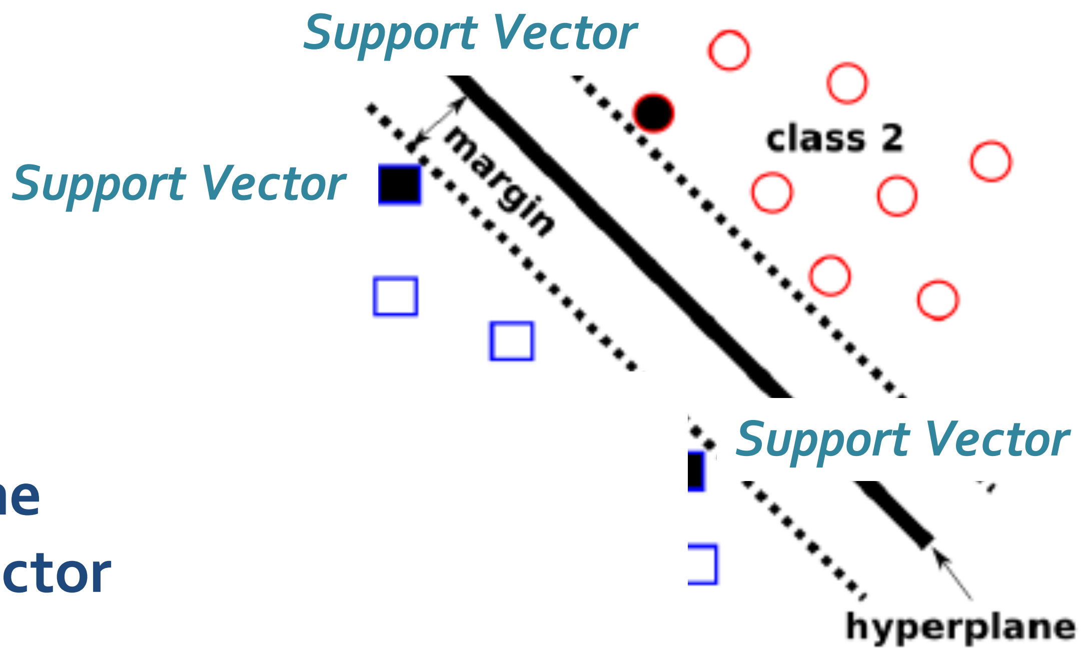


Chapter 9. Advanced Classification: Support Vector Machines (SVMs)

Meng Jiang

CSE 40647/60647 Data Science Fall 2017

Concepts



Quick start:

1. **Hyperplane**
2. **Supper Vector**
3. **Margin**
4. **SVMs**
5. **Maximize Margin Width**
6. **Non-linear SVMs: Kernel Function**

A Classification Problem

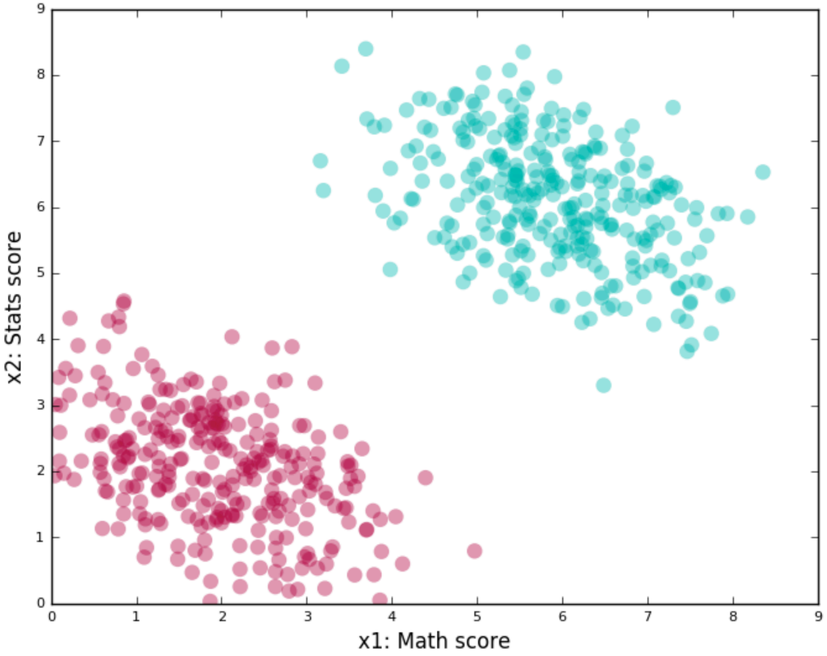
- Dataset
  - Suppose **data science** course instructors have observed that students get the *most out of the course* if they are good at *Math* or *Stats*.
  - Over time, they have recorded the *scores* of the enrolled students in these subjects.
  - Also, for each of these students, they have a *label* depicting their performance in the data science course: "*Good*" or "*Bad*".

A Classification Problem (cont.)

- Research problem
  - Now they want to determine the *relationship* between Math and Stats *scores* and the *performance (label)* in the ML course.
- Application
  - Perhaps, based on what they find, they want to specify a *prerequisite for enrolling in the course*.

A Classification Problem (cont.)

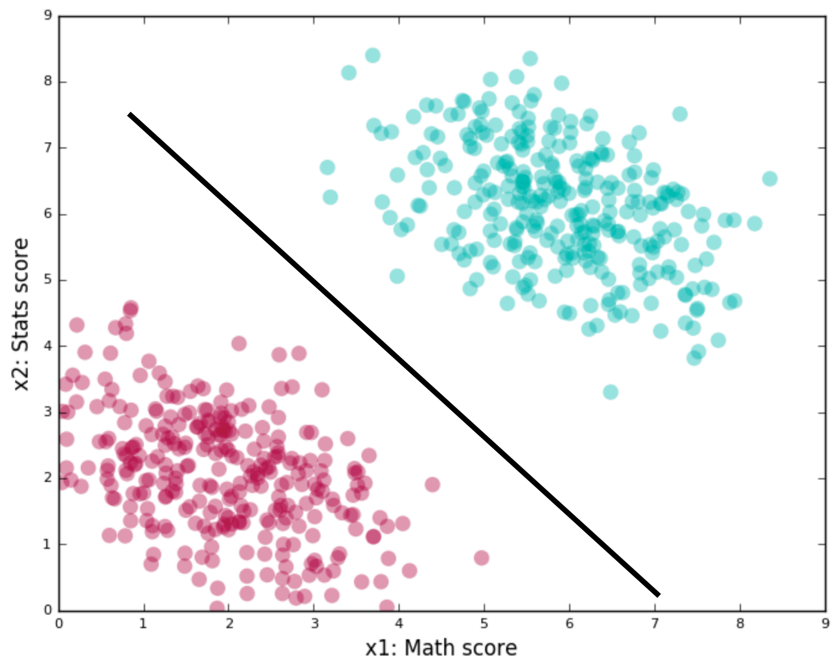
- Data Preprocessing + Visualization
  - Draw a two-dimensional *scatter plot*, where one axis represents scores in Math, while the other represents scores in Stats.
  - A *student* with certain scores is shown as a *point* on the graph.
  - The color of the point – *green* or *red* – represents how he/she did on the data science course: "*Good*" or "*Bad*" respectively.



Training data

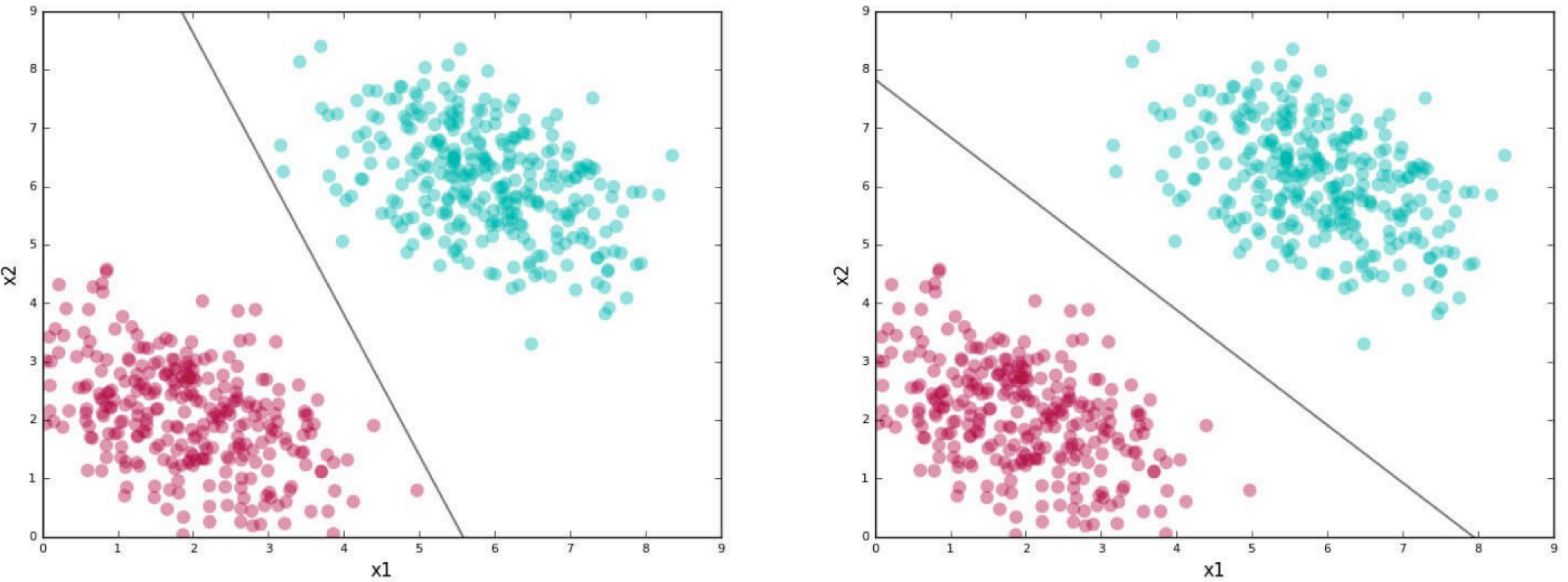
A Classification Problem (cont.)

- Goal
  - Finding a ***line*** that passes between the red and green clusters, and then *determining which side* of this line a score tuple lies on, is a good algorithm.



A Classification Problem (cont.)

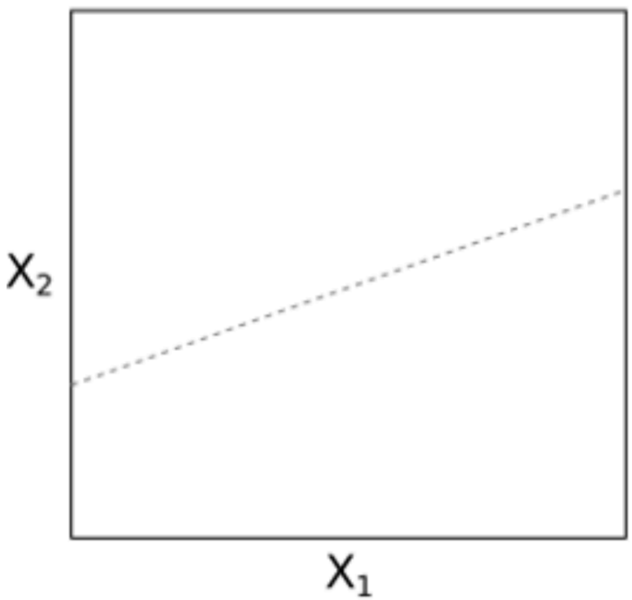
- The ***line*** is our *separating boundary* (because it separates out the labels) or *classifier* (we use it classify points).
- Two classifiers:

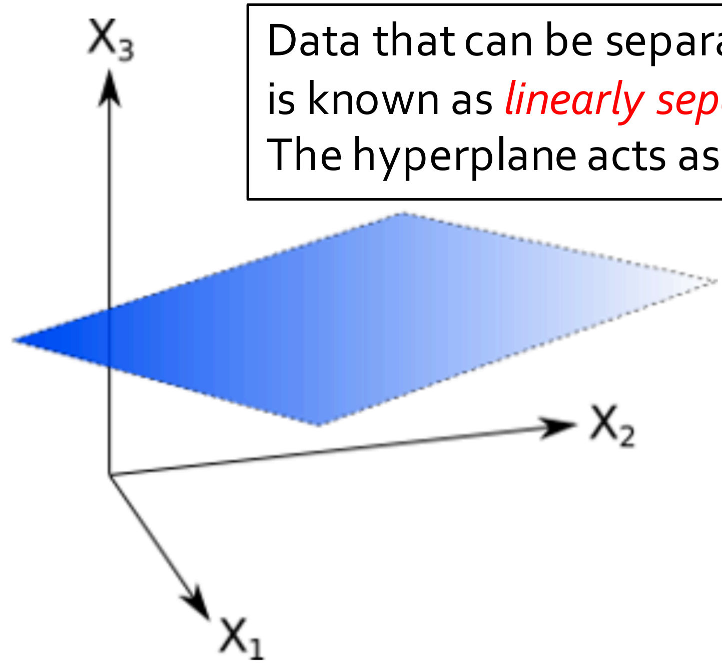


Concept 1: Hyperplane

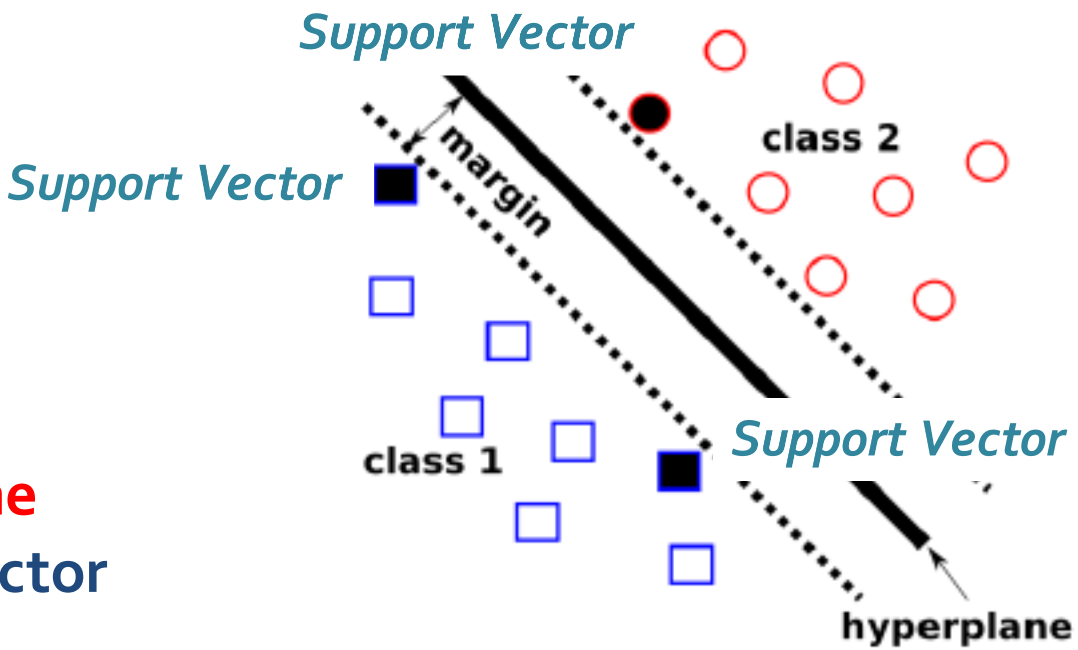
- In two dimensions we find a *line*.
- In three dimensions we find a *plane*.
- In high dimensions we find a *hyperplane* – a generalization of the two-dimensional line and three-dimensional plane to an *arbitrary number of dimensions*.

Data that can be separated by a hyperplane is known as *linearly separable* data.
The hyperplane acts as a *linear classifier*.





Concepts

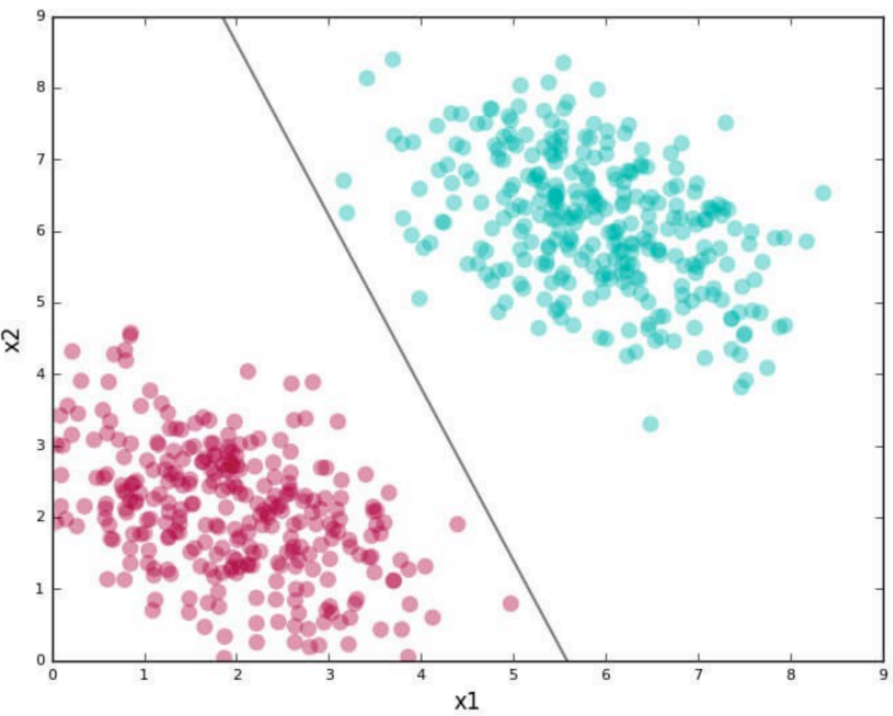


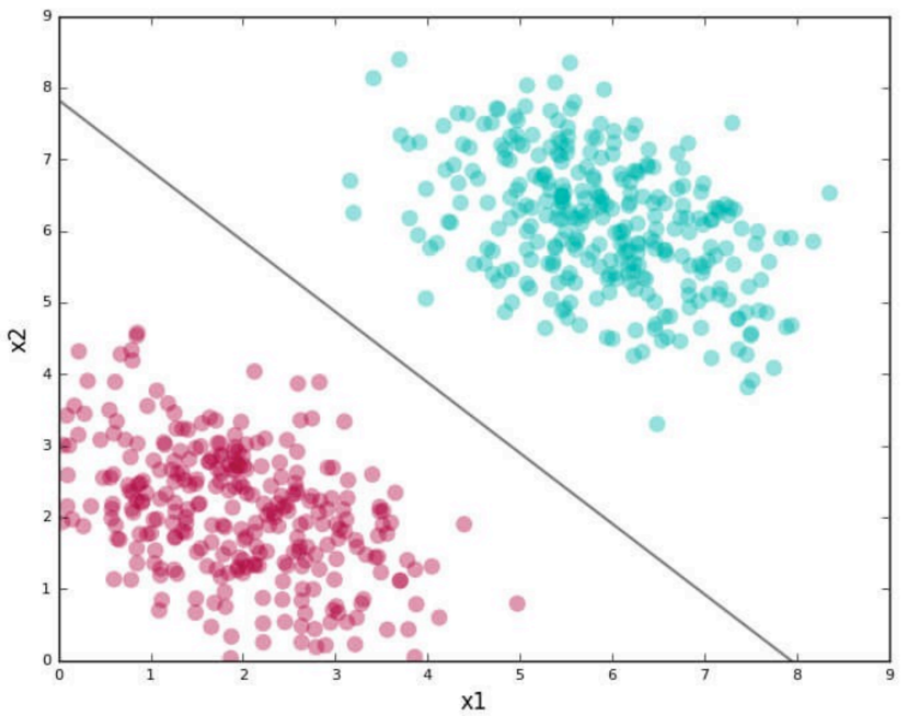
Quick start:

1. Hyperplane
2. Supper Vector
3. Margin
4. SVMs
5. Maximize Margin Width
6. Non-linear SVMs: Kernel Function

Good vs Bad Classifiers

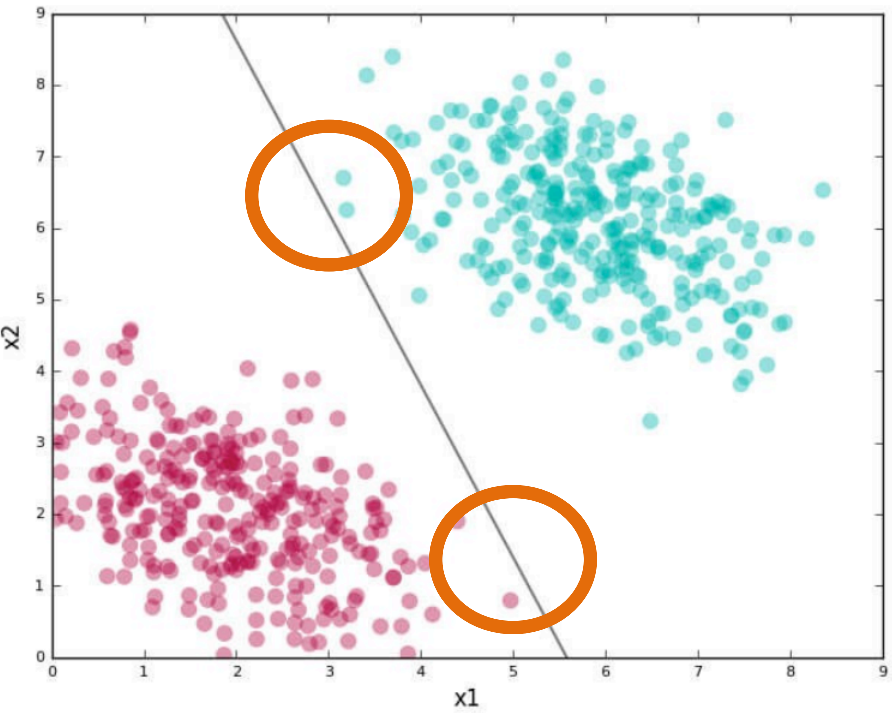
Q: Both lines separate the red and green clusters. Is there a good reason to choose one over another?

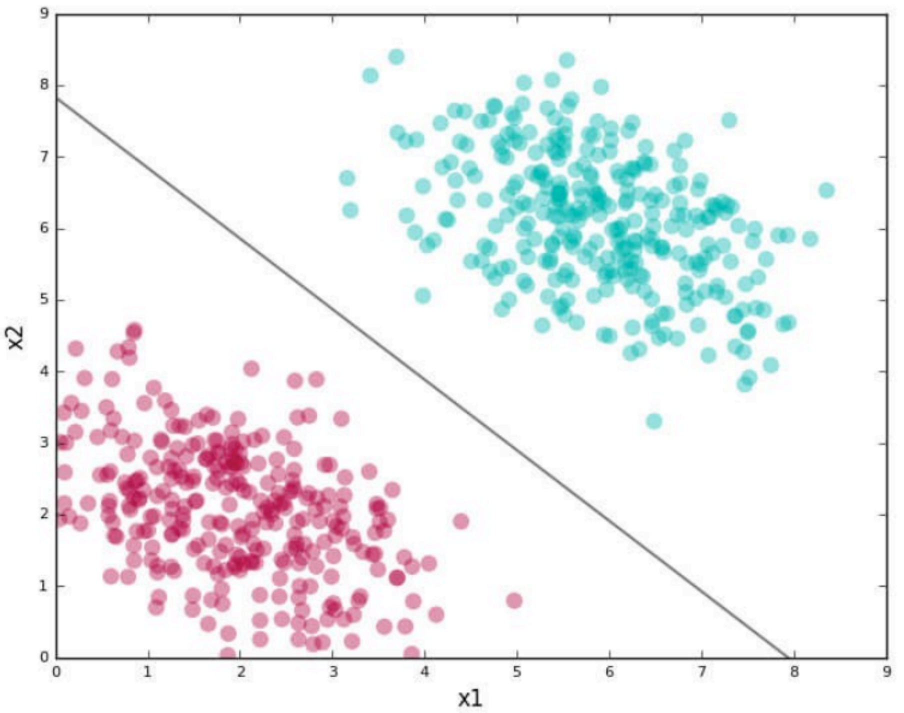




Good vs Bad Classifiers

Q: Both lines separate the red and green clusters. Is there a good reason to choose one over another?

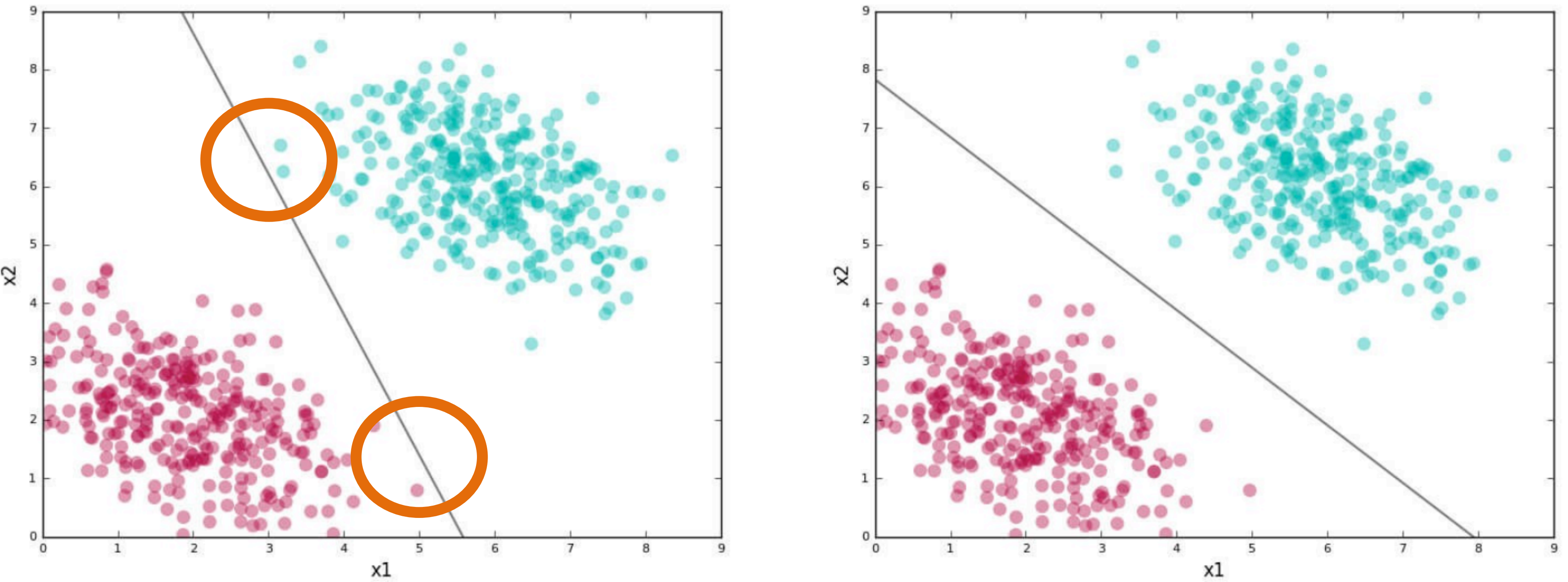




Good vs Bad Classifiers

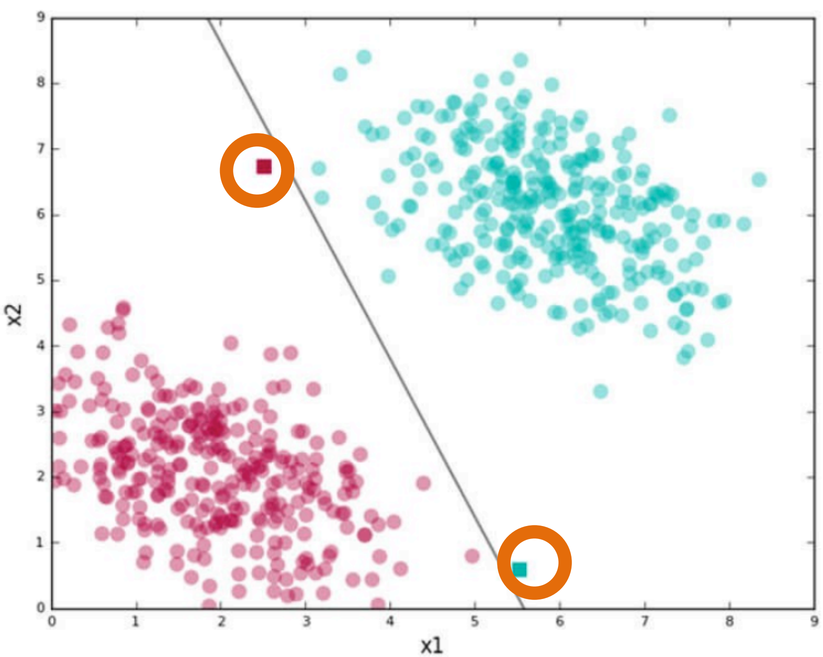
A: We try to find the second kind of line.

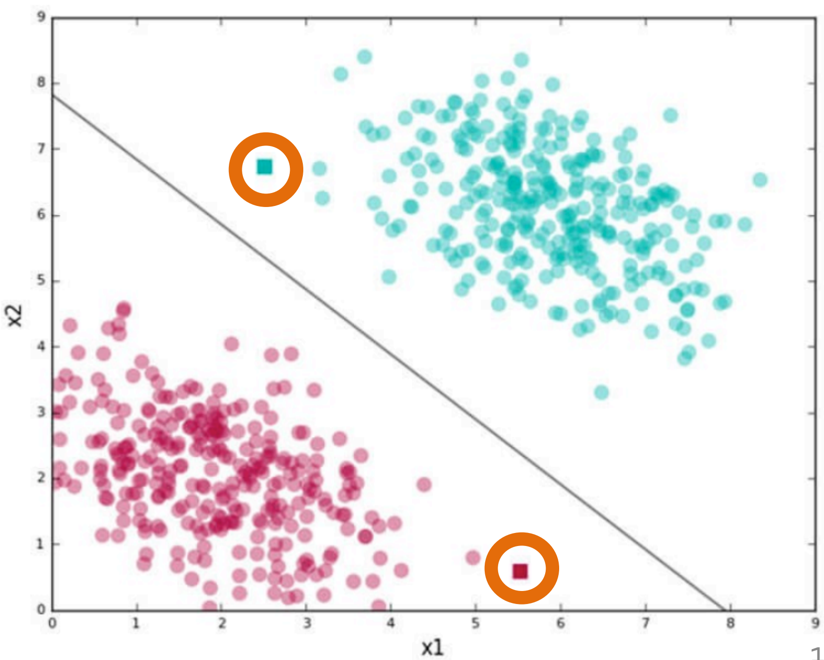
Q': Define ***underlying philosophy*** in the general case.



Good vs Bad Classifiers

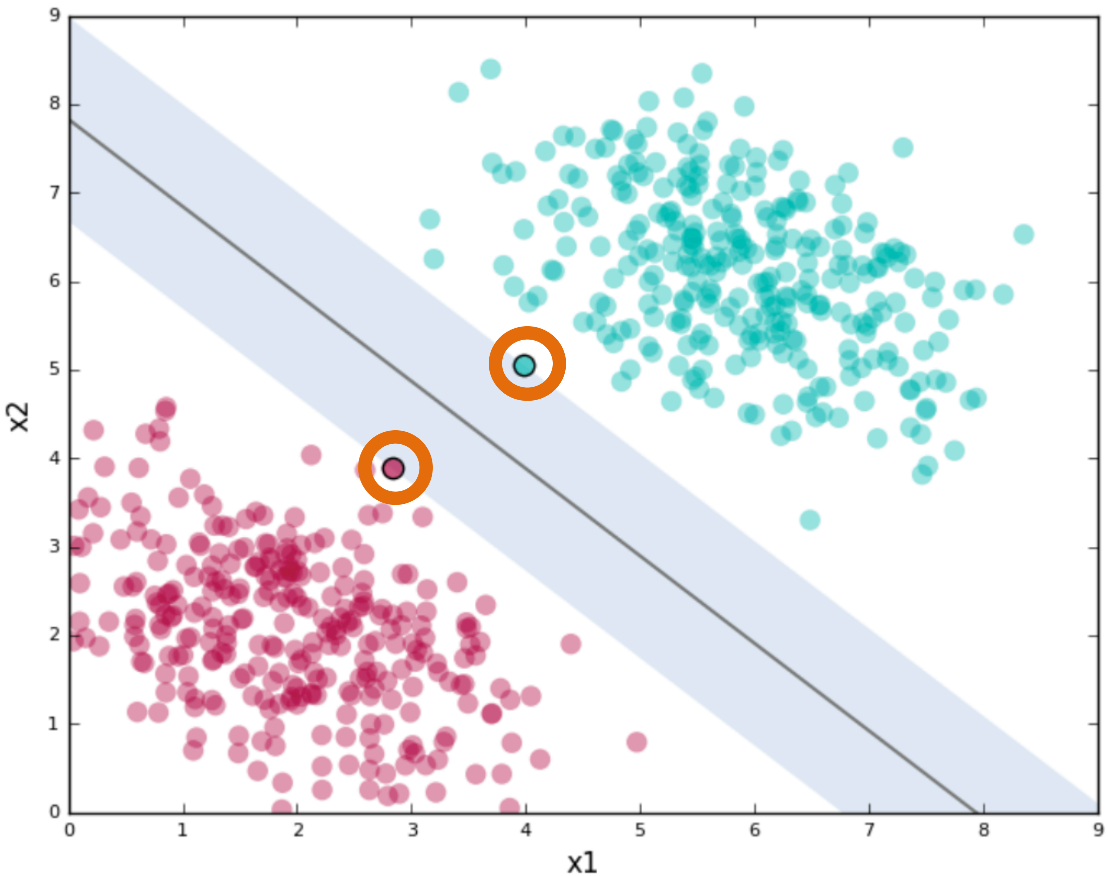
- Philosophy
  - Find lines that *correctly* classify the training data
  - Among all such lines, pick the one that has the *greatest distance* to the points closest to it





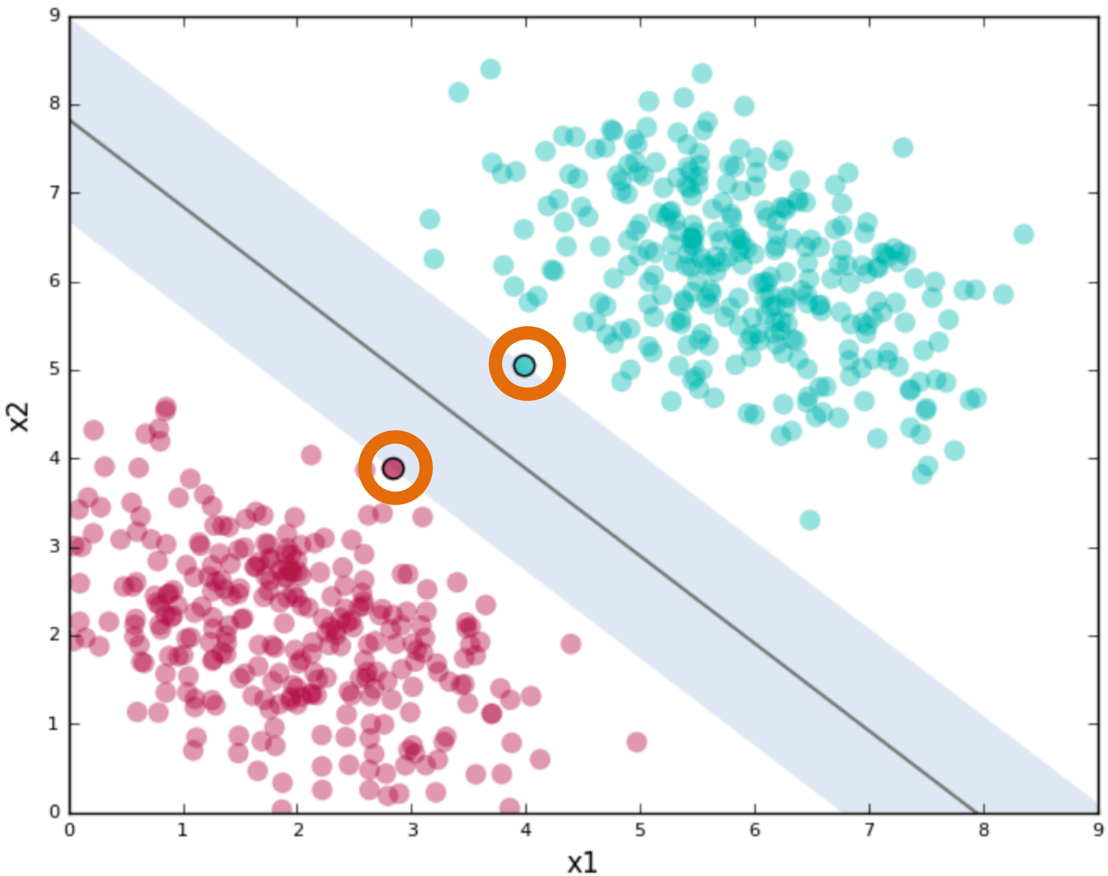
Concept 2: Support Vector

- The *closest points* that identify this line are known as ***support vectors***.

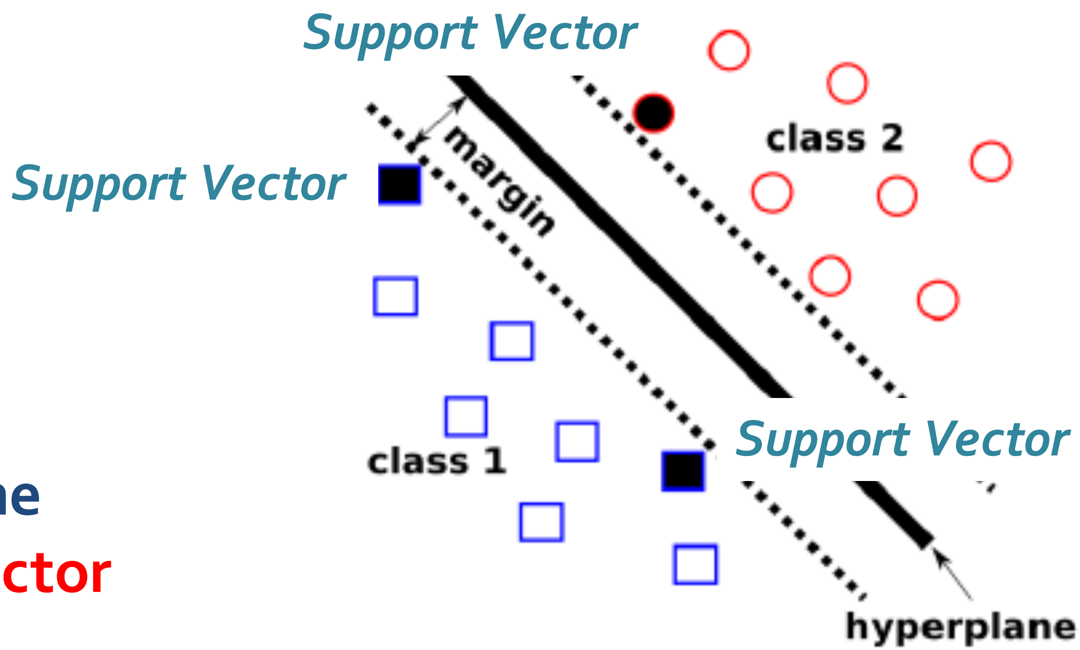


Concept 3: Margin

- The *(shaded) region* support vectors define around the line is known as the ***margin***.



Concepts



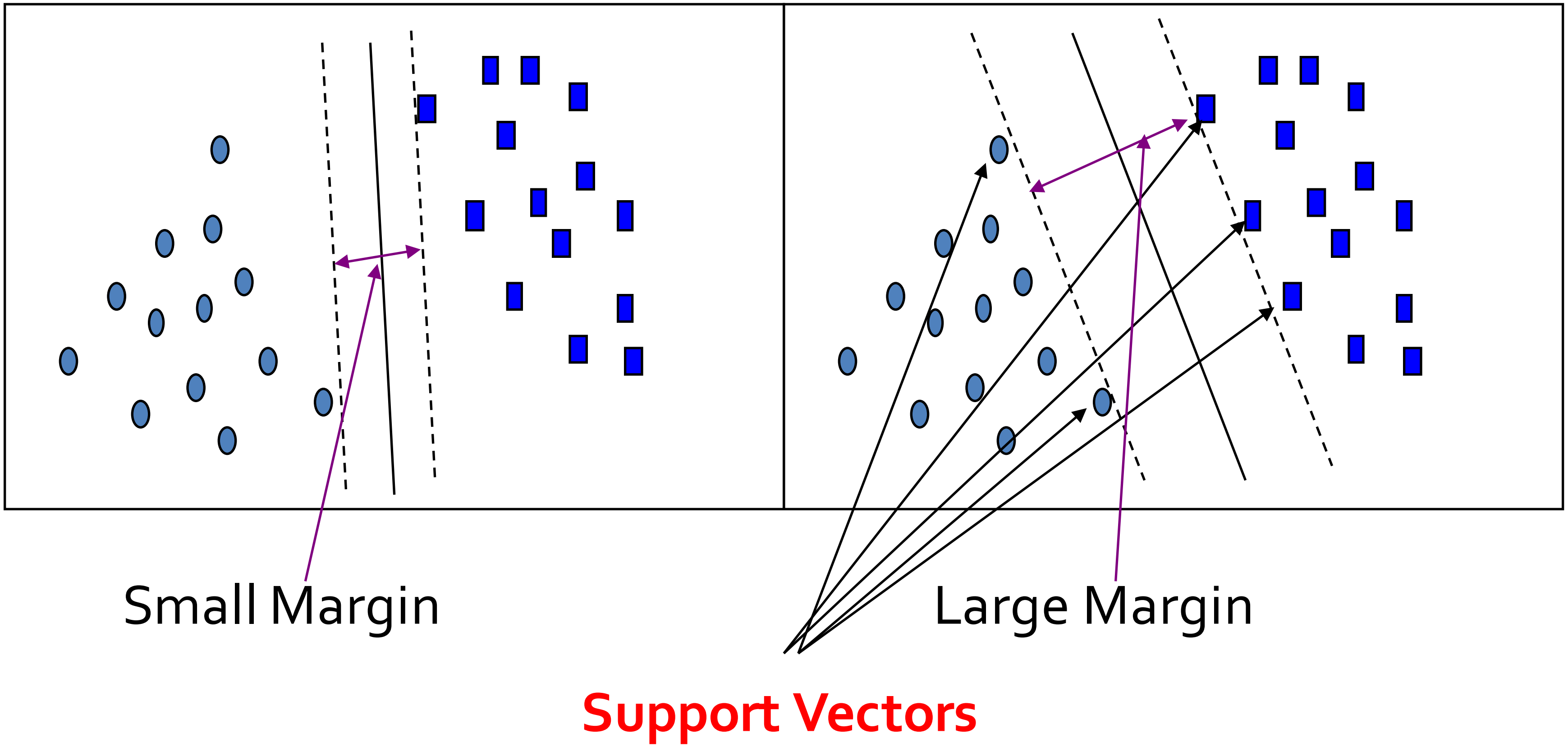
Quick start:

1. **Hyperplane**
2. **Supper Vector**
3. **Margin**
4. **SVMs**
5. **Maximize Margin Width**
6. **Non-linear SVMs: Kernel Function**

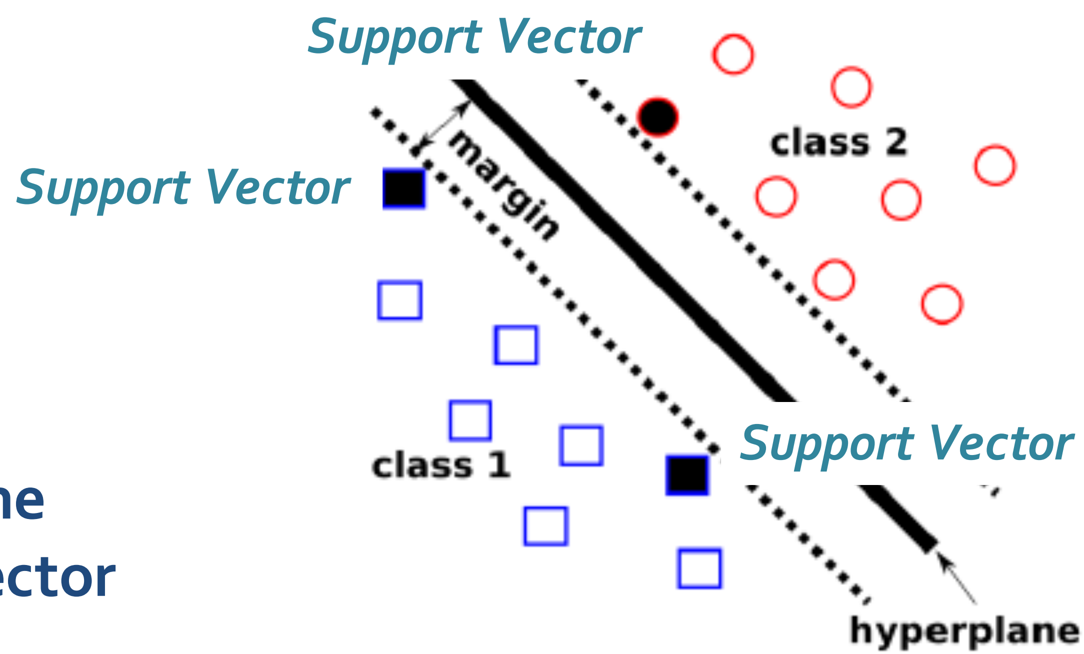
Support Vector Machines

- Definition (Wikipedia)
  - A SVM is a discriminative classifier formally defined by a *separating hyperplane*. In other words, given *labeled training data*, the algorithm outputs an *optimal hyperplane* which categorizes ***new examples***.
  - A *data point* is viewed as a *p*-dimensional vector, and we want to know whether we can separate such points with a *(p-1)*-dimensional hyperplane.
- Use
  - SVMs give you a way to *pick between many possible classifiers* in a way that guarantees a higher chance of correctly labeling your ***test data***.

SVM: General Philosophy



Concepts

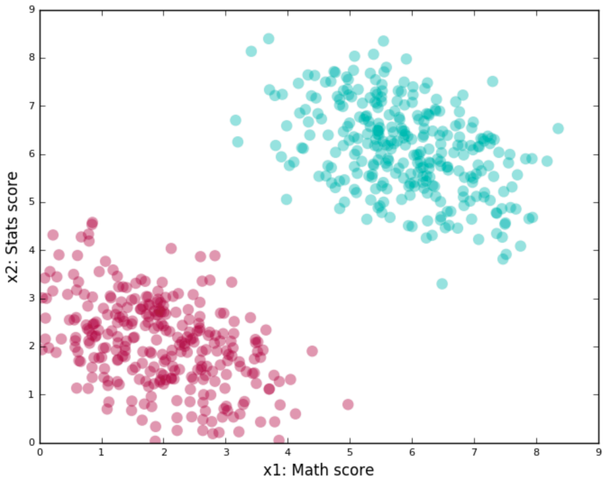


Quick start:

1. Hyperplane
2. Supper Vector
3. Margin
4. SVMs
5. Maximize Margin Width
6. Non-linear SVMs: Kernel Function

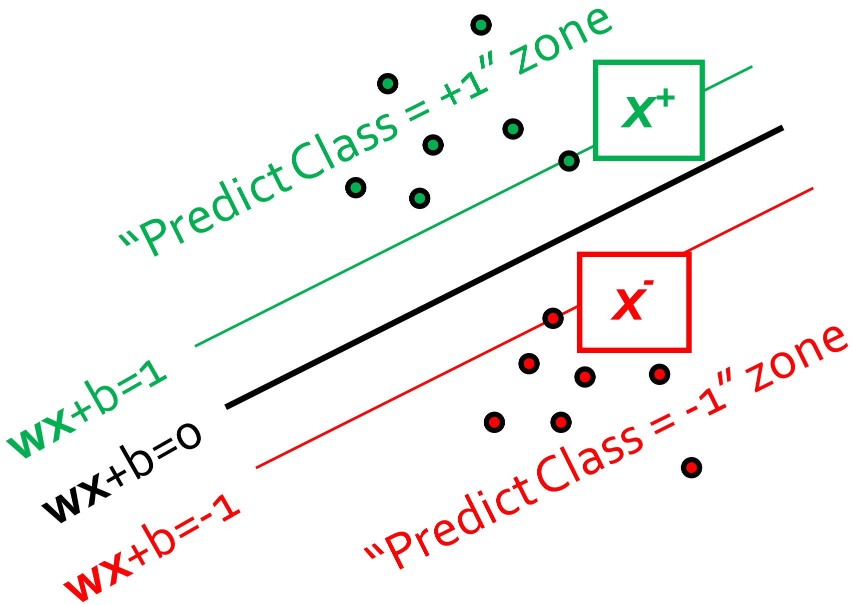
Formally Define the Problem

- Binary Classification
  - E.g., course performance classification
    - $x_i = (x_1, x_2, x_3, \ldots)$: Subject scores
    - $y_i = +1$ or -1: "Good" or "Bad"
    - x_1 : score of subject Math
    - x_2 : score of subject Stats



- Mathematically, $x \in \Re^n$, $y \in \{+1, -1\}$,
  - We want to derive a function **f: X → Y**

Optimization



Support vectors: $\boldsymbol{x}^{+}$ $\boldsymbol{x}^{-}$

Hyperplane: **wx**+b=0

What we know:

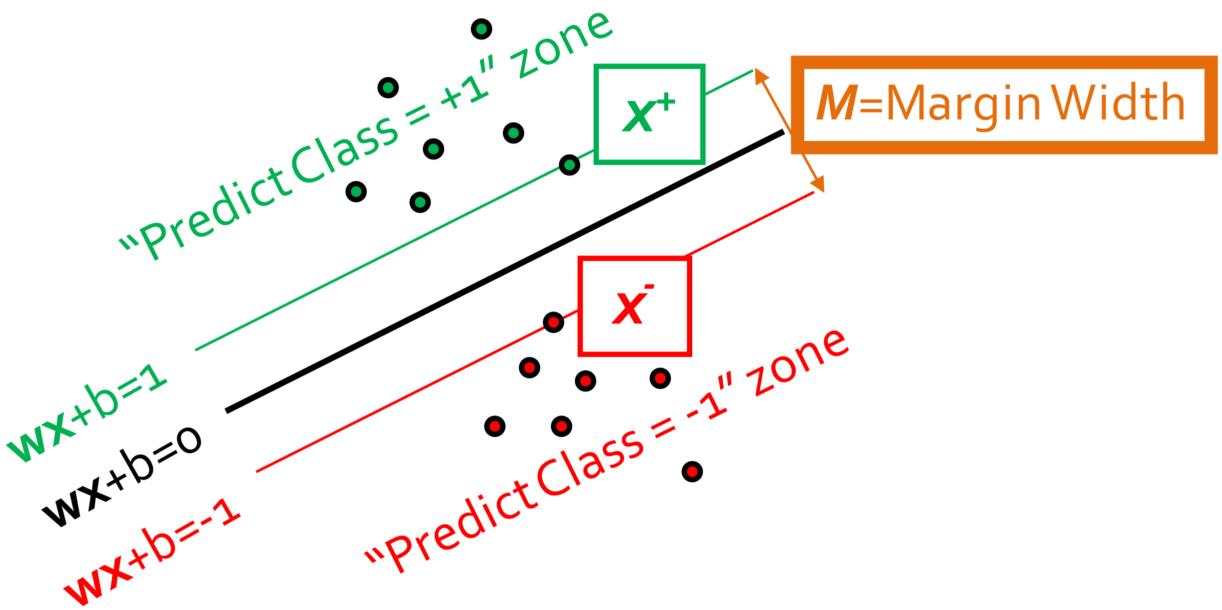
- $\mathbf{w} \, . \, \mathbf{x}^{+} + b = +1$
- $\mathbf{w} \, . \, \mathbf{x}^{-} + b = -1$

A separating hyperplane can be written as

$$\mathbf{w} \, . \, \mathbf{x} + b = 0,$$

where **w**=$\{w_1, w_2, \ldots, w_n\}$ is a weight vector and *b* a scalar (bias).

Optimization



Support vectors: x^+ x^-

Hyperplane: **wx**+b=0

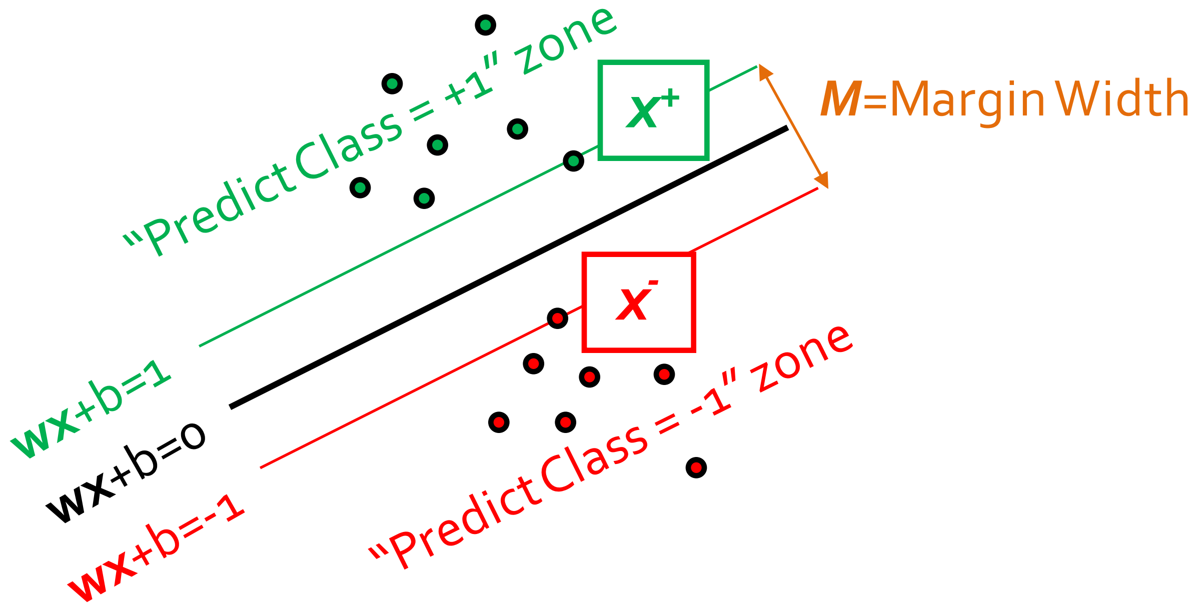
What we know:

- $\boldsymbol{w} \cdot \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \cdot \boldsymbol{x}^- + b = -1$

How to calculate the Margin Width?

Distance between two parallel lines.

Optimization: Maximize Margin Width



Support vectors: x^+ x^-

Hyperplane: **wx**+b=0

What we know:

- $\boldsymbol{w} \,.\, \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \,.\, \boldsymbol{x}^- + b = -1$

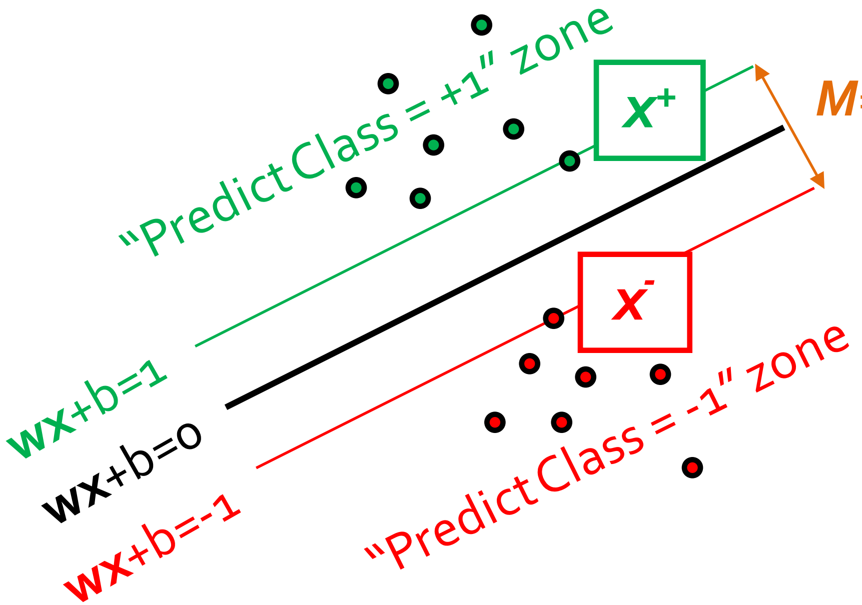
max $M = \dfrac{(x^+ - x^-) \cdot w}{|w|} = \dfrac{2}{|w|}$

same as *min* $\dfrac{1}{2} w^t w$

Any constraints?

Correctly categorize…

Optimization: Maximize Margin Width



Support vectors: x^+ x^-

Hyperplane: **wx**+b=0

What we know:

- $\boldsymbol{w} \,.\, \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \,.\, \boldsymbol{x}^- + b = -1$

max $M = \dfrac{(x^+ - x^-)\cdot w}{|w|} = \dfrac{2}{|w|}$

same as *min* $\dfrac{1}{2} w^t w$

H_1: $\mathbf{wx_i} + b \geq 1$ for $y_i = +1$

H_2: $\mathbf{wx_i} + b \leq -1$ for $y_i = -1$

→ $y_i (\mathbf{wx_i} + b) \geq 1$ for all i

The MMW Problem

$$\min \ \Phi(w) = \frac{1}{2} w^t w$$

$$\text{s.t.} \ y_i (w x_i + b) \geq 1 \ \text{for all i}$$

It's a Constrained (Convex) Quadratic Optimization Problem!

Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.

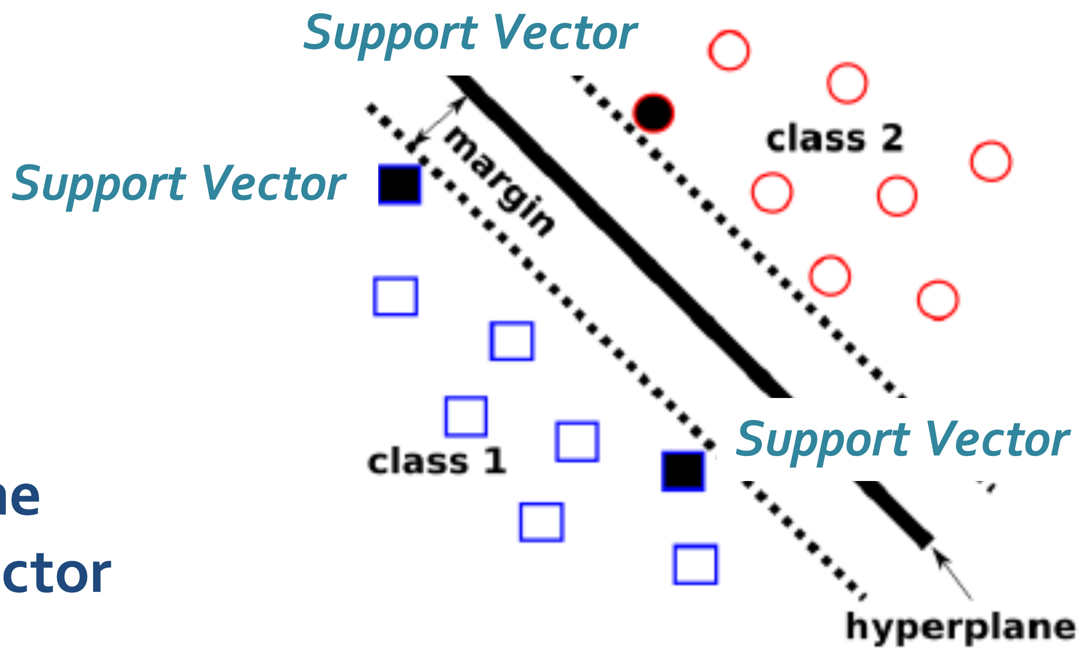
Solving the MMW Problem

- Need to optimize a quadratic function subject to linear constraints.
- Quadratic objective function and linear constraints → *Quadratic Programming* (*QP*) → Lagrangian multipliers
- The solution involves constructing a dual problem where a **Lagrange multiplier α_i** is associated with every constraint in the primary problem:

Find **w** and b such that
$\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$
is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

Concepts



Quick start:

1. Hyperplane
2. Supper Vector
3. Margin
4. SVMs
5. Maximize Margin Width
6. Non-linear SVMs: Kernel Function

LinearSVM: Summary

- The classifier is a separating hyperplane.
  - Linearly separable data points
- Most "important" training points are support vectors; they define the hyperplane.

maximum marginal hyperplane

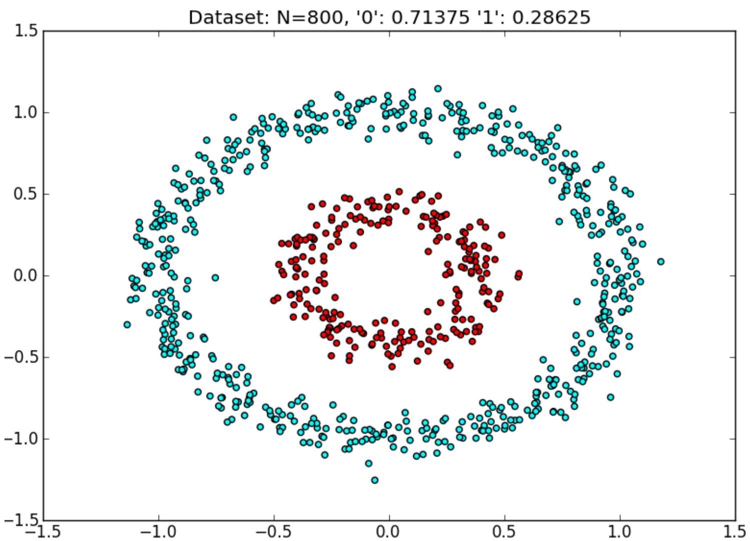
- Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i.

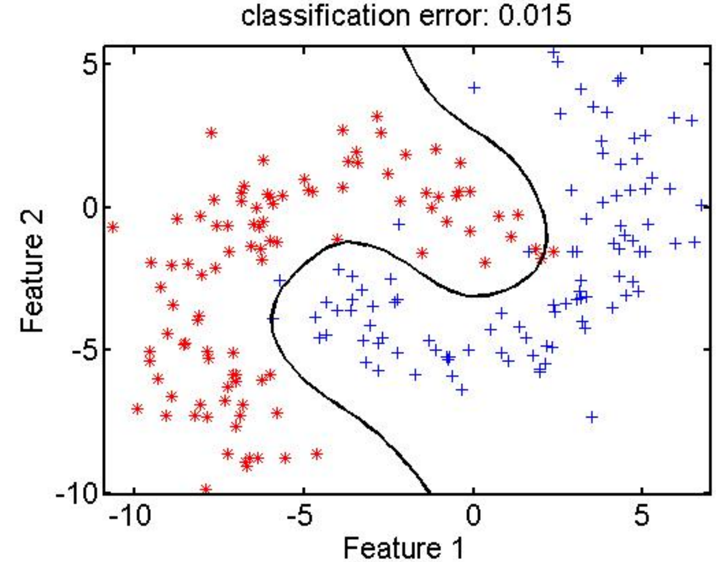
Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j x_i^T x_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

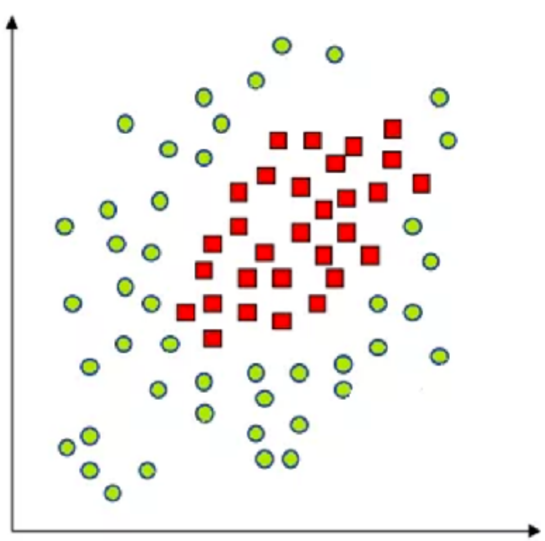
Why is SVM Effective on High Dimensional Data?

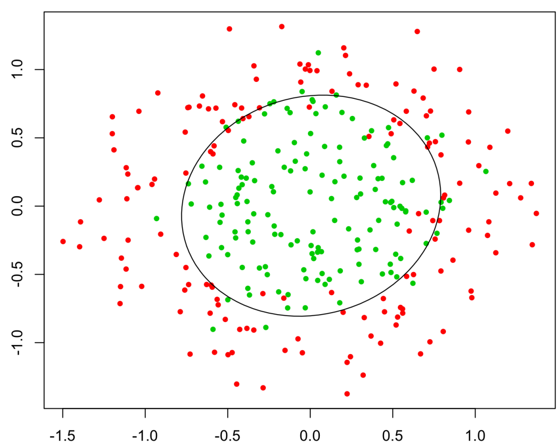
- The complexity of trained classifier is characterized by the *# of support vectors* rather than the *dimensionality of the data*.
- The *support vectors* are the *essential or critical* training examples — they lie closest to the decision boundary.
- If all other training examples are removed and the training is repeated, the ***same*** *separating hyperplane* would be found.

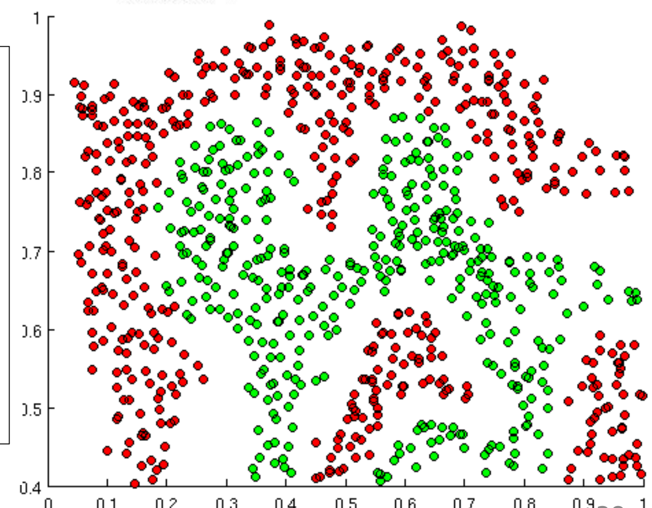
Non-Linear Data Points





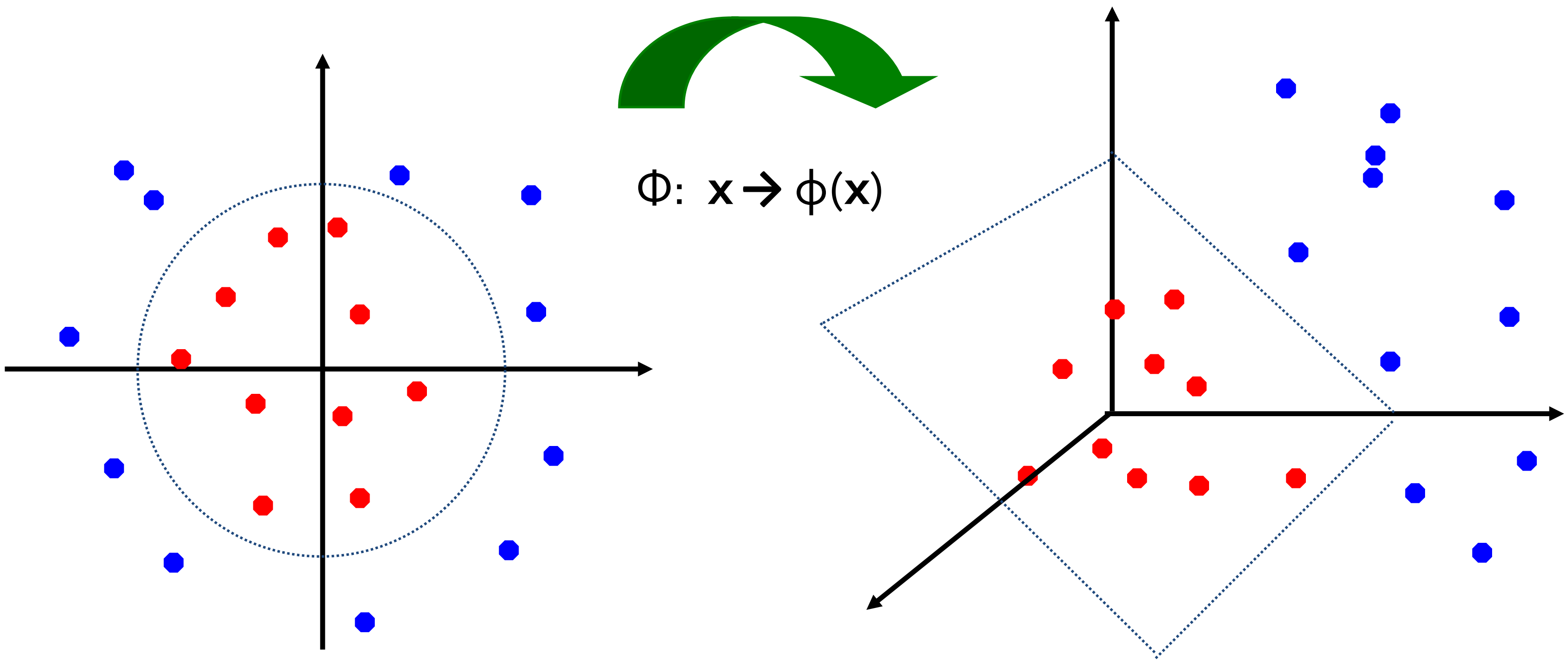






Non-linear SVMs: Feature Spaces

- General idea: The original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



The "Kernel Trick"

- The classifier relies on dot product between vectors

$$K(x_i, x_j) = x_i^T x_j$$

- If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \phi(x)$, the dot product becomes:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.

What Functions are Kernels?

- For some functions $K(x_i, x_j)$ checking that

  $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ can be cumbersome.
- Mercer's theorem:

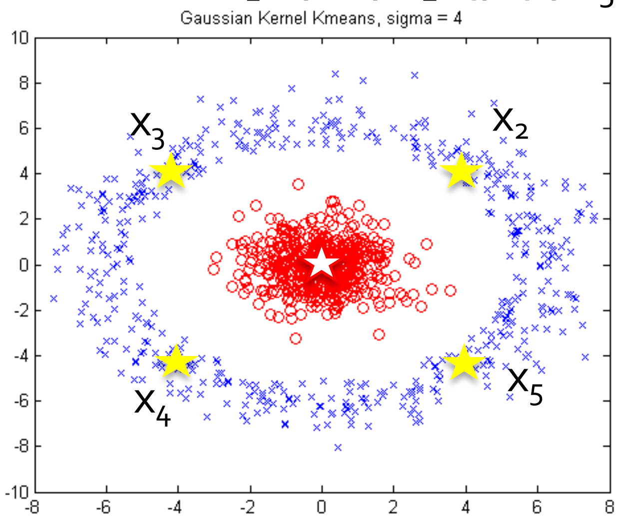
  Every semi-positive definite symmetric function is a kernel
- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

K =

$K(\mathbf{x}_1,\mathbf{x}_1)$	$K(\mathbf{x}_1,\mathbf{x}_2)$	$K(\mathbf{x}_1,\mathbf{x}_3)$	…	$K(\mathbf{x}_1,\mathbf{x}_N)$
$K(\mathbf{x}_2,\mathbf{x}_1)$	$K(\mathbf{x}_2,\mathbf{x}_2)$	$K(\mathbf{x}_2,\mathbf{x}_3)$		$K(\mathbf{x}_2,\mathbf{x}_N)$
…	…	…	…	…
$K(\mathbf{x}_N,\mathbf{x}_1)$	$K(\mathbf{x}_N,\mathbf{x}_2)$	$K(\mathbf{x}_N,\mathbf{x}_3)$	…	$K(\mathbf{x}_N,\mathbf{x}_N)$

Example: A Kernel Function

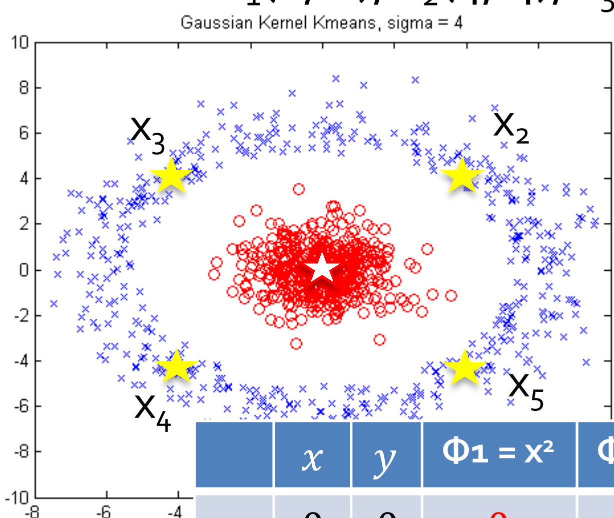
- Polynomial kernel of degree h=2: $K(\boldsymbol{X}_i, \boldsymbol{X}_j) = \boldsymbol{X}_i \cdot \boldsymbol{X}_j^{\ 2} \rightarrow \phi(x, y) = (x^2, \sqrt{2}xy, y^2)$
- Suppose there are 5 original 2-dimensional points:
  - $x_1(0, 0), x_2(4, 4), x_3(-4, 4), x_4(-4, -4), x_5(4, -4)$

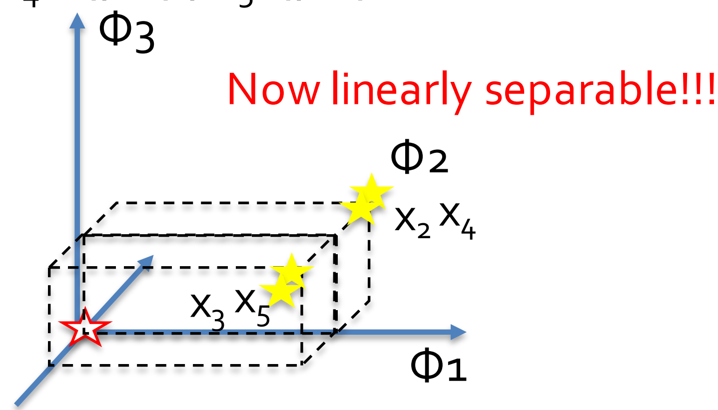


	x	y	$\Phi 1 = x^2$	$\Phi 2 = xy$	$\phi 3 = y^2$
x_1	0	0	0	0	0
x_2	4	4	16	$16\sqrt{2}$	16
x_3	-4	4	16	$-16\sqrt{2}$	16
x_4	-4	-4	16	$16\sqrt{2}$	16
x_5	4	-4	16	$-16\sqrt{2}$	16

Example: A Kernel Function

- Polynomial kernel of degree h=2: $K(\boldsymbol{X_i}, \boldsymbol{X_j}) = \boldsymbol{X_i} \cdot \boldsymbol{X_j}^{\,2} \rightarrow \phi(x, y) = (x^2, \sqrt{2}xy, y^2)$
- Suppose there are 5 original 2-dimensional points:
  - $x_1(0, 0)$, $x_2(4, 4)$, $x_3(-4, 4)$, $x_4(-4, -4)$, $x_5(4, -4)$





	x	y	Φ1 = x²	Φ2 = xy	ϕ3 = y²	$K(x_i,x_1)$	$K(x_i,x_2)$	$K(x_i,x_3)$	$K(x_i,x_4)$	$K(x_i,x_5)$
x_1	0	0	0	0	0	0	0	0	0	0
x_2	4	4	16	$16\sqrt{2}$	16	0	32^2	0	32^2	0
x_3	-4	4	16	$-16\sqrt{2}$	16	0	0	32^2	0	32^2
x_4	-4	-4	16	$16\sqrt{2}$	16	0	32^2	0	32^2	0
x_5	4	-4	16	$-16\sqrt{2}$	16	0	0	32^2	0	32^2

Kernel Functions for Nonlinear Classification

- Instead of computing the dot product on the transformed data, it is mathematically equivalent to applying a kernel function $K(\mathbf{X_i}, \mathbf{X_j})$ to the original data, i.e., $K(\mathbf{X_i}, \mathbf{X_j}) = \Phi(\mathbf{X_i})\, \Phi(\mathbf{X_j})$.
- Typical Kernel Functions

$$\textbf{Polynomial kernel of degree } h: \quad K(\boldsymbol{X_i}, \boldsymbol{X_j}) = (\boldsymbol{X_i} \cdot \boldsymbol{X_j} + 1)^h$$

$$\textbf{Gaussian radial basis function kernel}: \quad K(\boldsymbol{X_i}, \boldsymbol{X_j}) = e^{-\|X_i - X_j\|^2 / 2\sigma^2}$$

$$\textbf{Sigmoid kernel}: \quad K(\boldsymbol{X_i}, \boldsymbol{X_j}) = \tanh(\kappa \boldsymbol{X_i} \cdot \boldsymbol{X_j} - \delta)$$

Non-linear SVMs: Optimization

- Dual problem formulation:

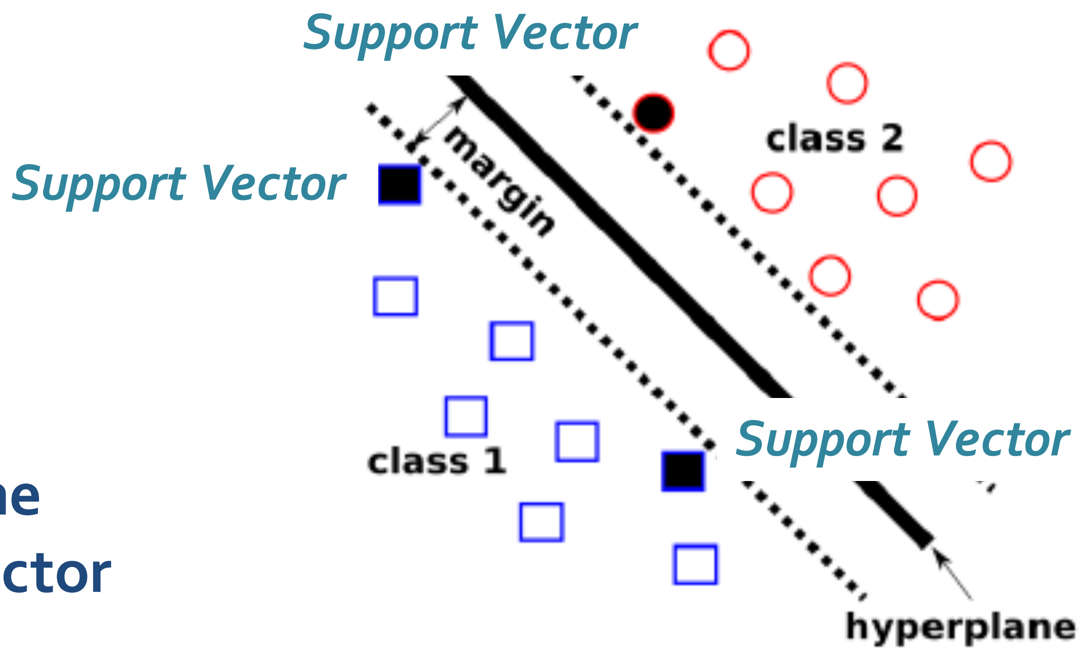
> Find $\alpha_1 \ldots \alpha_N$ such that
> $Q(\alpha) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(x_i, x_j)$ is maximized and
> (1) $\Sigma\alpha_i y_i = 0$
> (2) $\alpha_i \geq 0$ for all α_i

- The solution is:

$$f(x) = \Sigma\alpha_i y_i K(x_i, x_j) + b$$

- Optimization techniques for finding α_i's remain the same!

Concepts



Quick start:

1. Hyperplane
2. Supper Vector
3. Margin
4. SVMs
5. Maximize Margin Width
6. Non-linear SVMs: Kernel Function

SVM: History and Applications

- Vapnik and colleagues (**1992**)—groundwork from Vapnik & Chervonenkis' *statistical learning theory* in **1960s**
- Features: **training can be slow but accuracy is high** owing to their ability to model **complex nonlinear decision boundaries** (margin maximization)
- Applications:
  - Text categorization
  - Image classification
  - Hand-written digit/character recognition
  - Object recognition
  - Bioinformatics (Protein classification, Cancer classification…)

SVM Related Links

- SVM Website: http://www.kernel-machines.org/
- Representative implementations
  - **LIBSVM**: an efficient implementation of SVM, multi-class classifications, nu-SVM, one-class SVM, including also various interfaces with java, python, etc.
  - **SVM-light**: simpler but performance is not better than LIBSVM, support only binary classification and only in C
  - **SVM-torch**: another recent implementation also written in C
  - http://www.meng-jiang.com/teaching/SVMDemo.zip

References

- C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, 1995
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2): 121-168, 1998
- N. Cristianini and J. Shawe-Taylor, Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000
- H. Yu, J. Yang, and J. Han. Classifying large data sets using SVM with hierarchical clusters. KDD'03
- A. J. Dobson. An Introduction to Generalized Linear Models. Chapman & Hall, 1990
- R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, 2ed. John Wiley, 2001
- T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2001
- S. Haykin, Neural Networks and Learning Machines, Prentice Hall, 2008
- D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 1995
- H. Cheng, X. Yan, J. Han & C.-W. Hsu, Discriminative Frequent Pattern Analysis for Effective Classification, ICDE'07
- W. Cohen. Fast effective rule induction. ICML'95

References (cont.)

- H. Cheng, X. Yan, J. Han & P. S. Yu, Direct Discriminative Pattern Mining for Effective Classification, ICDE'08
- G. Cong, K. Tan, A. Tung & X. Xu. Mining Top-k Covering Rule Groups for Gene Expression Data, SIGMOD'05
- M. Deshpande, M. Kuramochi, N. Wale & G. Karypis. Frequent Substructure-based Approaches for Classifying Chemical Compounds, TKDE'05
- G. Dong & J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences, KDD'99
- W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. S. Yu & O. Verscheure. Direct Mining of Discriminative and Essential Graphical and Itemset Features via Model-based Search Tree, KDD'08
- W. Li, J. Han & J. Pei. CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules, ICDM'01
- B. Liu, W. Hsu & Y. Ma. Integrating Classification and Association Rule Mining, KDD'98
- J. R. Quinlan and R. M. Cameron-Jones. FOIL: A midterm report. ECML'93
- Jingbo Shang, Wenzhu Tong, Jian Peng, and Jiawei Han, "DPClass: An Effective but Concise Discriminative Patterns-Based Classification Framework", SDM'16
- J. Wang and G. Karypis. HARMONY: Efficiently Mining the Best Rules for Classification, SDM'05
- X. Yin & J. Han. CPAR: Classification Based on Predictive Association Rules, SDM'03