Support Vector Machines

Machine Learning Spring 2018

March 5 2018

Kasthuri Kannan

kasthuri.kannan@nyumc.org

Overview

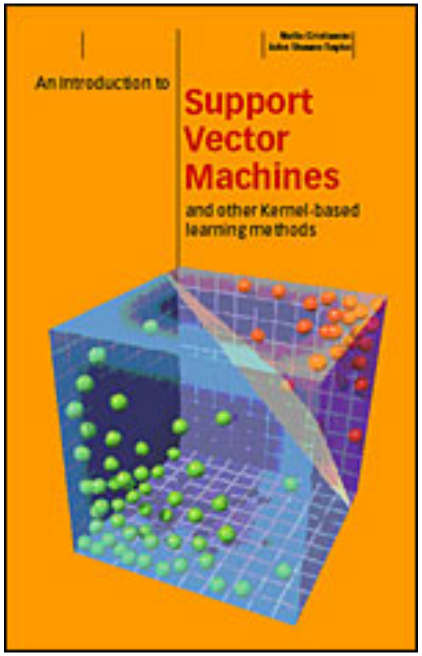
- Support Vector Machines for Classification
  - Linear Discrimination
  - Nonlinear Discrimination
- SVM Mathematically
- Extensions
- Application in Drug Design
- Data Classification
- Kernel Functions

Definition

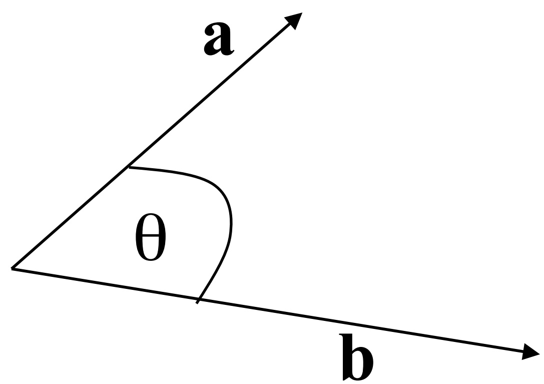
One of the excellent classification system based on a mathematical technique called convex optimization.

‘Support Vector Machine is a system for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalisation theory and exploiting optimisation theory.’

- AN INTRODUCTION TO SUPPORT VECTOR MACHINES (and other kernel-based learning methods)
  - N. Cristianini and J. Shawe-Taylor, Cambridge University Press 2000 ISBN: 0 521 78019 5
- Kernel Methods for Pattern Analysis
  - John Shawe-Taylor & Nello Cristianini Cambridge University Press, 2004



Dot product (aka inner product)



$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos\theta$$

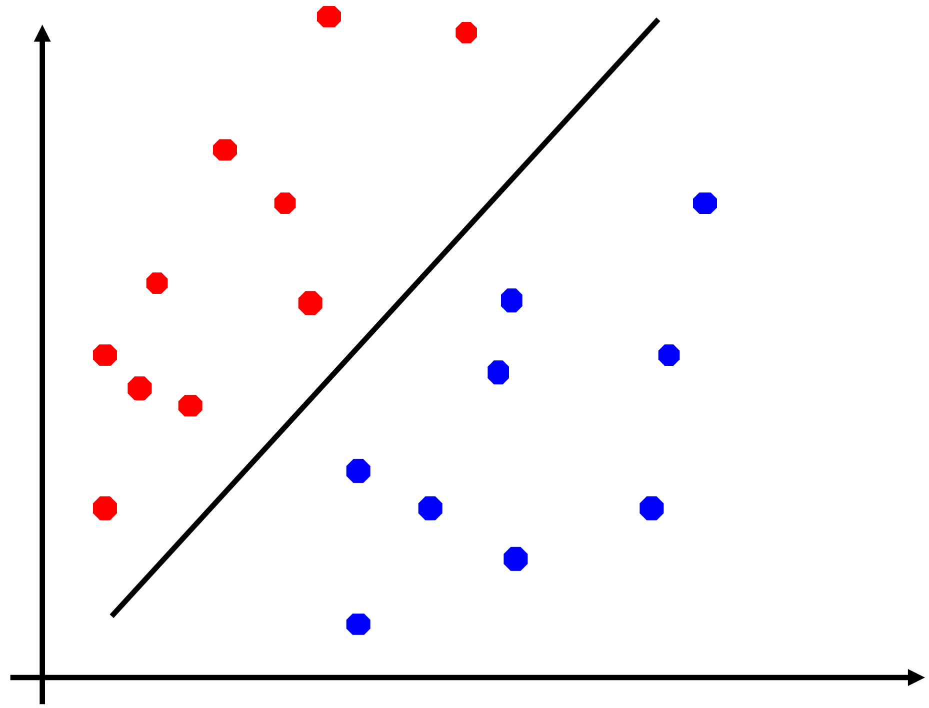
Recall: If the vectors are orthogonal, dot product is zero.

The scalar or dot product is, in some sense, a measure of **similarity**

Decision function for binary classification

$$f(x) \in \mathbf{R}$$

$$f(x_i) \geq 0 \Rightarrow y_i = 1$$
$$f(x_i) < 0 \Rightarrow y_i = -1$$



Support vector machines

- SVMs pick best separating hyper plane according to some criterion
  - e.g. maximum margin
- Training process is an optimisation
- Training set is effectively reduced to a relatively small number of support vectors
- Key words: optimization, kernels

Feature spaces

- We may separate data by mapping to a higher-dimensional feature space
  - The feature space may even have an infinite number of dimensions!

- We need not explicitly construct the new feature space
  - “Kernel trick”
  - Keeps the same computation time

- Key observation that optimization involves dot products

Kernels

- What are kernels?

$$(Tf)(u) = \int_{t_1}^{t_2} K(t,u)\, f(t)\, dt$$

- We may use Kernel functions to implicitly map to a new feature space

- Kernel functions: $K(\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{R}$

- In SVMs kernels preserve the inner product in the new feature space.

Examples of kernels

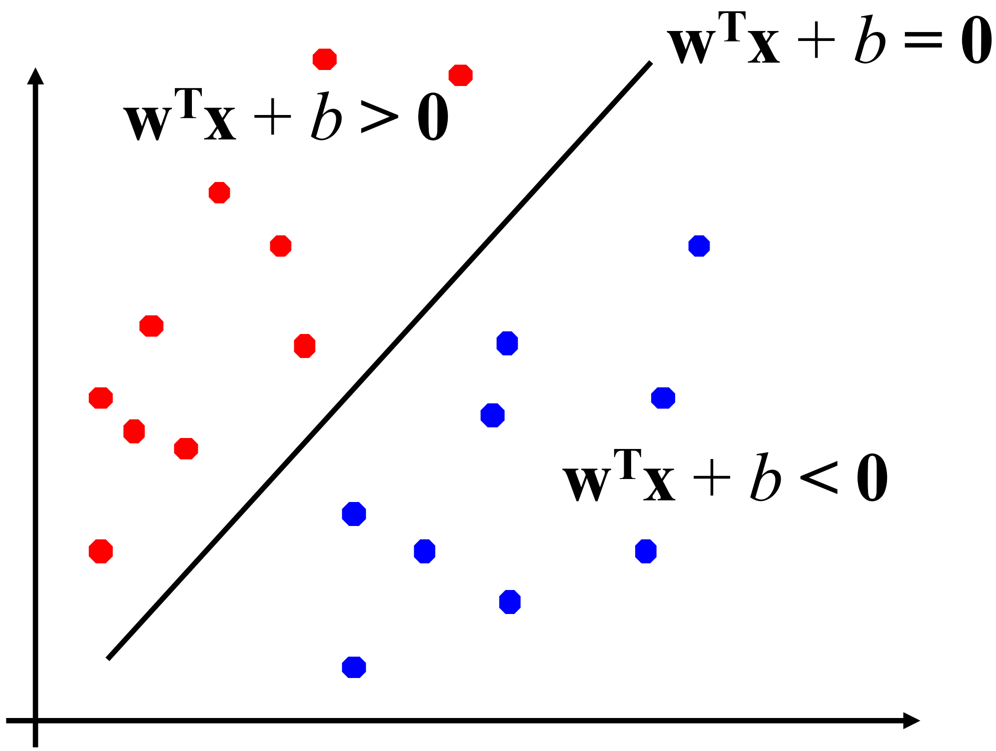
Linear: $\langle \mathbf{x} \cdot \mathbf{z} \rangle$

Polynomial (non-linear) $P(\langle \mathbf{x} \cdot \mathbf{z} \rangle)$

Gaussian (non-linear) $\exp\left(-\|\mathbf{x} - \mathbf{z}\|^2 / \sigma^2\right)$

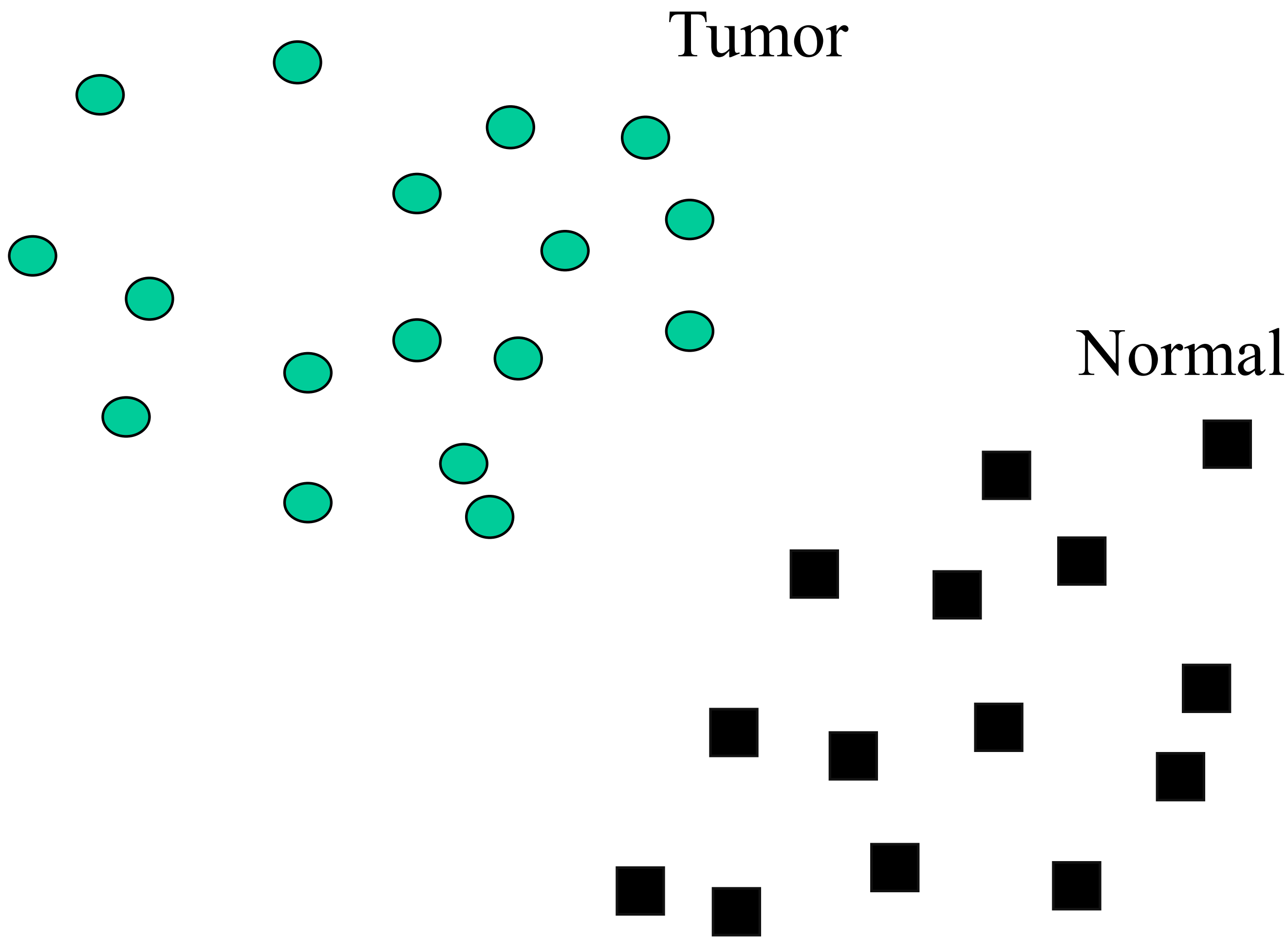
Perceptron as linear separator

- Binary classification can be viewed as the task of separating classes in feature space:

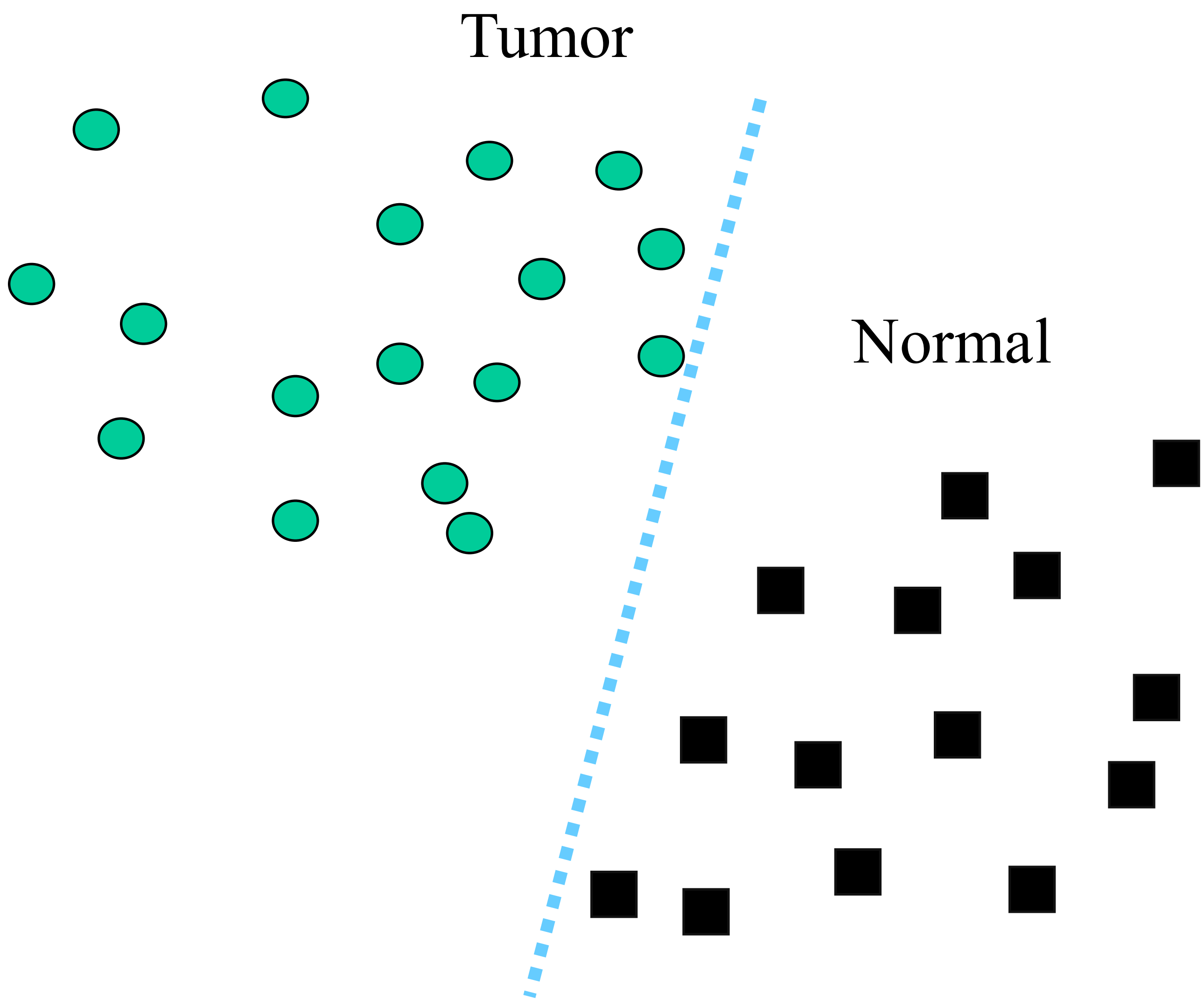


$$f(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$$

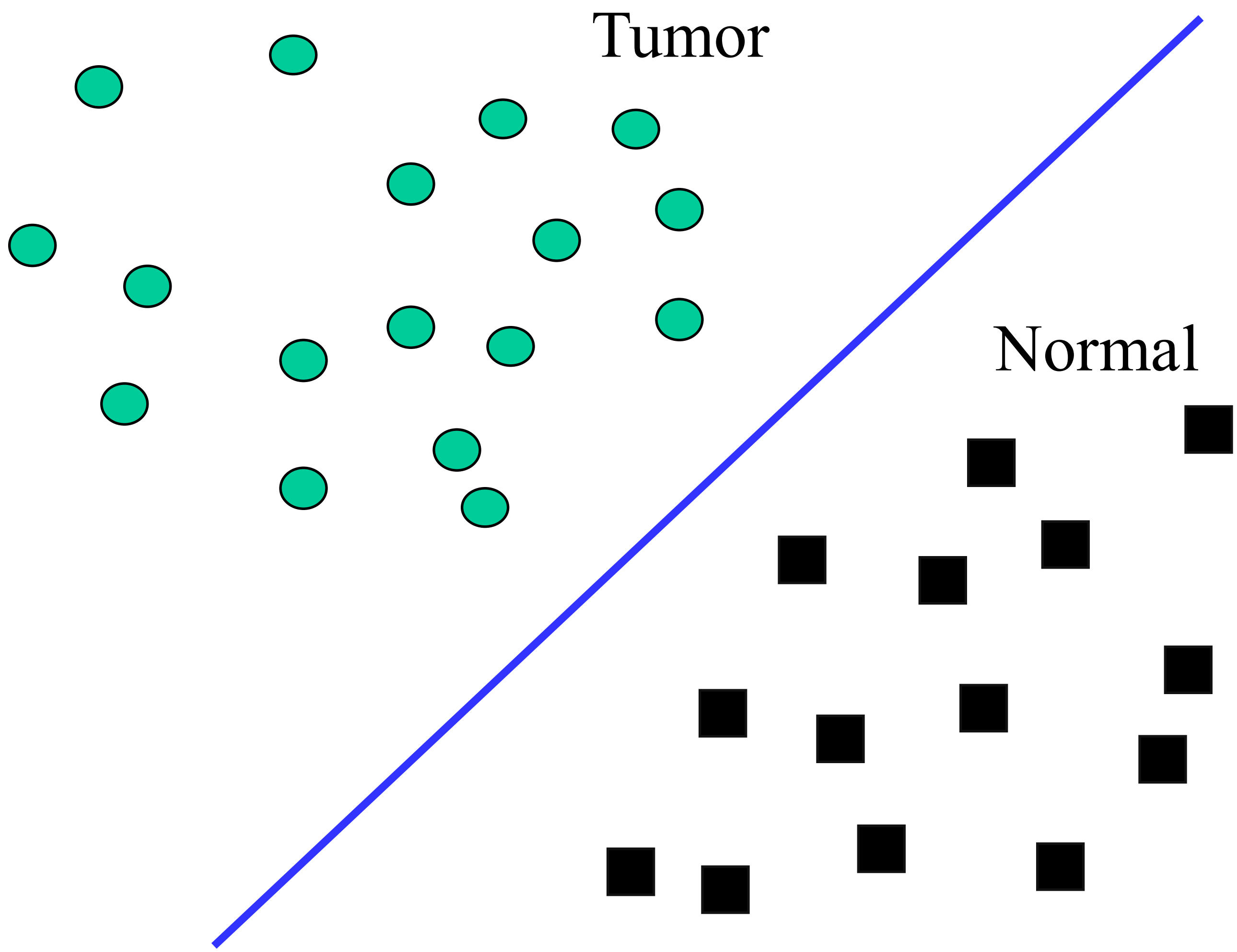
Which of the linear separators is optimal?



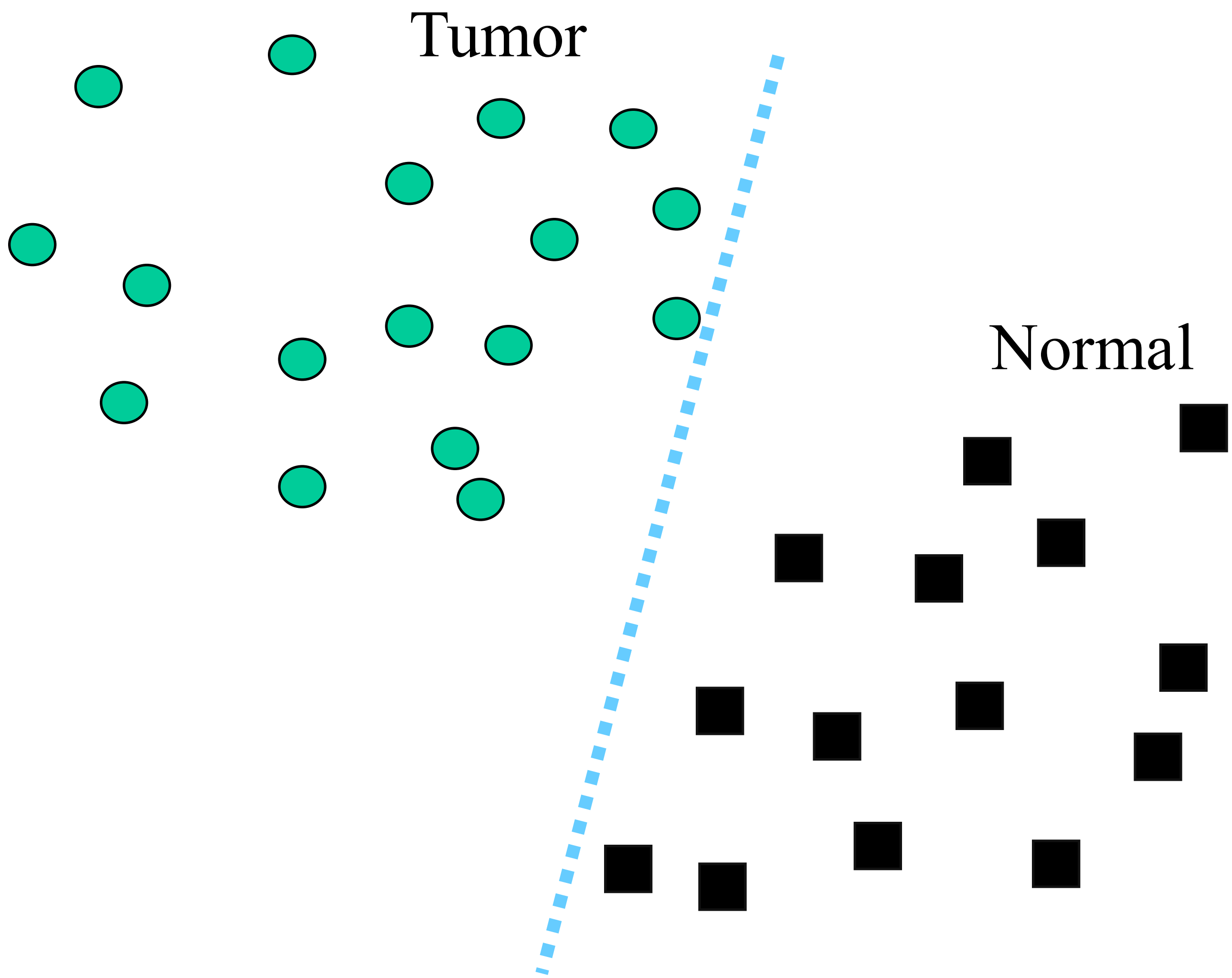
Best linear separator?



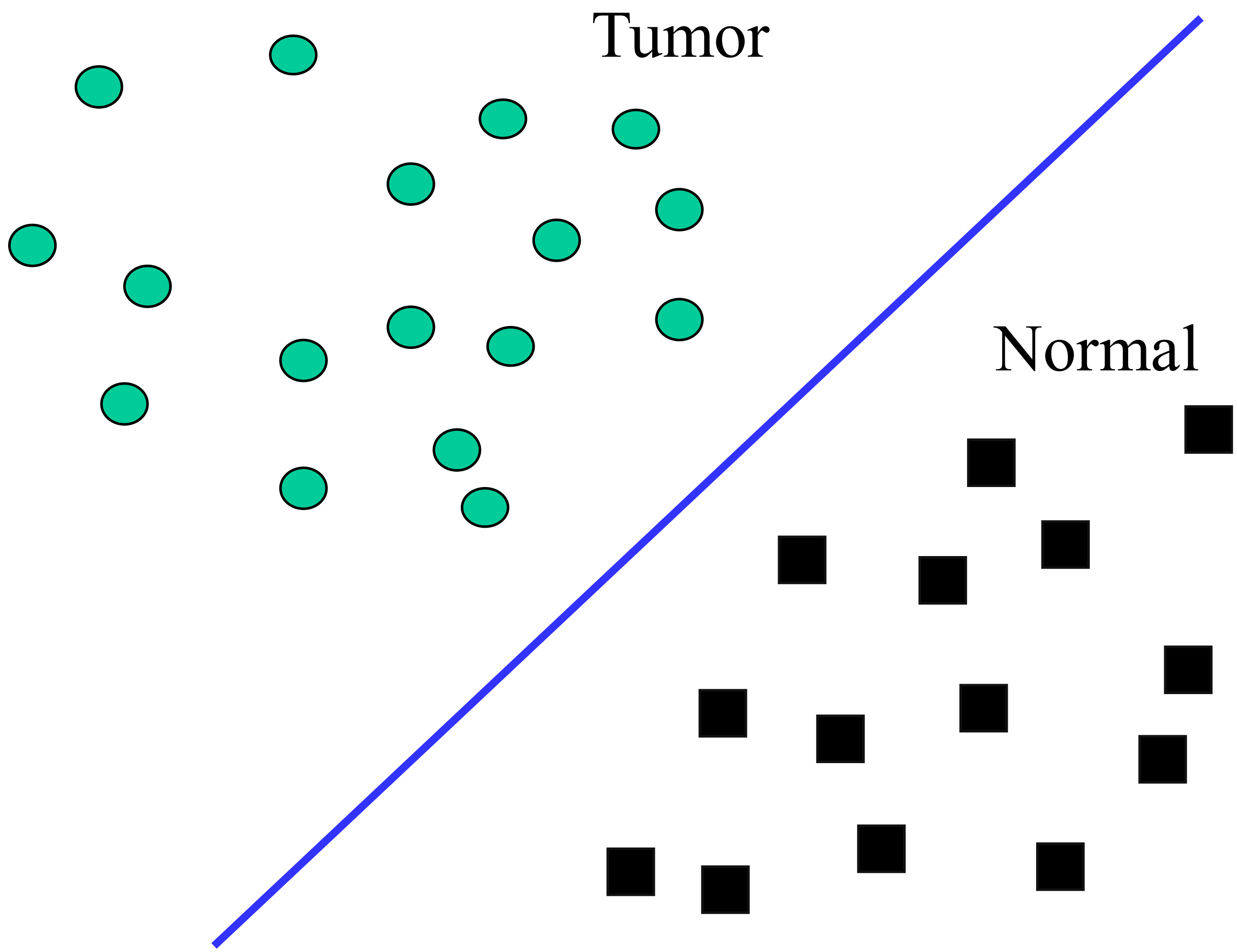
Best linear separator?



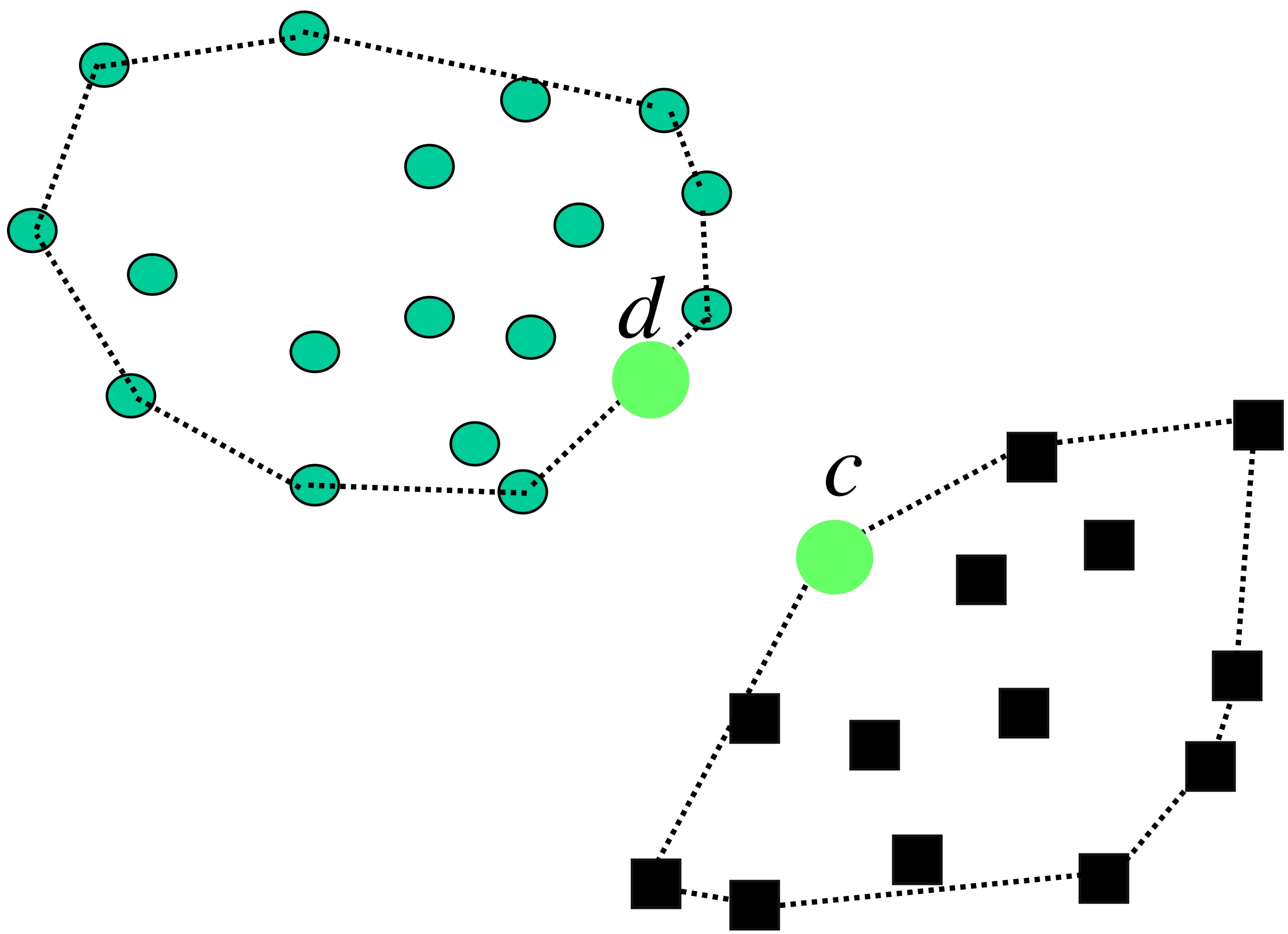
Best linear separator? Not so…



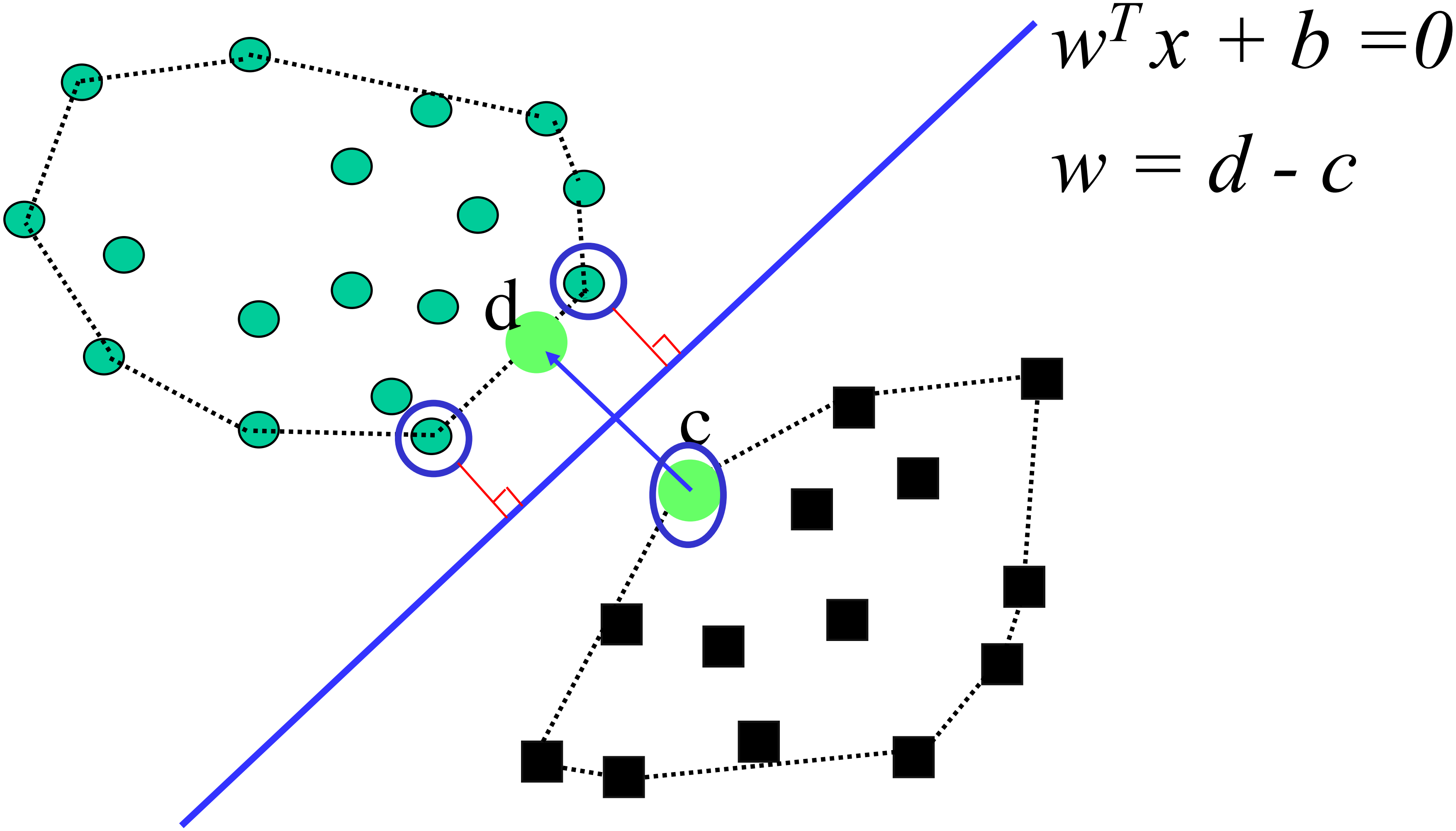
Best linear separator? Possibly…



Find closest points in convex hulls (3D)/convex polygon (2D)



Plane (3D)/line(2D) to bisect closest points

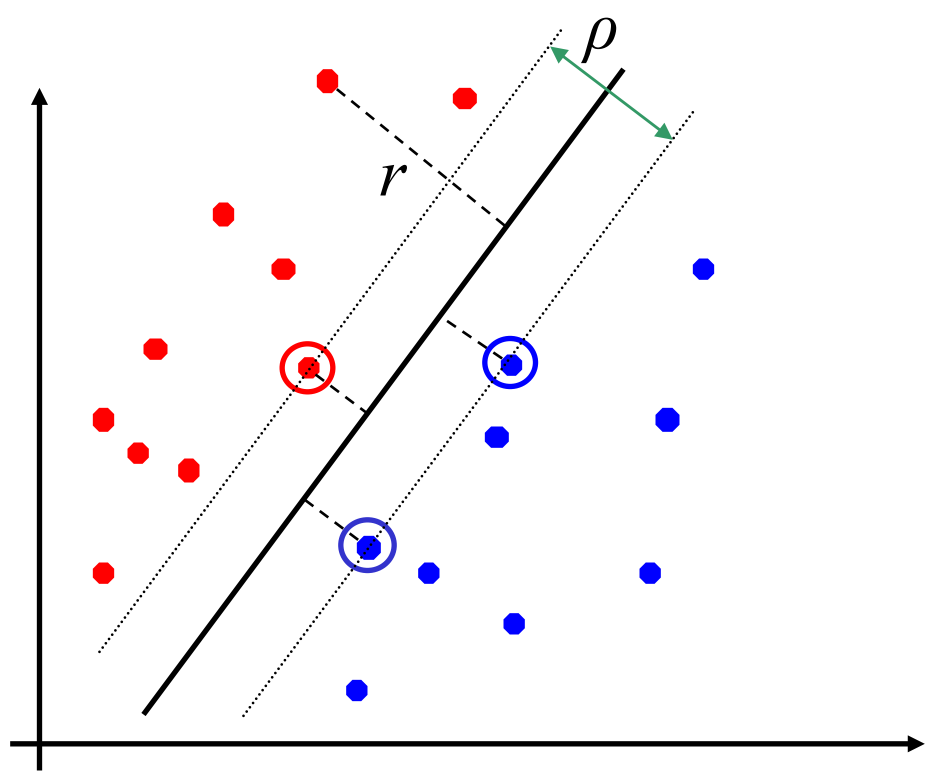


Classification margin

- Distance from example data to the separator is $r = \dfrac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$

- Data closest to the hyper plane are ***support vectors***.

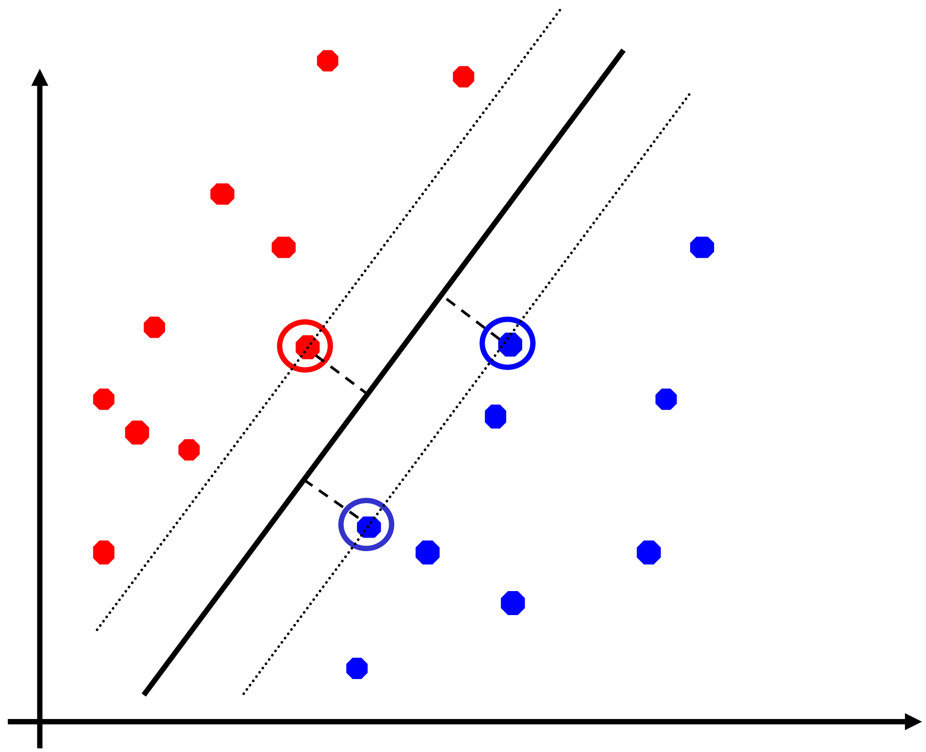
- ***Margin*** ρ of the separator is the width of separation between classes.



Maximum margin classification

- Maximize the margin (good according to intuition and theory).

- Implies that only support vectors are important; other training examples are ignorable.



Statistical learning theory

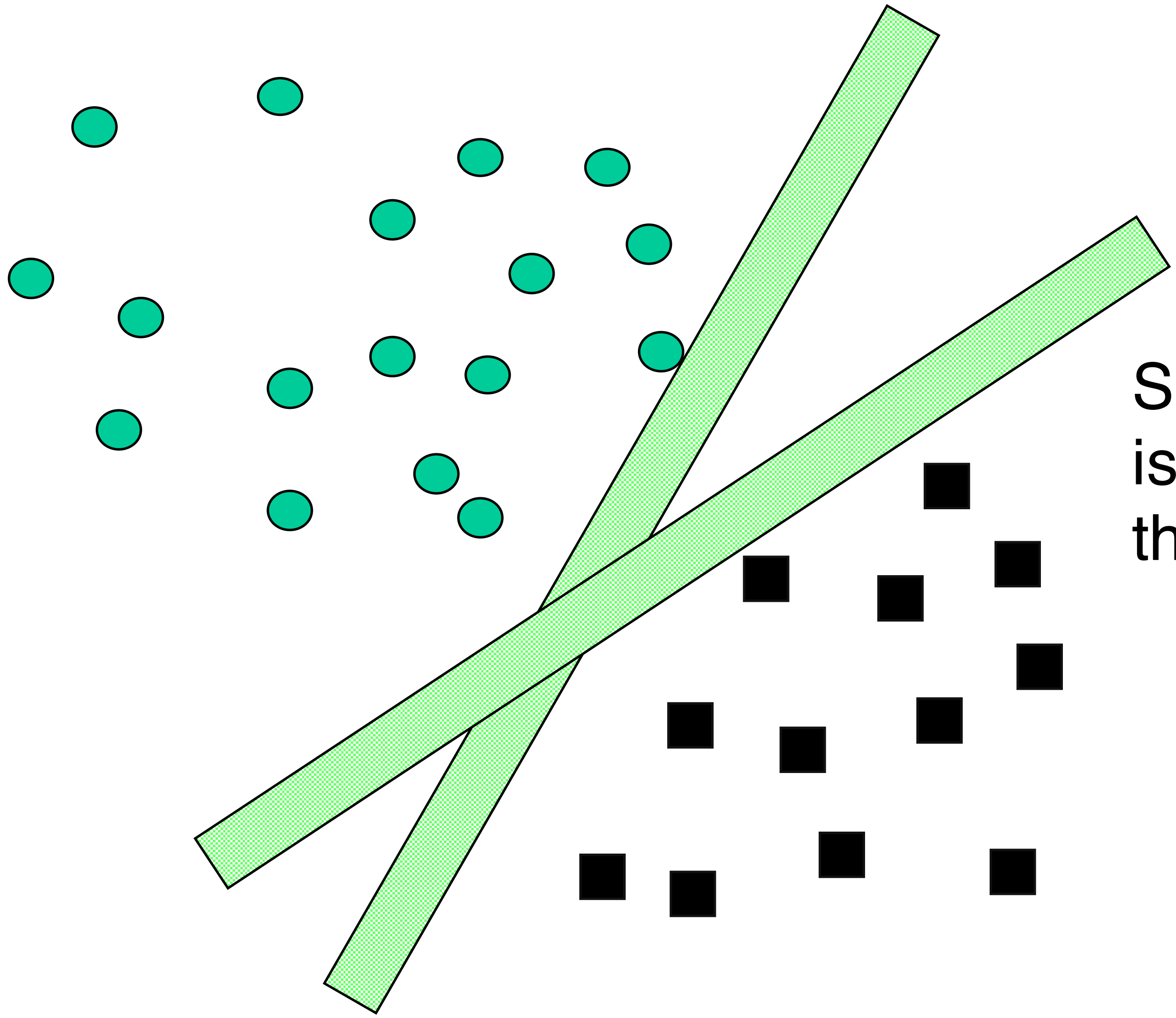
- Misclassification error and the function complexity bound generalization error (prediction).

- Maximizing margins minimizes complexity.

- “Eliminates” overfitting.

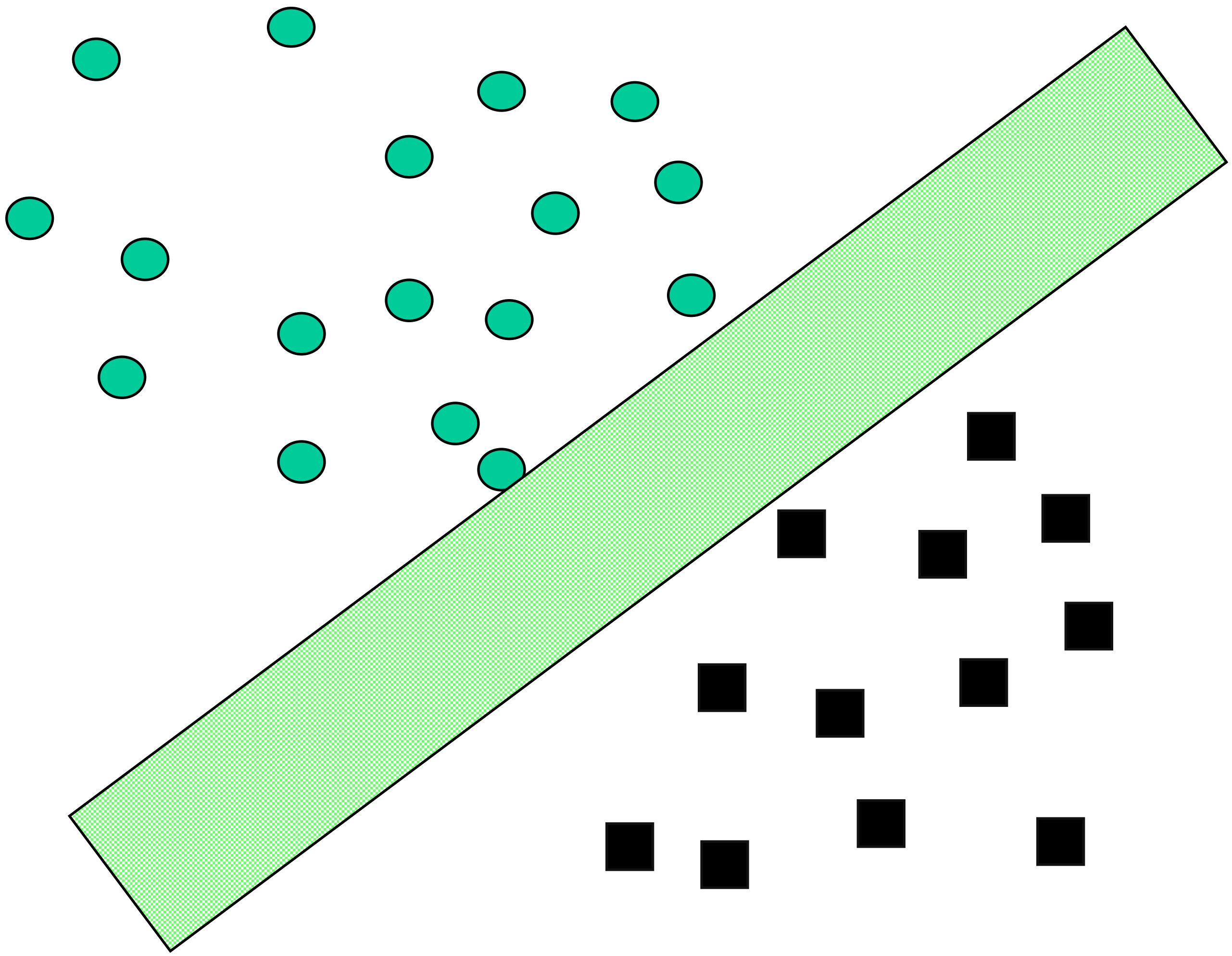
- Solution depends only on support vectors not number of attributes.

Margins and complexity



Skinny margin
is more flexible
thus more complex.

Margins and complexity



Fat margin
is less complex.

Linear SVM

- Assuming all data is at distance larger than 1 from the hyperplane, the following two constraints follow for a training set $\{(\mathbf{x_i}, y_i)\}$

$$\mathbf{w^T x_i} + b \geq 1 \quad \text{if } y_i = 1$$

$$\mathbf{w^T x_i} + b \leq -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality; then, since each example's distance from the
- hyperplane is $r = \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|}$ the margin is: $\rho = \frac{2}{\|\mathbf{w}\|}$

Linear SVM

We can formulate the problem:

Find $\mathbf{w}$ and b such that

$$\rho = \frac{2}{\|\mathbf{w}\|} \text{ is maximized and for all } \{(\mathbf{x_i}, y_i)\}$$

$$\mathbf{w^T x_i} + b \geq 1 \text{ if } y_i = 1; \quad \mathbf{w^T x_i} + b \leq -1 \quad \text{if } y_i = -1$$

into quadratic optimization formulation:

Find $\mathbf{w}$ and b such that

$$\mathbf{\Phi}(\mathbf{w}) = \tfrac{1}{2}\, \mathbf{w}^{\mathrm{T}}\mathbf{w} \text{ is minimized and for all } \{(\mathbf{x_i}, y_i)\}$$

$$y_i\,(\mathbf{w^T x_i} + b) \geq 1$$

Solving the optimization problem

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i\, (\mathbf{w^T x_i} + b) \geq 1$

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:

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

The quadratic optimization problem solution

- The solution has the form:

$$\boxed{\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w^T x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0}$$

- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.
- Then the classifying function will have the form:

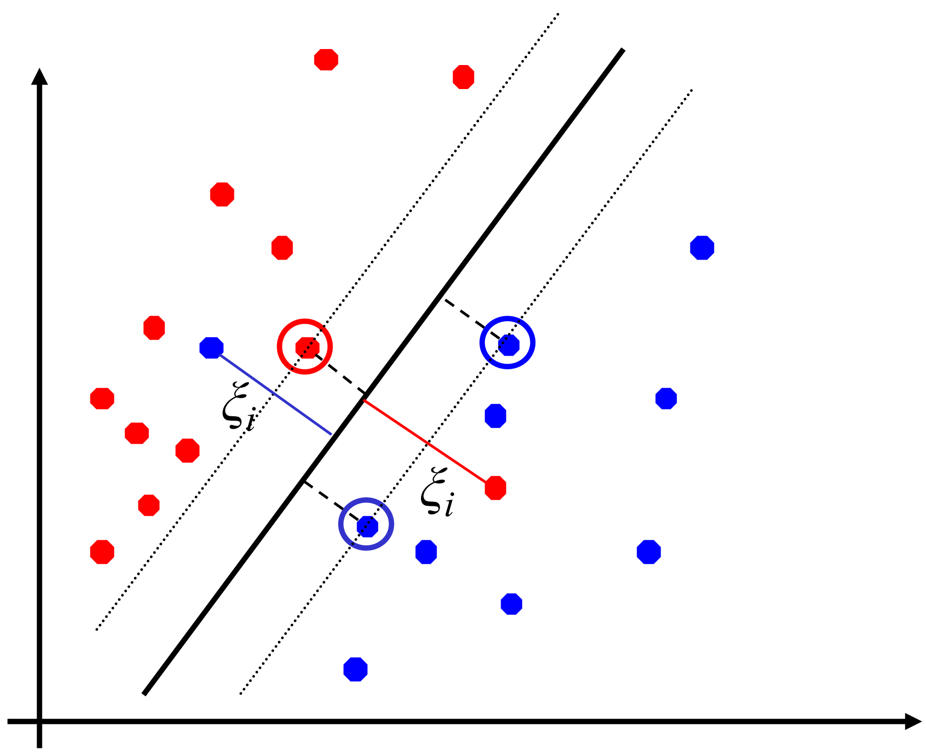
$$\boxed{f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^\mathbf{T}\mathbf{x} + b}$$

- Notice that it relies on an **inner product** between the test point $\mathbf{x}$ and the support vectors $\mathbf{x_i}$ – we will return to this later!

- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x_i}^\mathbf{T}\mathbf{x_j}$ between all training points!

Soft margin classification

- What if the training set is not linearly separable?
- Slack variables ξ_i can be added to allow misclassification of difficult or noisy examples.



Soft margin classification

- The old formulation:

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}\, \mathbf{w}^{\mathrm{T}}\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i\, (\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + \mathrm{b}) \geq 1$

- The new formulation incorporating slack variables:

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}\, \mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum\xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i\, (\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

- Parameter C can be viewed as a way to control overfitting.

Soft margin classification – solution

- The dual problem for soft margin classification:

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

- Neither slack variables ξ_i nor their Lagrange multipliers appear in the dual problem!

- Again, $\mathbf{x_i}$ with non-zero α_i will be support vectors.

- Solution to the dual problem is:

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$
$b = y_k(1 - \xi_k) - \mathbf{w}^T\mathbf{x_k}$ where $k = \underset{k}{\operatorname{argmax}}\ \alpha_k$

But neither **w** nor b are needed explicitly for classification!

$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$

Theoretical justification for maximum margins

- Vapnik has proved the following:

  The class of optimal linear separators has VC dimension h bounded from above as

  $$h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$$

  where ρ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples, and m_0 is the dimensionality.

- Intuitively, this implies that regardless of dimensionality m_0 we can minimize the VC dimension by maximizing the margin ρ.

- Thus, complexity of the classifier is kept small regardless of dimensionality.

Linear SVM: Overview

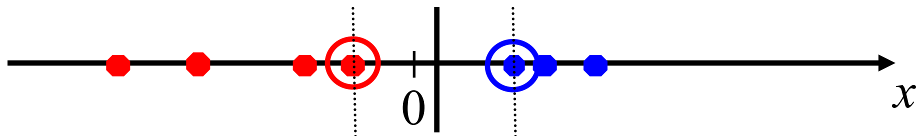
- The classifier is a *separating hyperplane.*
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x_i}$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of the problem and in the solution training points appear only inside inner products:

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

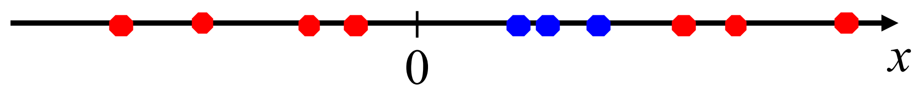
$$f(\mathbf{x}) = \Sigma\alpha_i y_i \boxed{\mathbf{x_i}^T\mathbf{x}} + b$$

Non-linear SVMs

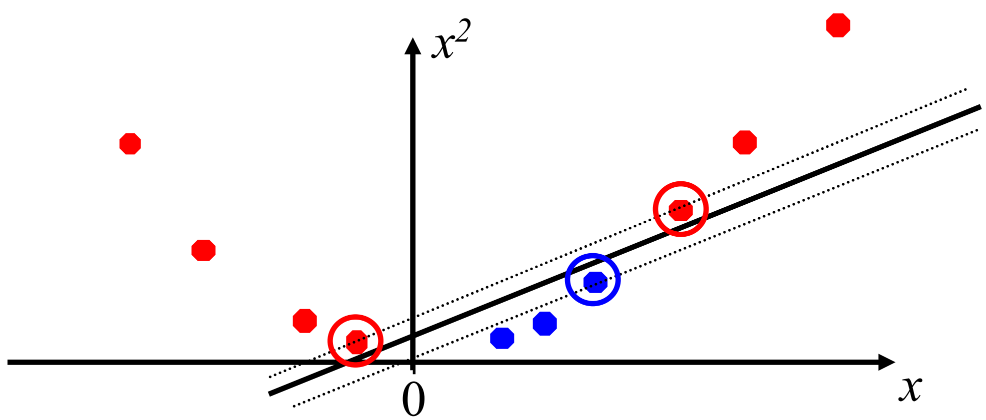
- Datasets that are linearly separable with some noise work out great:



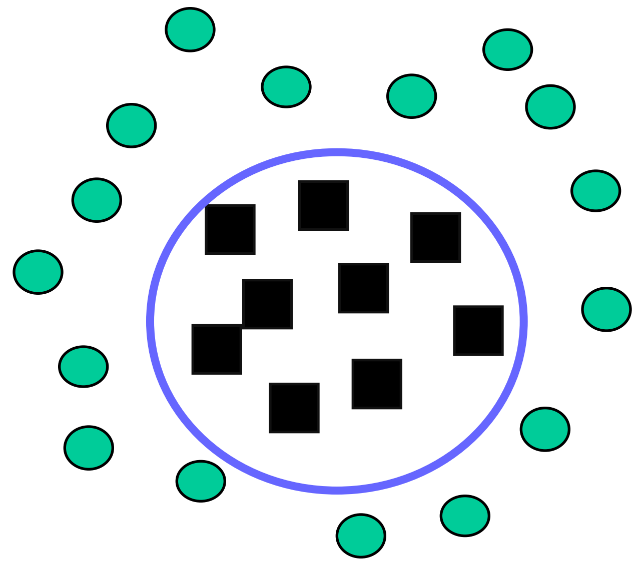
- But what are we going to do if the dataset is just too hard?



- How about… mapping data to a higher-dimensional space:



Nonlinear classification



$$x = [a, b]$$

$$x \bullet w = w_1 a + w_2 b$$

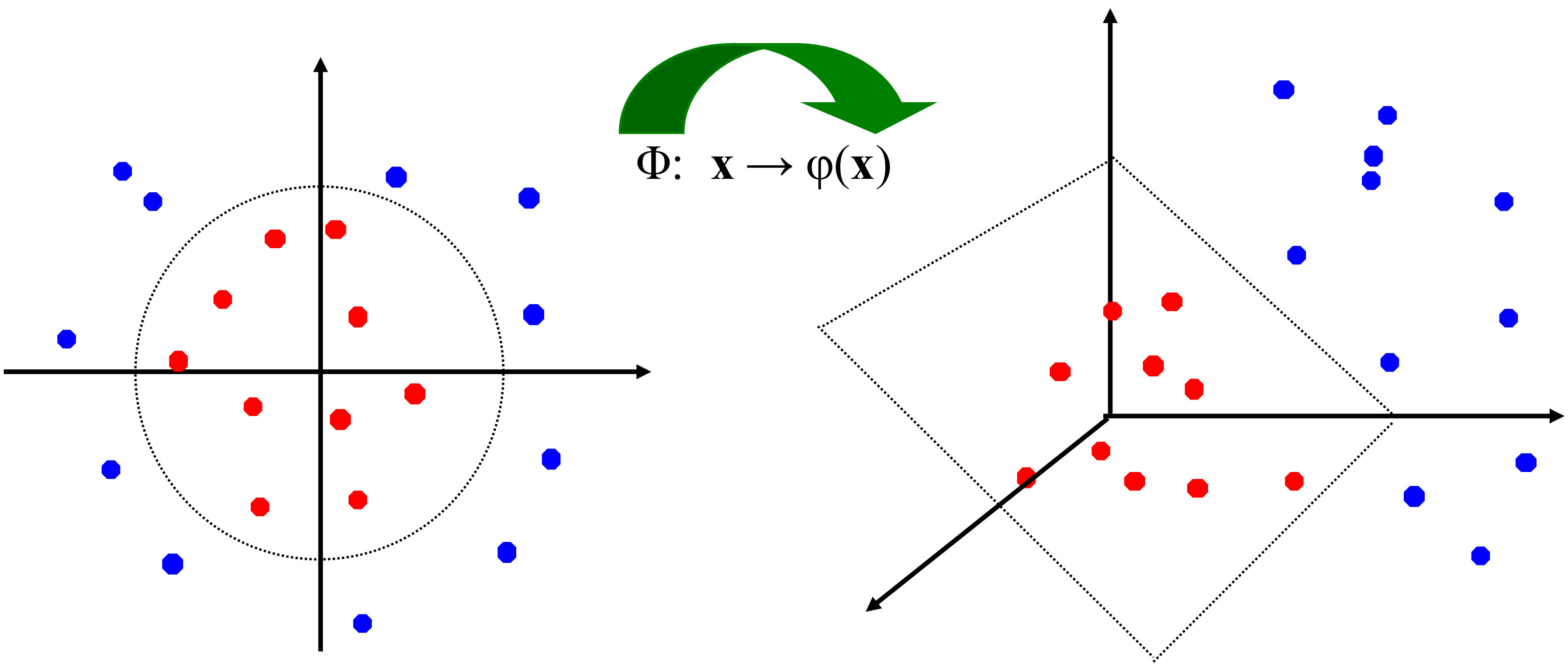
$$\downarrow$$

$$\theta(x) = [a, b, ab, a^2, b^2]$$

$$\theta(x) \bullet w = w_1 a + w_2 b + w_3 ab + w_4 a^2 + w_5 b^2$$

Non-linear SVMs: Feature spaces

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



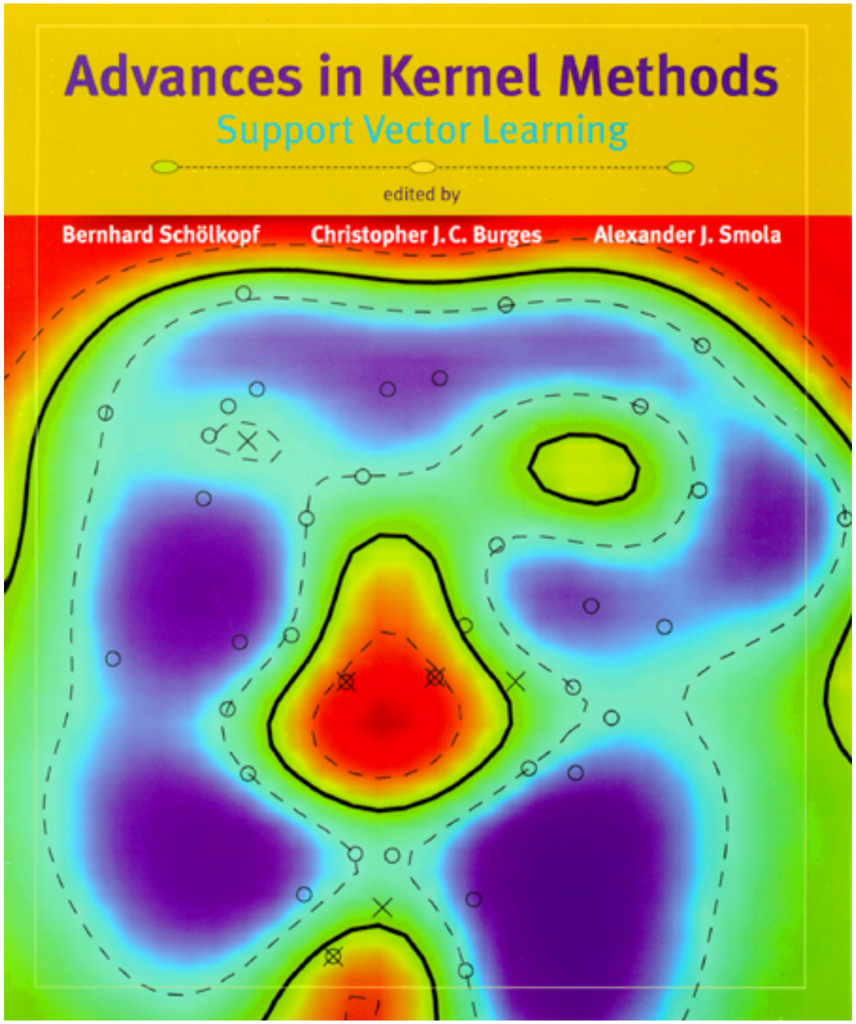
The "Kernel Trick"

- The linear classifier relies on inner product between vectors $K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$
- If every datapoint is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \varphi(\mathbf{x_i})^{\mathbf{T}}\varphi(\mathbf{x_j})$$

- A *kernel function* is some function that corresponds to an inner product into some feature space.
- Example:
  - 2-dimensional vectors $\mathbf{x}=[x_1 \;\; x_2]$; let $K(\mathbf{x_i},\mathbf{x_j})=(1 + \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^2$,
  - Need to show that $K(\mathbf{x_i},\mathbf{x_j}) = \varphi(\mathbf{x_i})^{\mathbf{T}}\varphi(\mathbf{x_j})$:

$$\begin{aligned} K(\mathbf{x_i},\mathbf{x_j}) &= (1 + \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^2 = 1 + x_{i1}^2 x_{j1}^2 + 2\, x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} \\ &= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^{\mathbf{T}} [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}] \\ &= \varphi(\mathbf{x_i})^{\mathbf{T}}\varphi(\mathbf{x_j}), \quad \text{where } \varphi(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2] \end{aligned}$$


Advances in Kernel Methods
Support Vector Learning
edited by
Bernhard Schölkopf
Christopher J.C. Burges
Alexander J. Smola

Positive definite matrices

- A square matrix A is positive definite *if* $x^T A x > 0$ for all nonzero column vectors x.

- It is negative definite if $x^T A x < 0$ for all nonzero x.

- It is positive semi-definite if $x^T A x \geq 0$.

- And negative semi-definite if $x^T A x \leq 0$ for all x.

What functions are kernels?

- For some functions $K(\mathbf{x_i},\mathbf{x_j})$ checking that $K(\mathbf{x_i},\mathbf{x_j}) = \varphi(\mathbf{x_i})^{\mathbf{T}}\varphi(\mathbf{x_j})$ 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=

$K(\mathbf{x_1},\mathbf{x_1})$	$K(\mathbf{x_1},\mathbf{x_2})$	$K(\mathbf{x_1},\mathbf{x_3})$	…	$K(\mathbf{x_1},\mathbf{x_N})$
$K(\mathbf{x_2},\mathbf{x_1})$	$K(\mathbf{x_2},\mathbf{x_2})$	$K(\mathbf{x_2},\mathbf{x_3})$		$K(\mathbf{x_2},\mathbf{x_N})$
…	…	…	…	…
$K(\mathbf{x_N},\mathbf{x_1})$	$K(\mathbf{x_N},\mathbf{x_2})$	$K(\mathbf{x_N},\mathbf{x_3})$	…	$K(\mathbf{x_N},\mathbf{x_N})$

Examples of kernel functions

- Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$
- Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j}) = (1+ \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^p$
- Gaussian (radial-basis function network): $K(\mathbf{x_i},\mathbf{x_j}) = e^{-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{2\sigma^2}}$
- Two-layer perceptron: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0 \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j} + \beta_1)$

Non-linear SVMs - optimization

- Dual problem formulation:

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

- The solution is:

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

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

SVM applications

- SVMs were originally proposed by Boser, Guyon and Vapnik in 1992 and gained increasing popularity in late 1990s.
- SVMs are currently among the best performers for a number of classification tasks ranging from text to genomic data.
- SVM techniques have been extended to a number of tasks such as regression [Vapnik *et al.* '97], principal component analysis [Schölkopf *et al.* '99], etc.
- Most popular optimization algorithms for SVMs are SMO [Platt '99] and SVM^{light} [Joachims' 99], both use *decomposition* to hill-climb over a subset of α_i's at a time.
- Tuning SVMs remains a black art: selecting a specific kernel and parameters is usually done in a try-and-see manner.

SVM extensions

- Regression
- Variable Selection
- Boosting
- Density Estimation
- Unsupervised Learning
  - Novelty/Outlier Detection
  - Feature Detection
  - Clustering

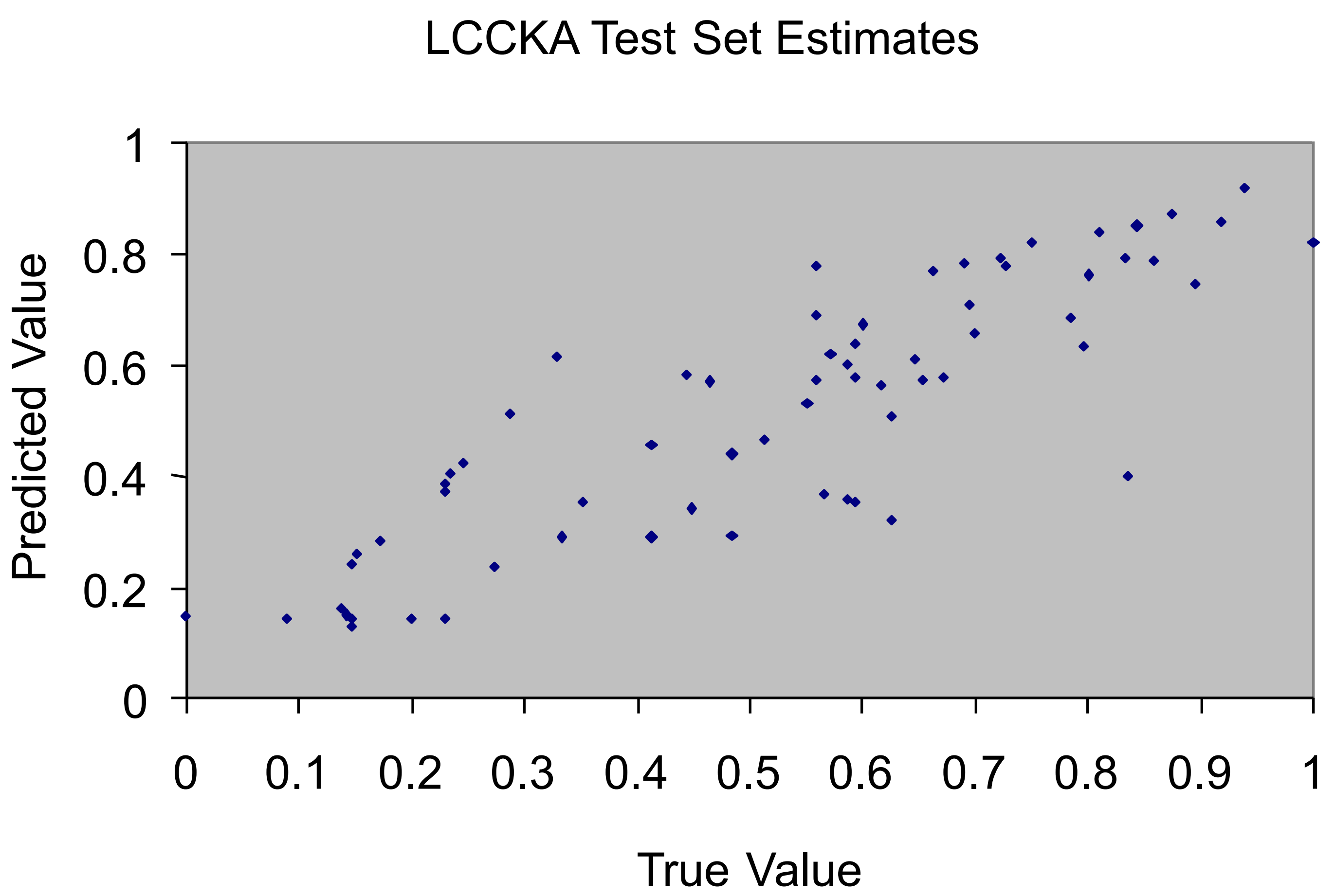
Example in drug design

- Goal to predict bio-reactivity of molecules to decrease drug development time.
- Target is to predict the logarithm of inhibition concentration for site "A" on the Cholecystokinin (CCK) molecule.
- Constructs quantitative structure activity relationship (QSAR) model.

LCCKA problem

- Training data – 66 molecules
- 323 original attributes are wavelet coefficients of TAE Descriptors.
- 39 subset of attributes selected by linear 1-norm SVM (with no kernels).
- For details see DDASSL project link off of http://www.rpi.edu/~bennek
- Testing set results reported.

LCCK prediction



Many other applications

- Speech Recognition
- Data Base Marketing
- Quark Flavors in High Energy Physics
- Dynamic Object Recognition
- Knock Detection in Engines
- Protein Sequence Problem
- Text Categorization
- Breast Cancer Diagnosis
- Cancer Tissue classification
- Translation initiation site recognition in DNA
- Protein fold recognition

One of the best!!

- Generalization theory and practice meet
- General methodology for many types of problems
- Same Program + New Kernel = New method
- No problems with local minima
- Few model parameters. Selects capacity
- Robust optimization methods
- Successful Applications

Open questions

- Will SVMs beat my best hand-tuned method Z for X?
- Do SVM scale to massive datasets?
- How to chose C and Kernel?
- What is the effect of attribute scaling?
- How to handle categorical variables?
- How to incorporate domain knowledge?
- How to interpret results?

Support Vector Machine Resources

- SVM Application List
  http://www.clopinet.com/isabelle/Projects/SVM/applist.html
- Kernel machines
  http://www.kernel-machines.org/
- Pattern Classification and Machine Learning
  http://clopinet.com/isabelle/#projects
- R a GUI language for statistical computing and graphics
  http://www.r-project.org/
- Kernel Methods for Pattern Analysis – 2004
  http://www.kernel-methods.net/
- An Introduction to Support Vector Machines
  (and other kernel-based learning methods)
  http://www.support-vector.net/
- Kristin P. Bennett web page
  http://www.rpi.edu/~bennek
- Isabelle Guyon's home page
  http://clopinet.com/isabelle