A Gentle Introduction to Support Vector Machines in Biomedicine

Alexander Statnikov[*], Douglas Hardin[#], Isabelle Guyon[†], Constantin F. Aliferis[*]

(Materials about SVM Clustering were contributed by **Nikita Lytkin**[*])

[*]New York University, [#]Vanderbilt University, [†]ClopiNet

Part I

- Introduction
- Necessary mathematical concepts
- Support vector machines for binary classification: classical formulation
- Basic principles of statistical machine learning

Introduction

About this tutorial

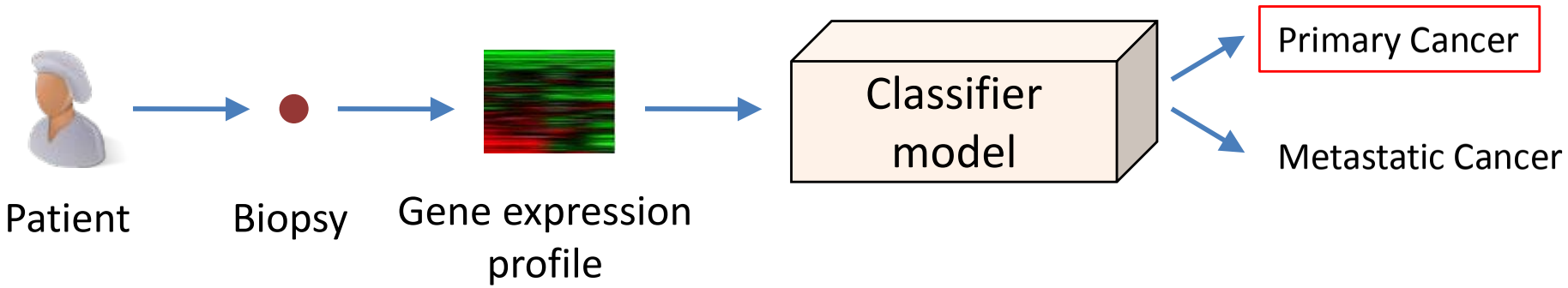
Main goal: Fully understand support vector machines (and important extensions) with a modicum of mathematics knowledge.

- This tutorial is both modest *(it does not invent anything new)* and ambitious *(support vector machines are generally considered mathematically quite difficult to grasp)*.
- Tutorial approach:

  learning problem → main idea of the SVM solution → geometrical interpretation → math/theory → basic algorithms → extensions → case studies.

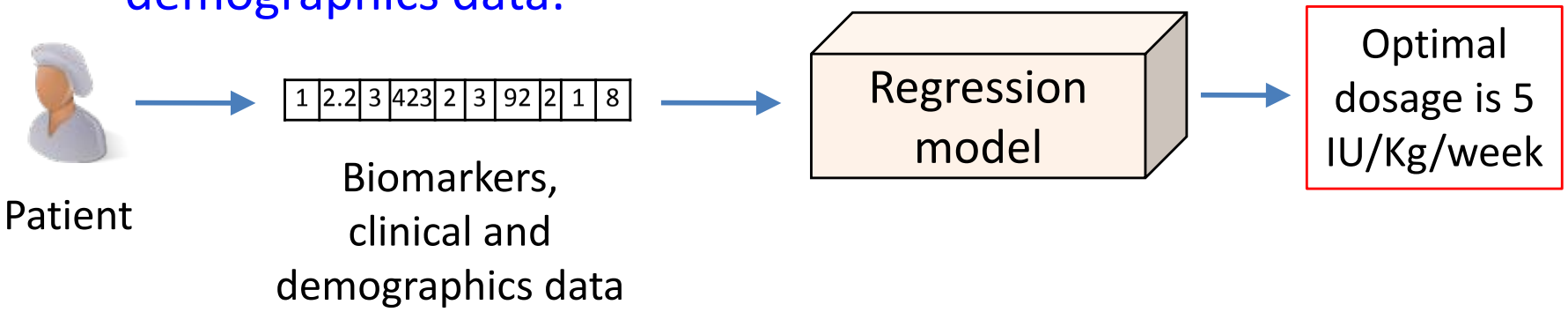
Data-analysis problems of interest

1. Build computational classification models (or "*classifiers*") that assign patients/samples into two or more classes.
   - Classifiers can be used for diagnosis, outcome prediction, and other classification tasks.
   - E.g., build a decision-support system to diagnose primary and metastatic cancers from gene expression profiles of the patients:



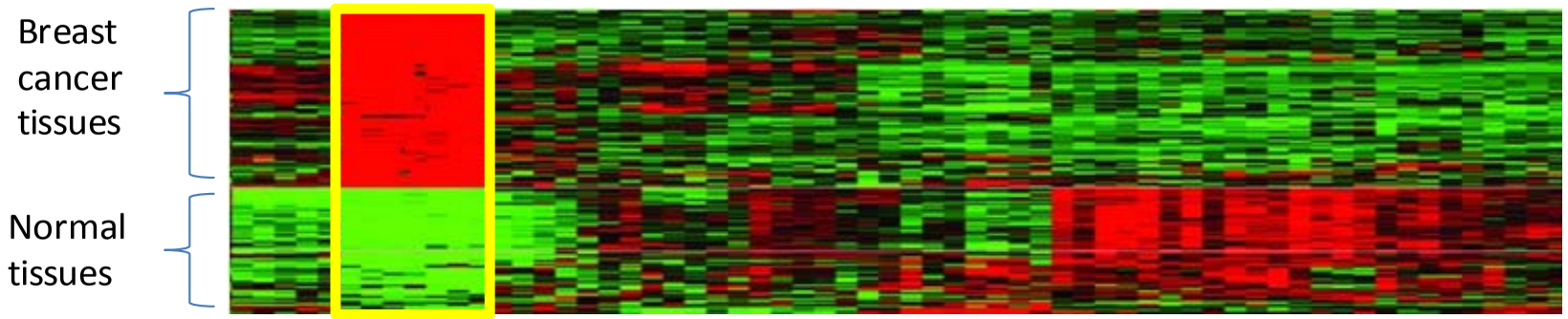
Data-analysis problems of interest

2. Build computational regression models to predict values of some continuous response variable or outcome.
   - Regression models can be used to predict survival, length of stay in the hospital, laboratory test values, etc.
   - E.g., build a decision-support system to predict optimal dosage of the drug to be administered to the patient. This dosage is determined by the values of patient biomarkers, and clinical and demographics data:



Data-analysis problems of interest

3. Out of all measured variables in the dataset, select the smallest subset of variables that is necessary for the most accurate prediction (classification or regression) of some variable of interest (e.g., phenotypic response variable).
   - E.g., find the most compact panel of breast cancer biomarkers from microarray gene expression data for 20,000 genes:

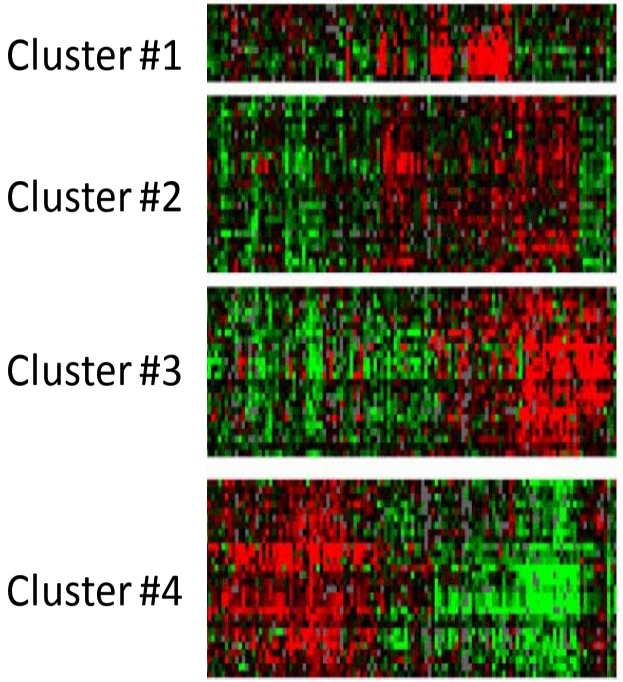


Data-analysis problems of interest

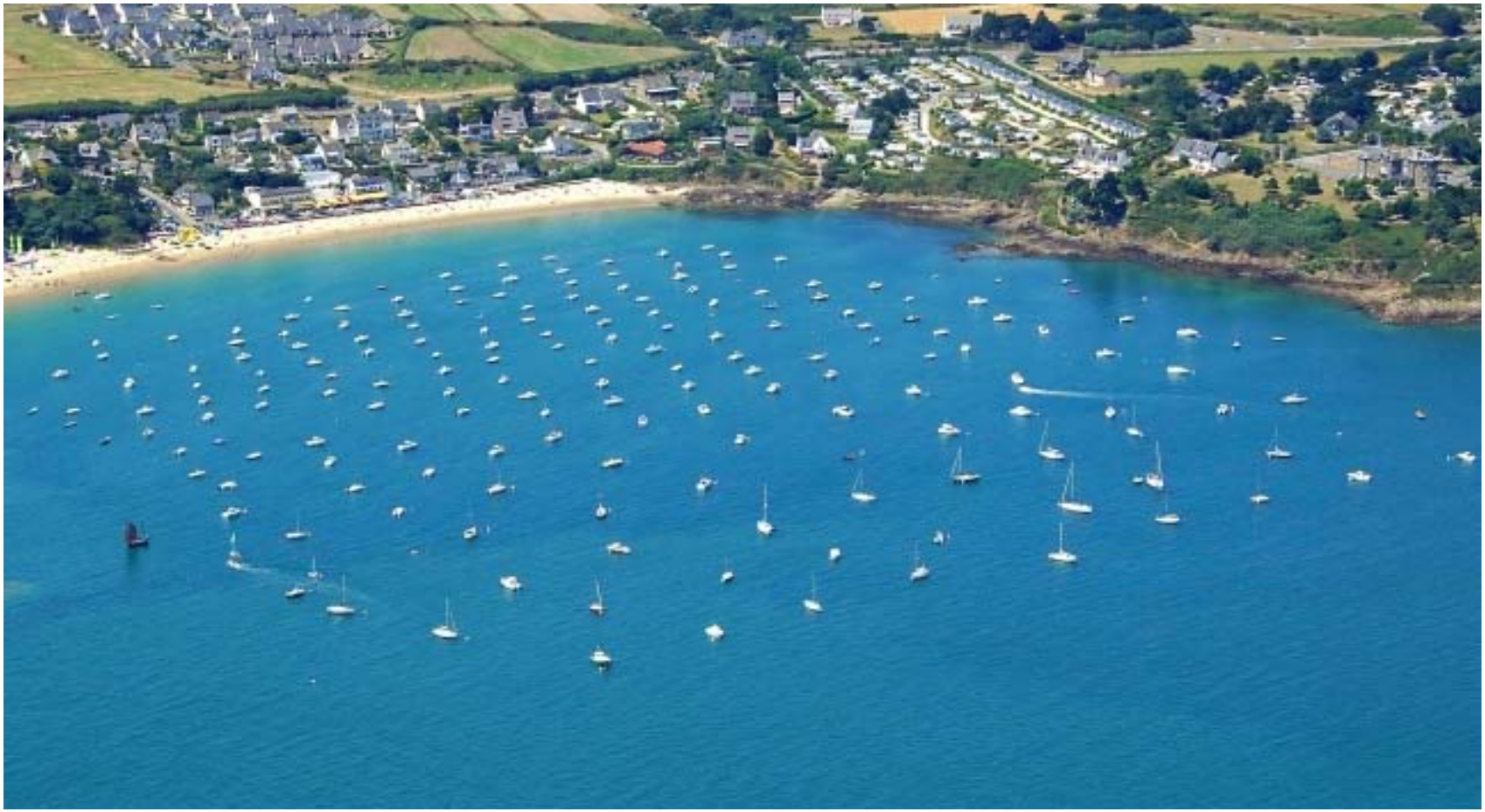
4. Build a computational model to identify novel or outlier patients/samples.
   - Such models can be used to discover deviations in sample handling protocol when doing quality control of assays, etc.
   - E.g., build a decision-support system to identify aliens.

Data-analysis problems of interest

5. Group patients/samples into several clusters based on their similarity.
   - These methods can be used to discovery disease sub-types and for other tasks.
   - E.g., consider clustering of brain tumor patients into 4 clusters based on their gene expression profiles. All patients have the same pathological sub-type of the disease, and clustering discovers new disease subtypes that happen to have different characteristics in terms of patient survival and time to recurrence after treatment.

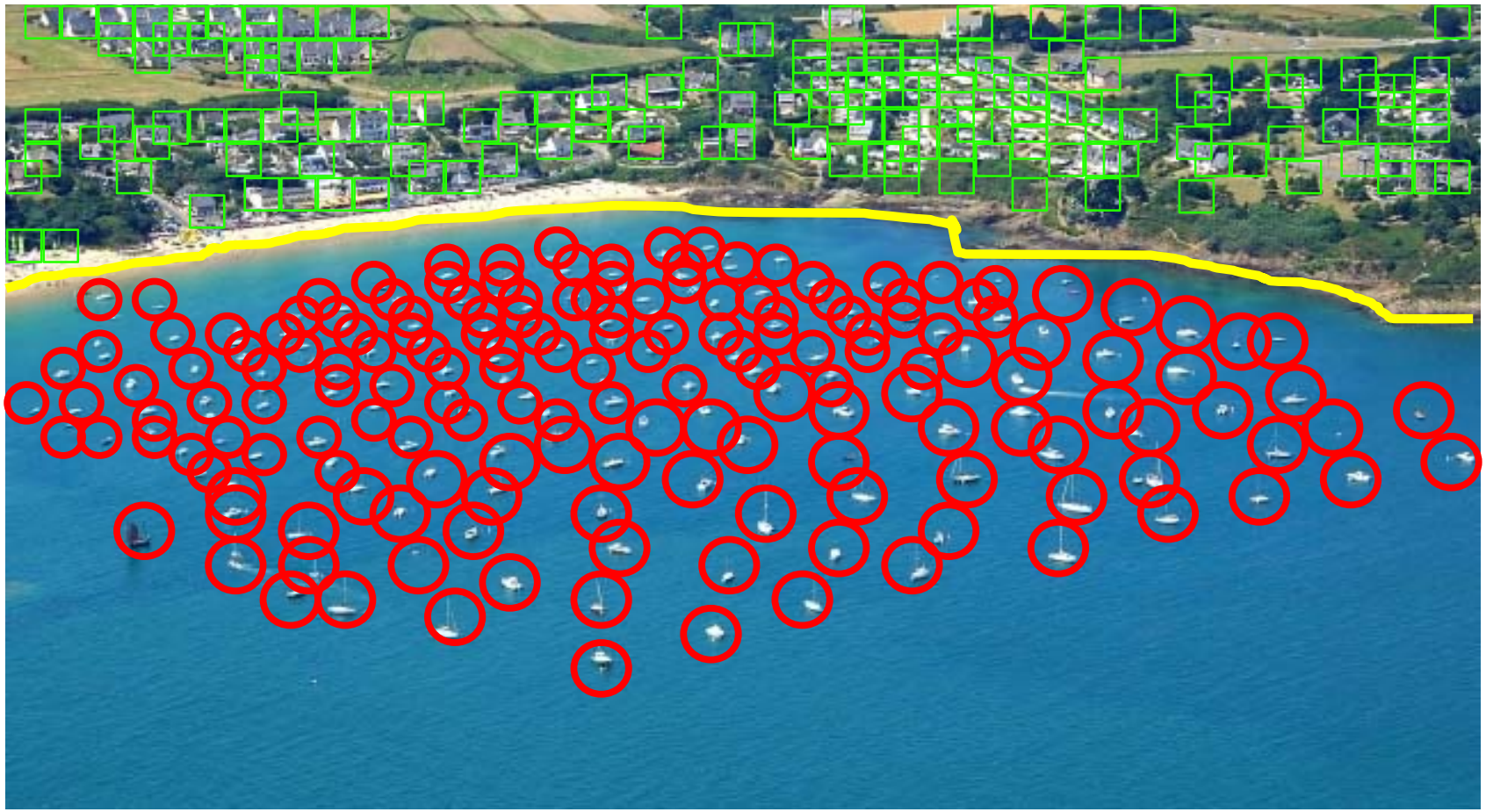


Basic principles of classification



- Want to classify objects as boats and houses.

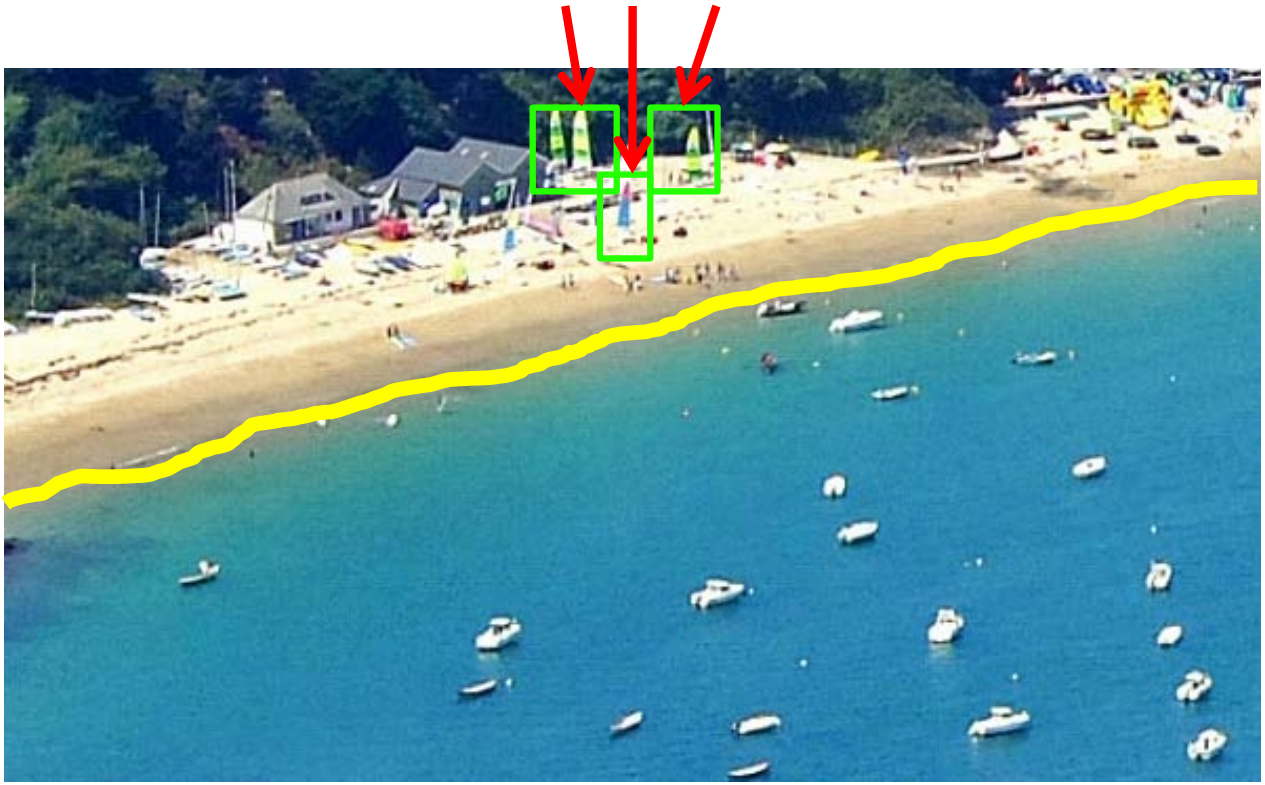
Basic principles of classification



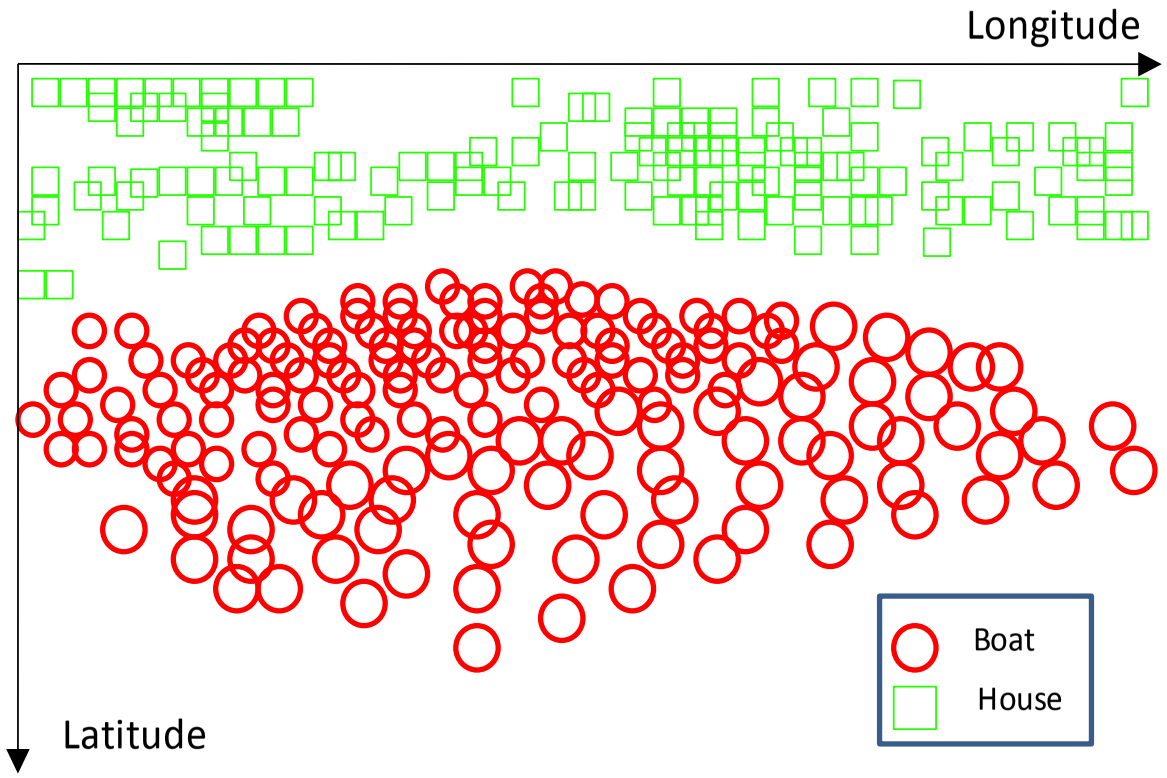
- All objects before the coast line are boats and all objects after the coast line are houses.
- Coast line serves as a *decision surface* that separates two classes.

Basic principles of classification

These boats will be misclassified as houses

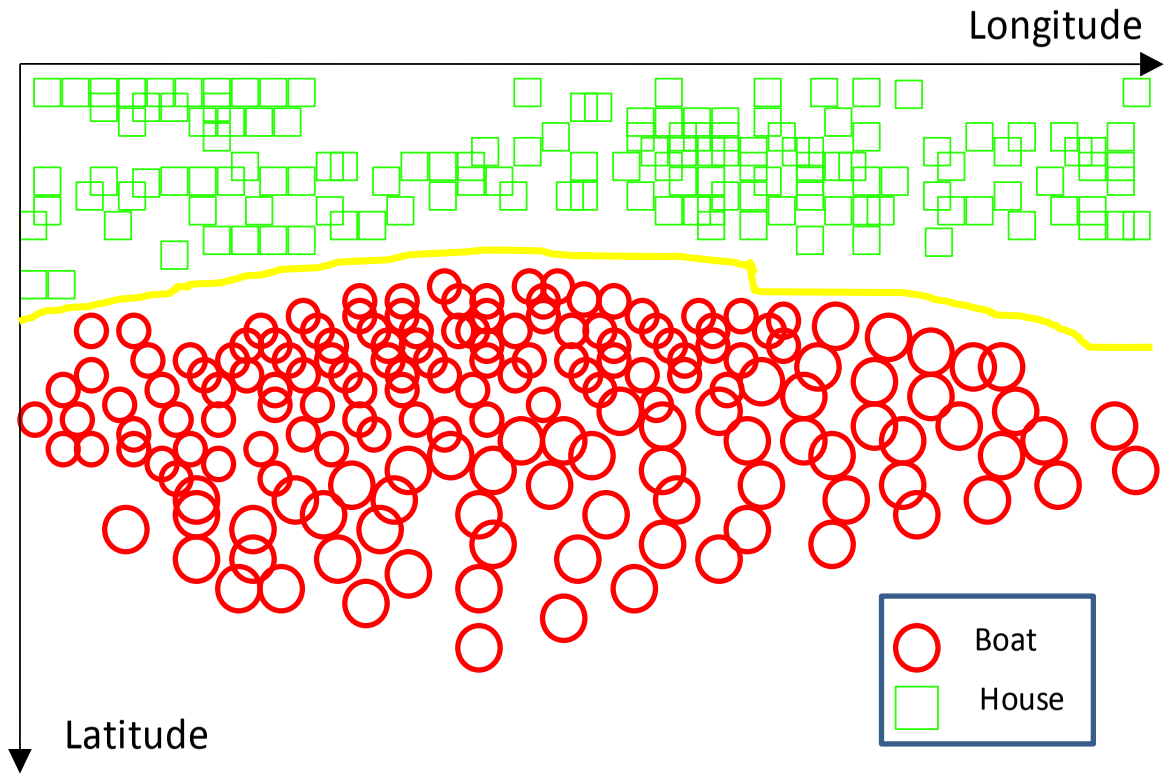


Basic principles of classification



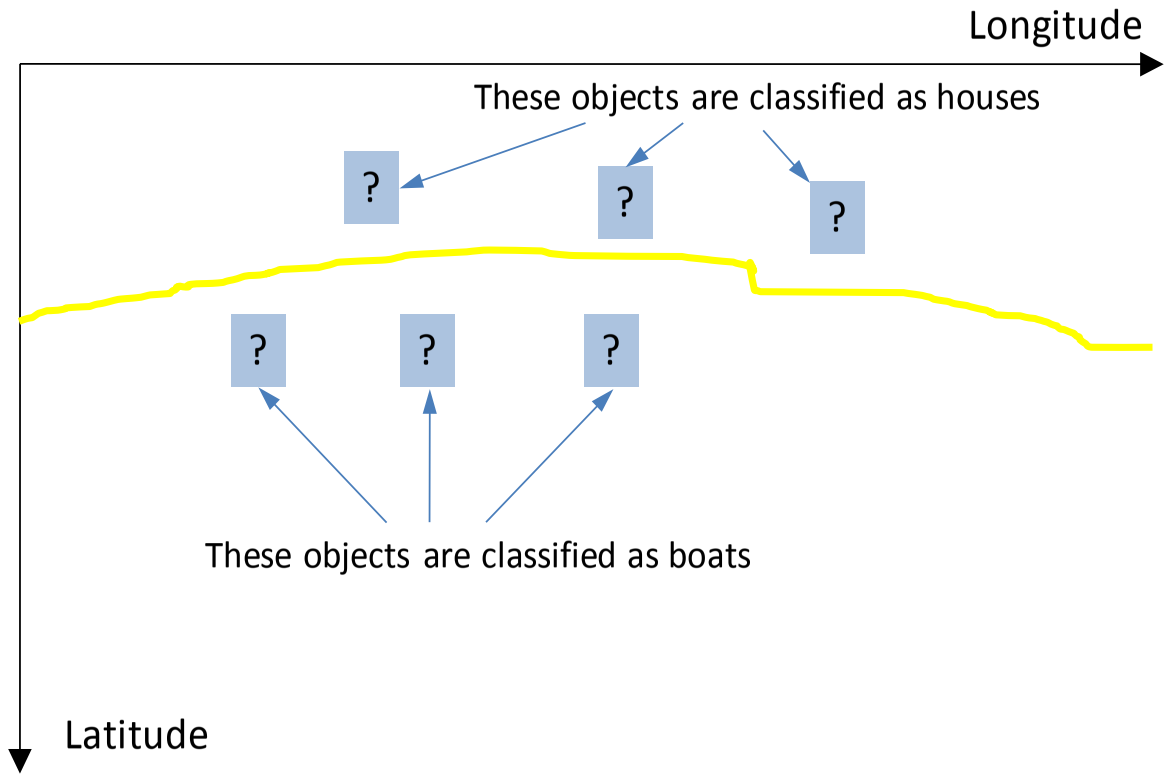
- The methods that build classification models (i.e., "*classification algorithms*") operate very similarly to the previous example.
- First all objects are represented geometrically.

Basic principles of classification



Then the algorithm seeks to find a decision surface that separates classes of objects

Basic principles of classification



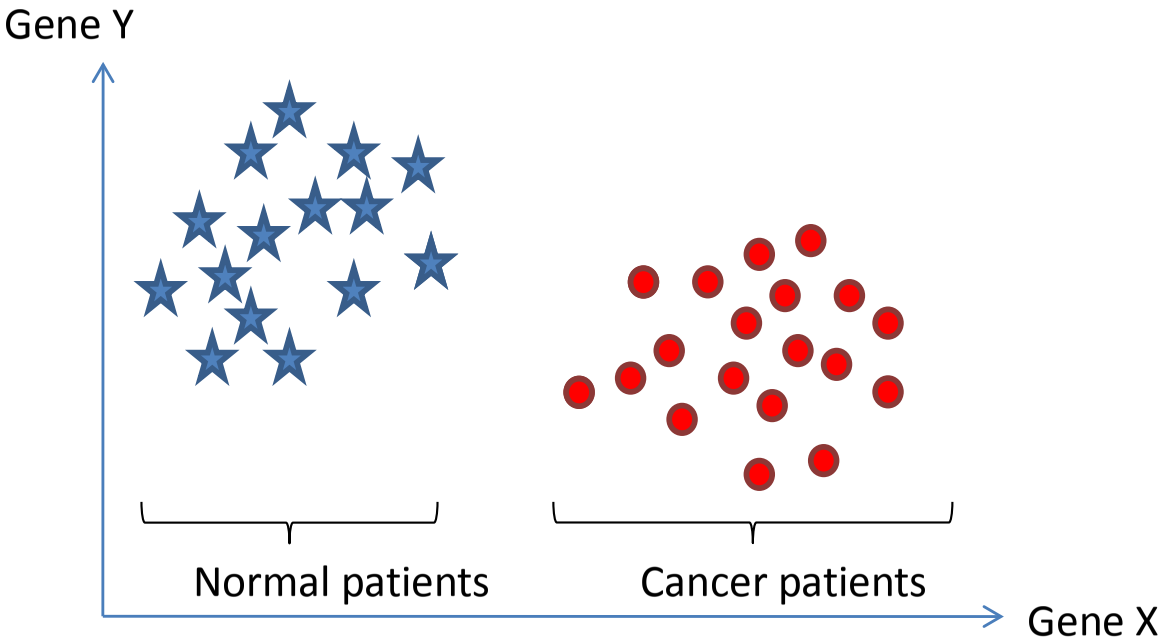
Unseen (new) objects are classified as “boats” if they fall below the decision surface and as “houses” if the fall above it

The Support Vector Machine (SVM) approach

- Support vector machines (SVMs) is a binary classification algorithm that offers a solution to problem #1.
- Extensions of the basic SVM algorithm can be applied to solve problems #1-#5.
- SVMs are important because of (a) theoretical reasons:
  - Robust to very large number of variables and small samples
  - Can learn both simple and highly complex classification models
  - Employ sophisticated mathematical principles to avoid overfitting

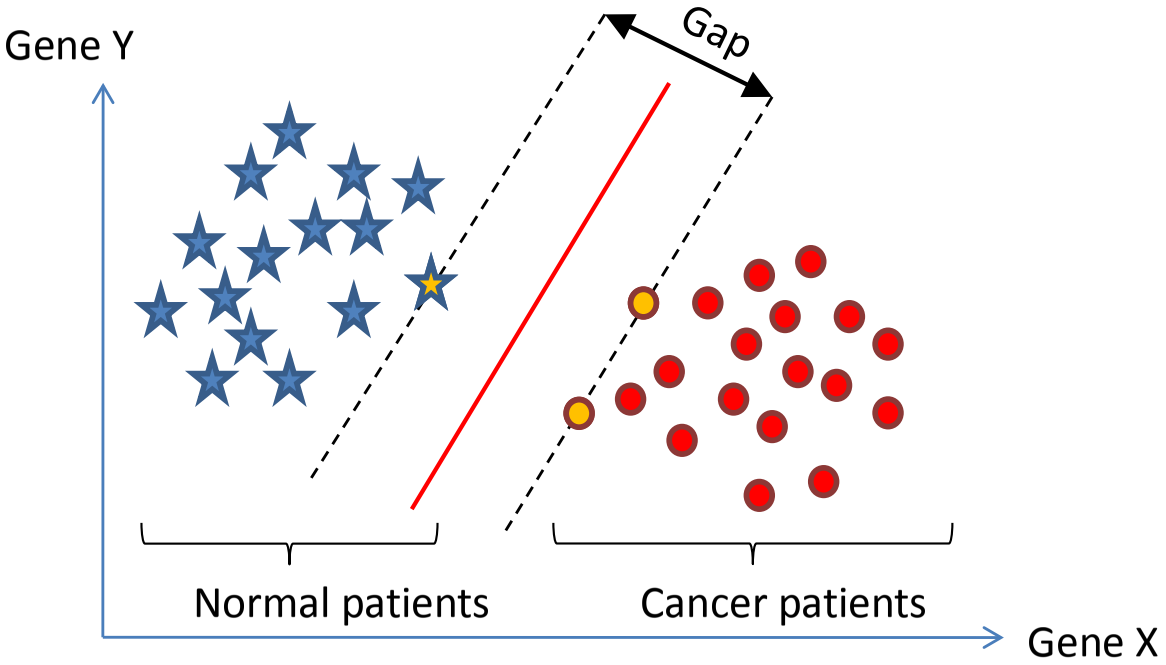
  and (b) superior empirical results.

Main ideas of SVMs



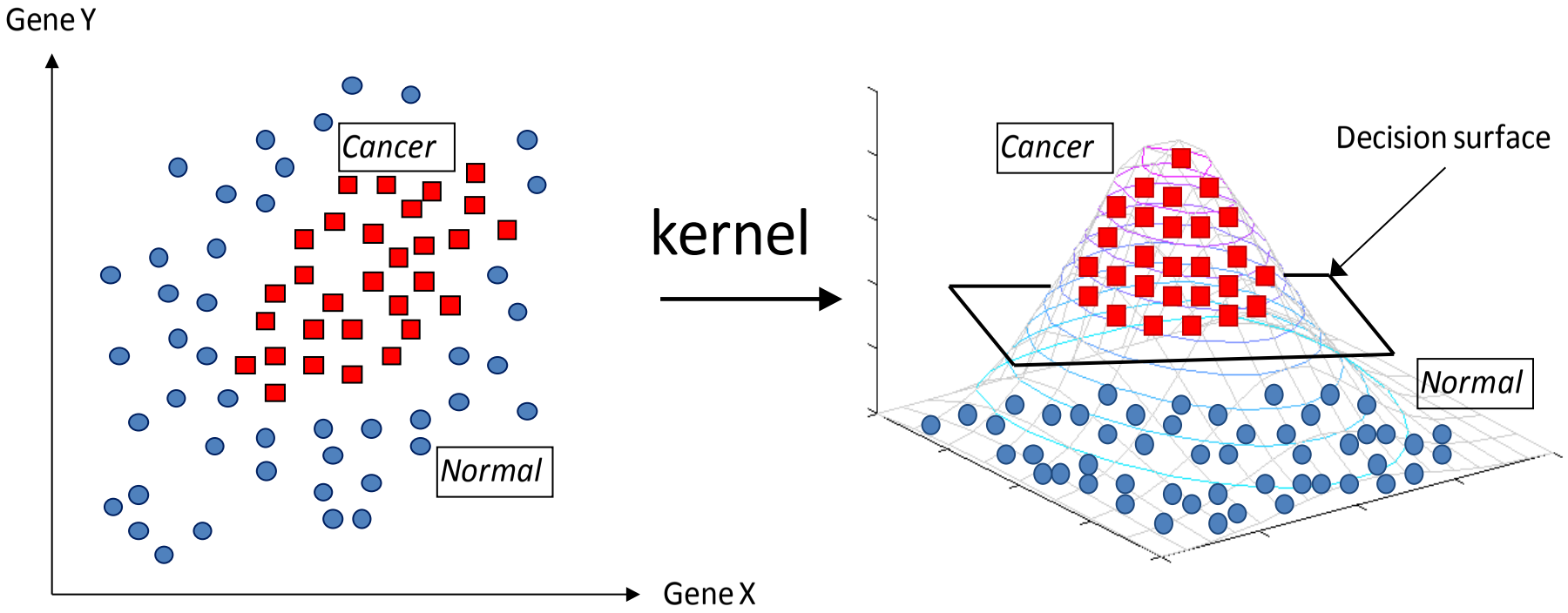
- Consider example dataset described by 2 genes, gene X and gene Y
- Represent patients geometrically (by “vectors”)

Main ideas of SVMs



- Find a linear decision surface ("hyperplane") that can separate patient classes and has the largest distance (i.e., largest "gap" or "margin") between border-line patients (i.e., "support vectors");

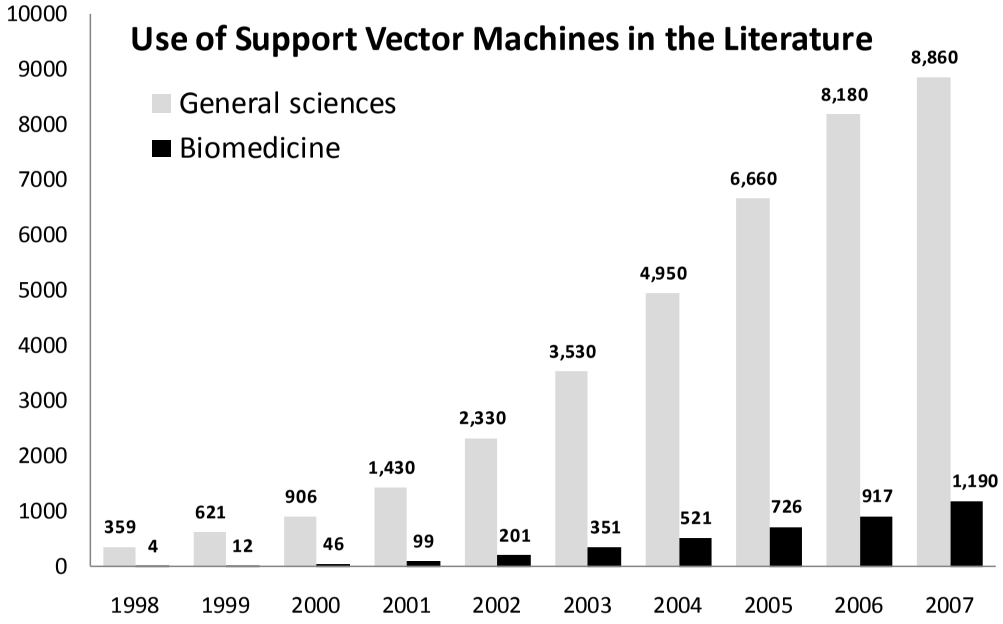
Main ideas of SVMs

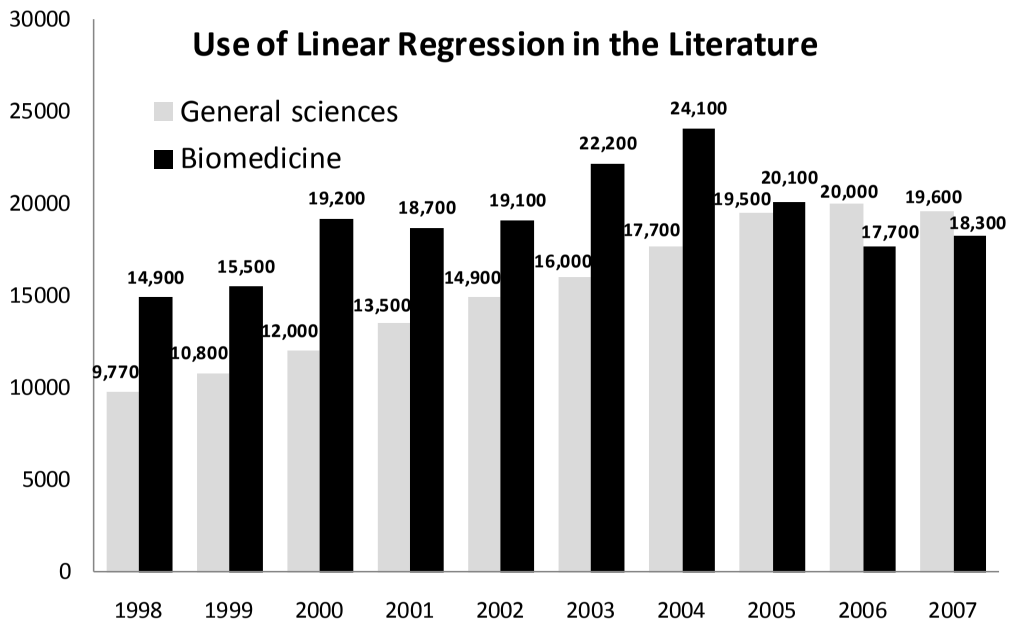


- If such linear decision surface does not exist, the data is mapped into a much higher dimensional space ("feature space") where the separating decision surface is found;
- The feature space is constructed via very clever mathematical projection ("kernel trick").

History of SVMs and usage in the literature

- Support vector machine classifiers have a long history of development starting from the 1960's.
- The most important milestone for development of modern SVMs is the 1992 paper by Boser, Guyon, and Vapnik ("*A training algorithm for optimal margin classifiers*")





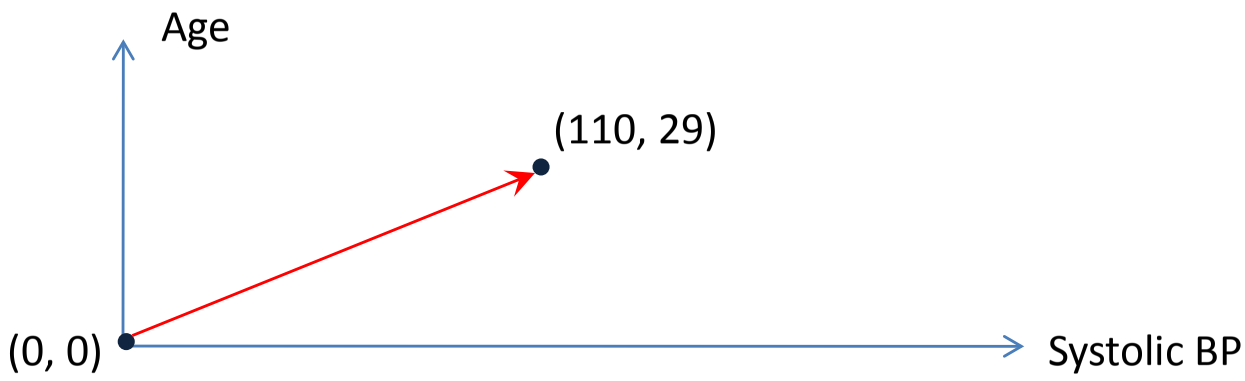
Necessary mathematical concepts

How to represent samples geometrically? Vectors in n-dimensional space ($\mathbb{R}^n$)

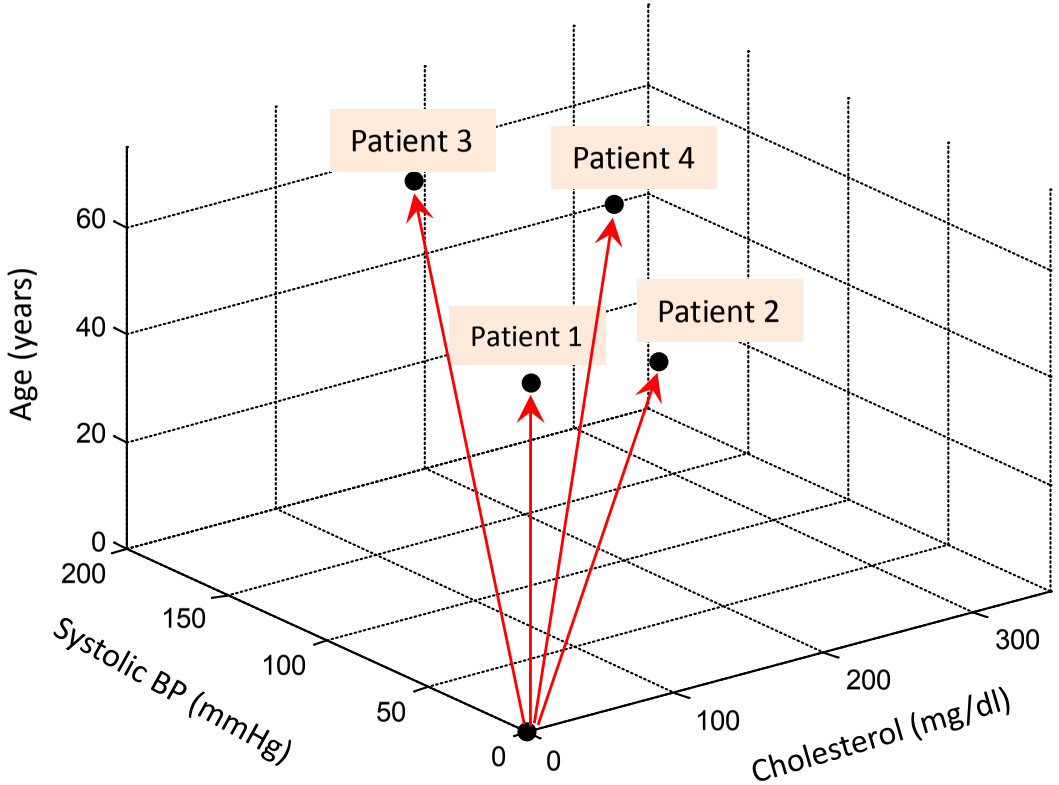
- Assume that a sample/patient is described by *n* characteristics ("features" or "variables")
- **Representation:** Every sample/patient is a vector in $\mathbb{R}^n$ with tail at point with 0 coordinates and arrow-head at point with the feature values.
- **Example:** Consider a patient described by 2 features:

  Systolic BP = 110 and *Age* = 29.

  This patient can be represented as a vector in $\mathbb{R}^2$:

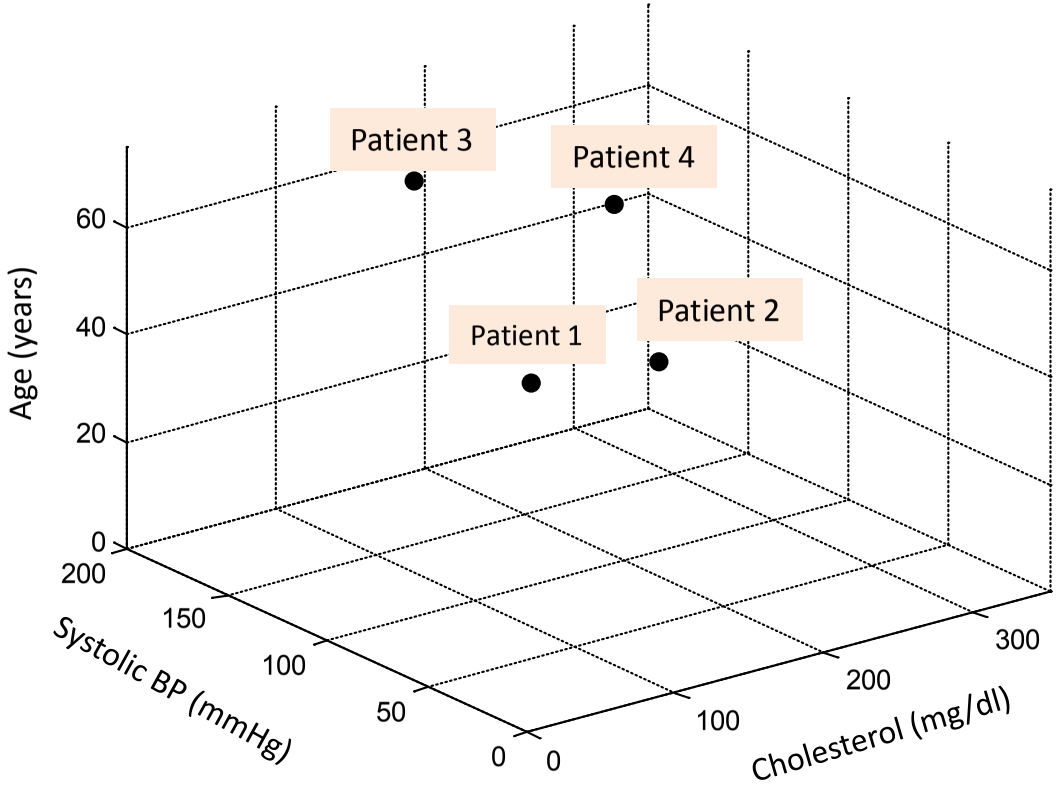


How to represent samples geometrically? Vectors in n-dimensional space ($\mathbb{R}^n$)



Patient id	Cholesterol (mg/dl)	Systolic BP (mmHg)	Age (years)	Tail of the vector	Arrow-head of the vector
1	150	110	35	(0,0,0)	(150, 110, 35)
2	250	120	30	(0,0,0)	(250, 120, 30)
3	140	160	65	(0,0,0)	(140, 160, 65)
4	300	180	45	(0,0,0)	(300, 180, 45)

How to represent samples geometrically? Vectors in n-dimensional space ($\mathbb{R}^n$)

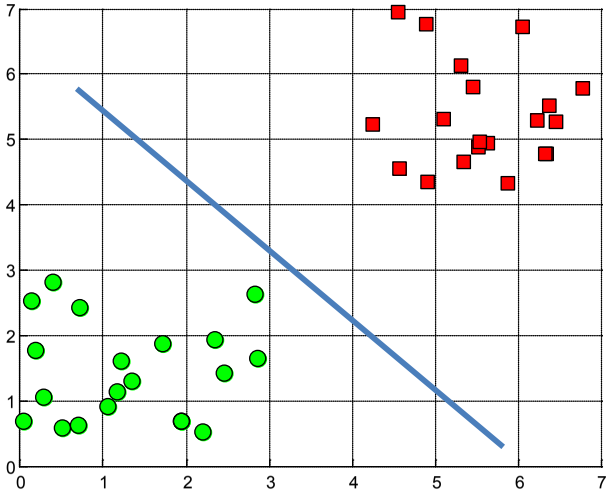


Since we assume that the tail of each vector is at point with 0 coordinates, we will also depict vectors as points (where the arrow-head is pointing).

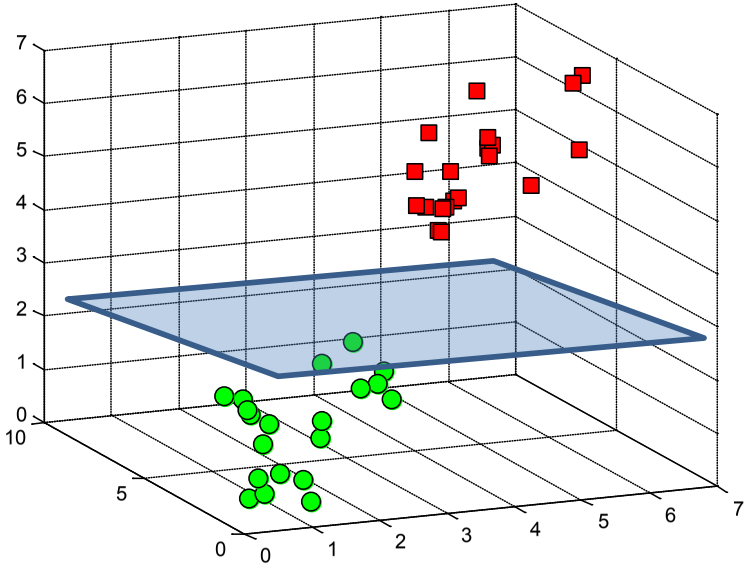
Purpose of vector representation

- Having represented each sample/patient as a vector allows now to geometrically represent the decision surface that separates two groups of samples/patients.

A decision surface in $\mathbb{R}^2$



A decision surface in $\mathbb{R}^3$



- In order to define the decision surface, we need to introduce some basic math elements…

Basic operation on vectors in $\mathbb{R}^n$

1. Multiplication by a scalar

Consider a vector $\vec{a} = (a_1, a_2, \ldots, a_n)$ and a scalar c

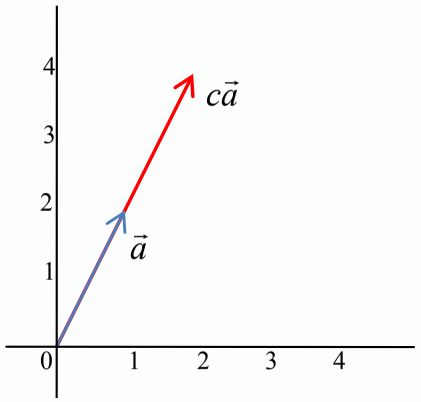
Define: $c\vec{a} = (ca_1, ca_2, \ldots, ca_n)$

When you multiply a vector by a scalar, you "stretch" it in the same or opposite direction depending on whether the scalar is positive or negative.

$\vec{a} = (1,2)$

$c = 2$

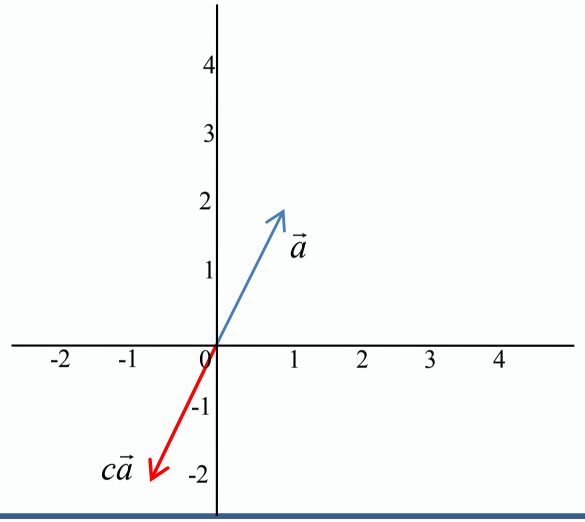
$\vec{c}a = (2,4)$



$\vec{a} = (1,2)$

$c = -1$

$\vec{c}a = (-1,-2)$



Basic operation on vectors in $\mathbb{R}^n$

2. Addition

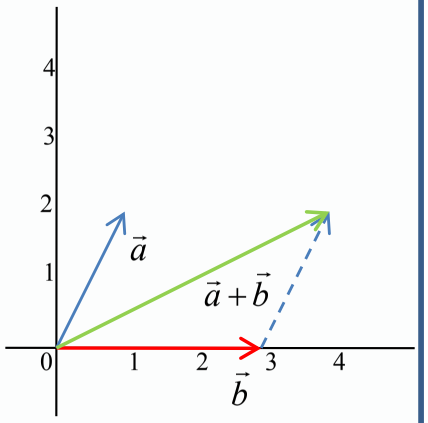
Consider vectors $\vec{a} = (a_1, a_2, \ldots, a_n)$ and $\vec{b} = (b_1, b_2, \ldots, b_n)$

Define: $\vec{a} + \vec{b} = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n)$

$\vec{a} = (1,2)$

$\vec{b} = (3,0)$

$\vec{a} + \vec{b} = (4,2)$



Recall addition of forces in classical mechanics.

Basic operation on vectors in $\mathbb{R}^n$

3. Subtraction

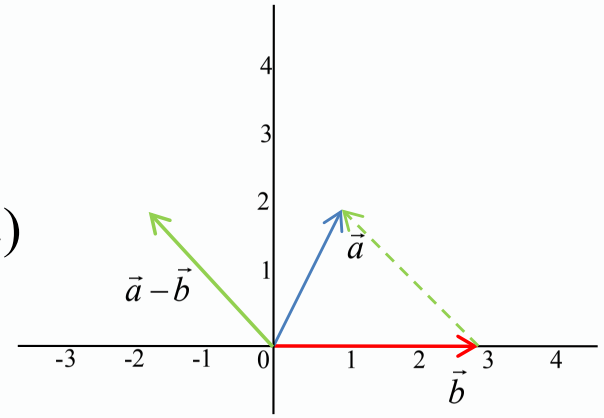
Consider vectors $\vec{a} = (a_1, a_2, \ldots, a_n)$ and $\vec{b} = (b_1, b_2, \ldots, b_n)$

Define: $\vec{a} - \vec{b} = (a_1 - b_1, a_2 - b_2, \ldots, a_n - b_n)$

$\vec{a} = (1,2)$

$\vec{b} = (3,0)$

$\vec{a} - \vec{b} = (-2,2)$



What vector do we need to add to $\vec{b}$ to get $\vec{a}$? I.e., similar to subtraction of real numbers.

Basic operation on vectors in $\mathbb{R}^n$

4. Euclidian length or L2-norm

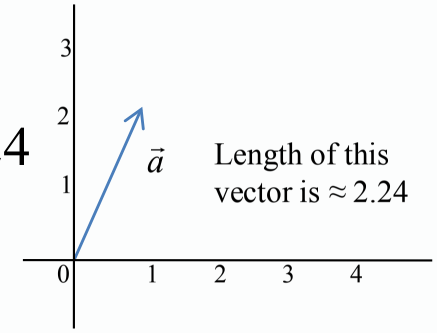
Consider a vector $\vec{a} = (a_1, a_2, \ldots, a_n)$

Define the L2-norm: $\|\vec{a}\|_2 = \sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}$

We often denote the L2-norm without subscript, i.e. $\|\vec{a}\|$

$\vec{a} = (1,2)$

$\|\vec{a}\|_2 = \sqrt{5} \approx 2.24$



L2-norm is a typical way to measure length of a vector; other methods to measure length also exist.

Basic operation on vectors in $\mathbb{R}^n$

5. Dot product

Consider vectors $\vec{a} = (a_1, a_2, ..., a_n)$ and $\vec{b} = (b_1, b_2, ..., b_n)$

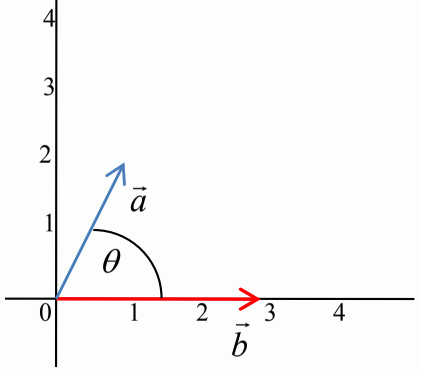
Define dot product: $\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + ... + a_n b_n = \sum_{i=1}^{n} a_i b_i$

The law of cosines says that $\vec{a} \cdot \vec{b} = \| \vec{a} \|_2 \| \vec{b} \|_2 \cos\theta$ where θ is the angle between $\vec{a}$ and $\vec{b}$. Therefore, when the vectors are perpendicular $\vec{a} \cdot \vec{b} = 0$.

$\vec{a} = (1,2)$

$\vec{b} = (3,0)$

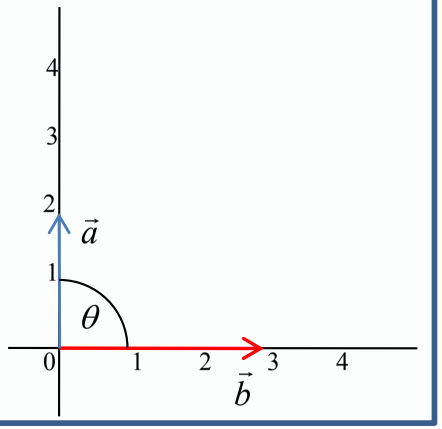
$\vec{a} \cdot \vec{b} = 3$



$\vec{a} = (0,2)$

$\vec{b} = (3,0)$

$\vec{a} \cdot \vec{b} = 0$



Basic operation on vectors in $\mathbb{R}^n$

5. Dot product (continued)

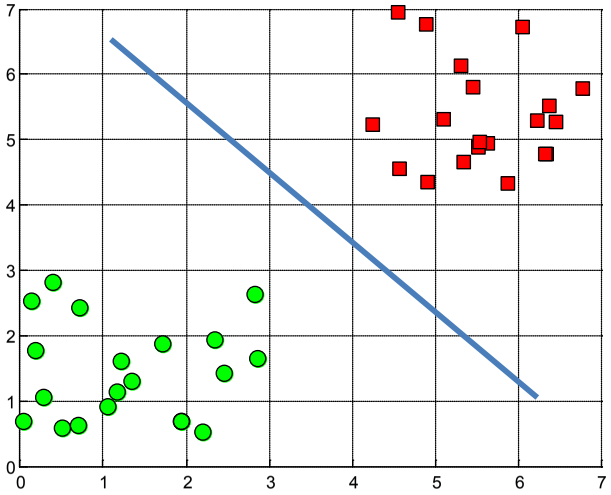
$$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i$$

- Property: $\vec{a} \cdot \vec{a} = a_1 a_1 + a_2 a_2 + \ldots + a_n a_n = \left\| \vec{a} \right\|_2^2$
- In the classical regression equation $y = \vec{w} \cdot \vec{x} + b$ the response variable *y* is just a dot product of the vector representing patient characteristics ($\vec{x}$) and the regression weights vector ($\vec{w}$) which is common across all patients plus an offset *b*.

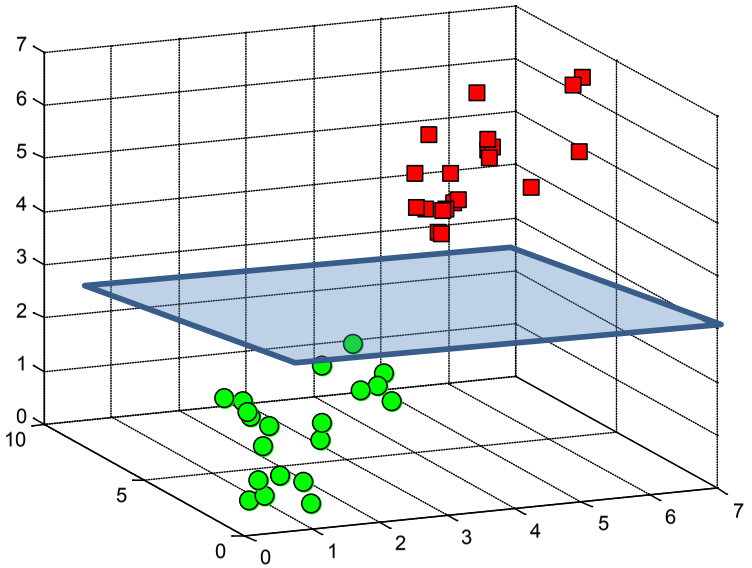
Hyperplanes as decision surfaces

- A hyperplane is a linear decision surface that splits the space into two parts;
- It is obvious that a hyperplane is a binary classifier.

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



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



A hyperplane in $\mathbb{R}^n$ is an *n*-1 dimensional subspace

Equation of a hyperplane

First we show with show the definition of hyperplane by an interactive demonstration.

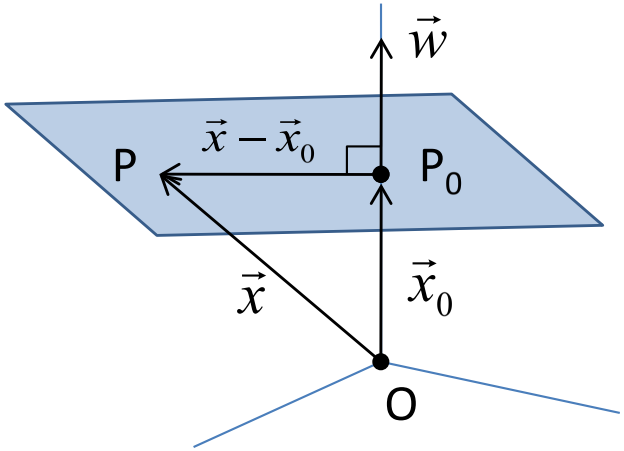
Click here for demo to begin

or go to http://www.dsl-lab.org/svm_tutorial/planedemo.html

Source: http://www.math.umn.edu/~nykamp/

Equation of a hyperplane

Consider the case of $\mathbb{R}^3$:



An equation of a hyperplane is defined by a point (P_0) and a perpendicular vector to the plane ($\vec{w}$) at that point.

Define vectors: $\vec{x}_0 = \overrightarrow{OP_0}$ and $\vec{x} = \overrightarrow{OP}$, where *P* is an arbitrary point on a hyperplane.

A condition for *P* to be on the plane is that the vector $\vec{x} - \vec{x}_0$ is perpendicular to $\vec{w}$:

$$\vec{w} \cdot (\vec{x} - \vec{x}_0) = 0 \quad \text{or}$$

$$\vec{w} \cdot \vec{x} - \vec{w} \cdot \vec{x}_0 = 0 \quad \text{define } b = -\vec{w} \cdot \vec{x}_0$$

$$\boxed{\vec{w} \cdot \vec{x} + b = 0}$$

The above equations also hold for $\mathbb{R}^n$ when n>3.

Equation of a hyperplane

Example

$\vec{w} = (4,-1,6)$

$P_0 = (0,1,-7)$

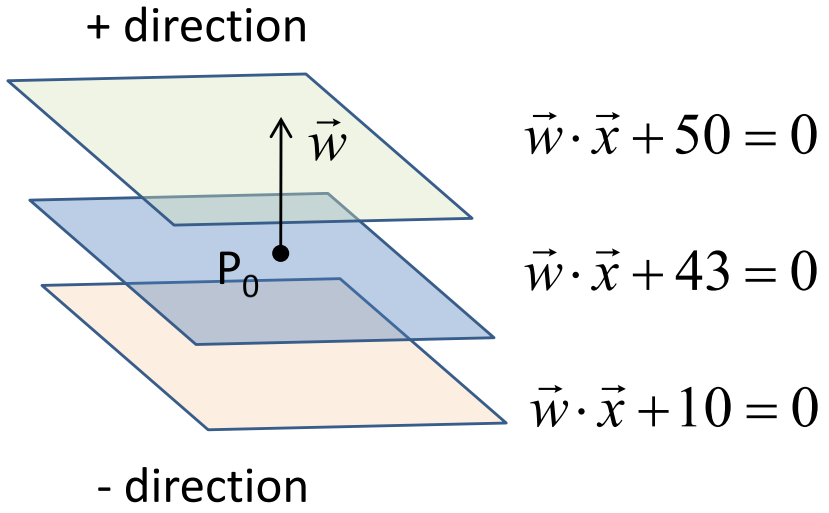
$b = -\vec{w} \cdot \vec{x}_0 = -(0-1-42) = 43$

$\Rightarrow \vec{w} \cdot \vec{x} + 43 = 0$

$\Rightarrow (4,-1,6) \cdot \vec{x} + 43 = 0$

$\Rightarrow (4,-1,6) \cdot (x_{(1)}, x_{(2)}, x_{(3)}) + 43 = 0$

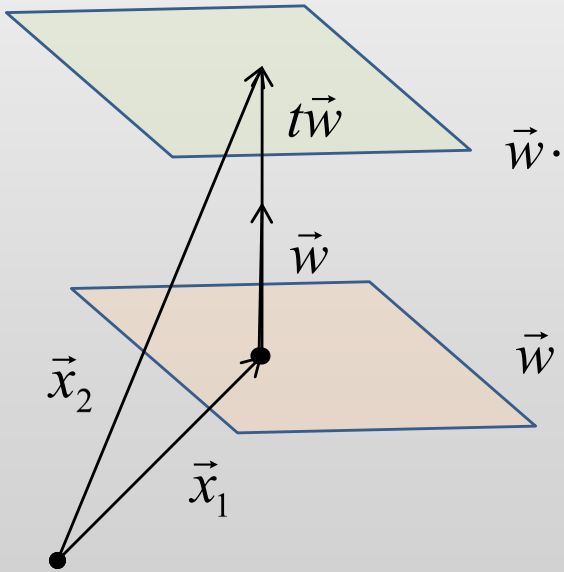
$\Rightarrow 4x_{(1)} - x_{(2)} + 6x_{(3)} + 43 = 0$



What happens if the *b* coefficient changes?
The hyperplane moves along the direction of $\vec{w}$.
We obtain "parallel hyperplanes".

Distance between two parallel hyperplanes $\vec{w} \cdot \vec{x} + b_1 = 0$ and $\vec{w} \cdot \vec{x} + b_2 = 0$ is equal to $D = |b_1 - b_2| / \|\vec{w}\|$.

(Derivation of the distance between two parallel hyperplanes)



$$\vec{w}\cdot\vec{x}+b_2=0$$

$$\vec{w}\cdot\vec{x}+b_1=0$$

$$\vec{x}_2=\vec{x}_1+t\vec{w}$$

$$D=\left\|t\vec{w}\right\|=\left|t\right|\left\|\vec{w}\right\|$$

$$\vec{w}\cdot\vec{x}_2+b_2=0$$

$$\vec{w}\cdot(\vec{x}_1+t\vec{w})+b_2=0$$

$$\vec{w}\cdot\vec{x}_1+t\left\|\vec{w}\right\|^2+b_2=0$$

$$(\vec{w}\cdot\vec{x}_1+b_1)-b_1+t\left\|\vec{w}\right\|^2+b_2=0$$

$$-b_1+t\left\|\vec{w}\right\|^2+b_2=0$$

$$t=(b_1-b_2)/\left\|\vec{w}\right\|^2$$

$$\Rightarrow D=\left|t\right|\left\|\vec{w}\right\|=\left|b_1-b_2\right|/\left\|\vec{w}\right\|$$

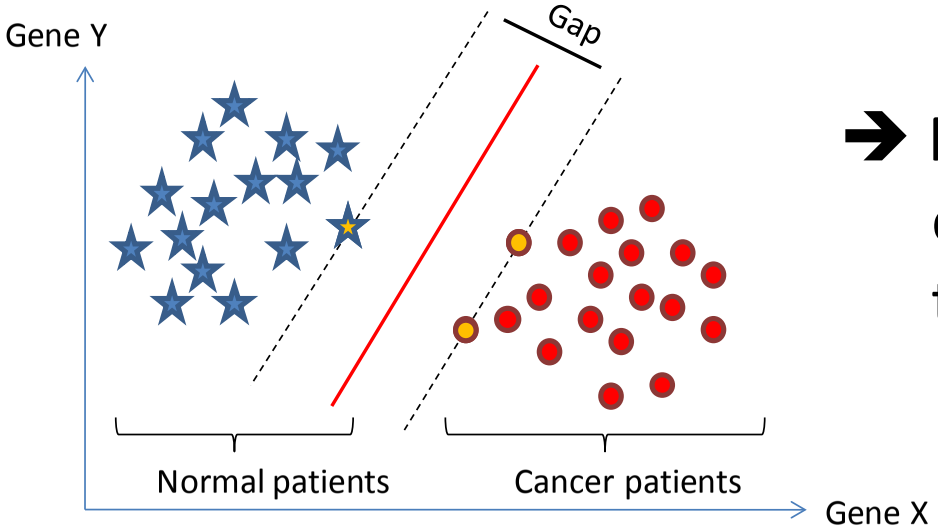
Recap

We know…

- How to represent patients (as “vectors”)
- How to define a linear decision surface (“hyperplane”)

We need to know…

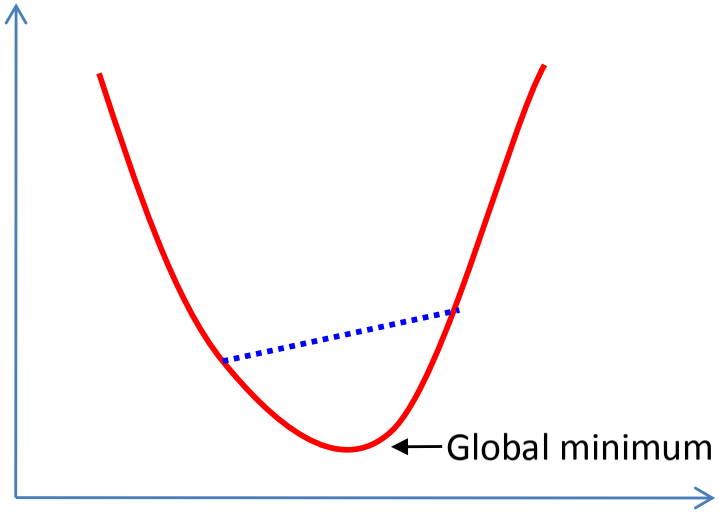
- How to efficiently compute the hyperplane that separates two classes with the largest “gap”?



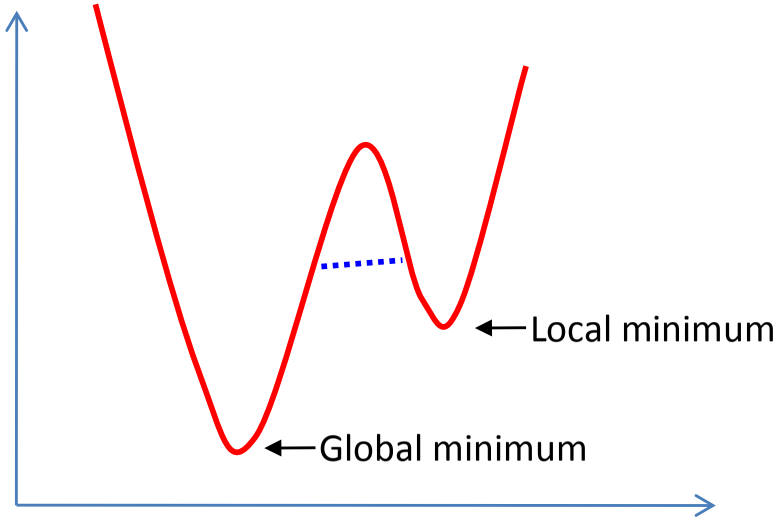
➔ Need to introduce basics of relevant optimization theory

Basics of optimization: Convex functions

- A function is called *convex* if the function lies below the straight line segment connecting two points, for any two points in the interval.
- Property: Any local minimum is a global minimum!



Convex function



Non-convex function

Basics of optimization: Quadratic programming (QP)

- Quadratic programming (QP) is a special optimization problem: the function to optimize ("*objective*") is quadratic, subject to linear *constraints*.
- Convex QP problems have convex objective functions.
- These problems can be solved easily and efficiently by greedy algorithms (because every local minimum is a global minimum).

Basics of optimization: Example QP problem

Consider $\vec{x} = (x_1, x_2)$

Minimize $\frac{1}{2} \| \vec{x} \|_2^2$ subject to $x_1 + x_2 - 1 \geq 0$

quadratic objective *linear constraints*

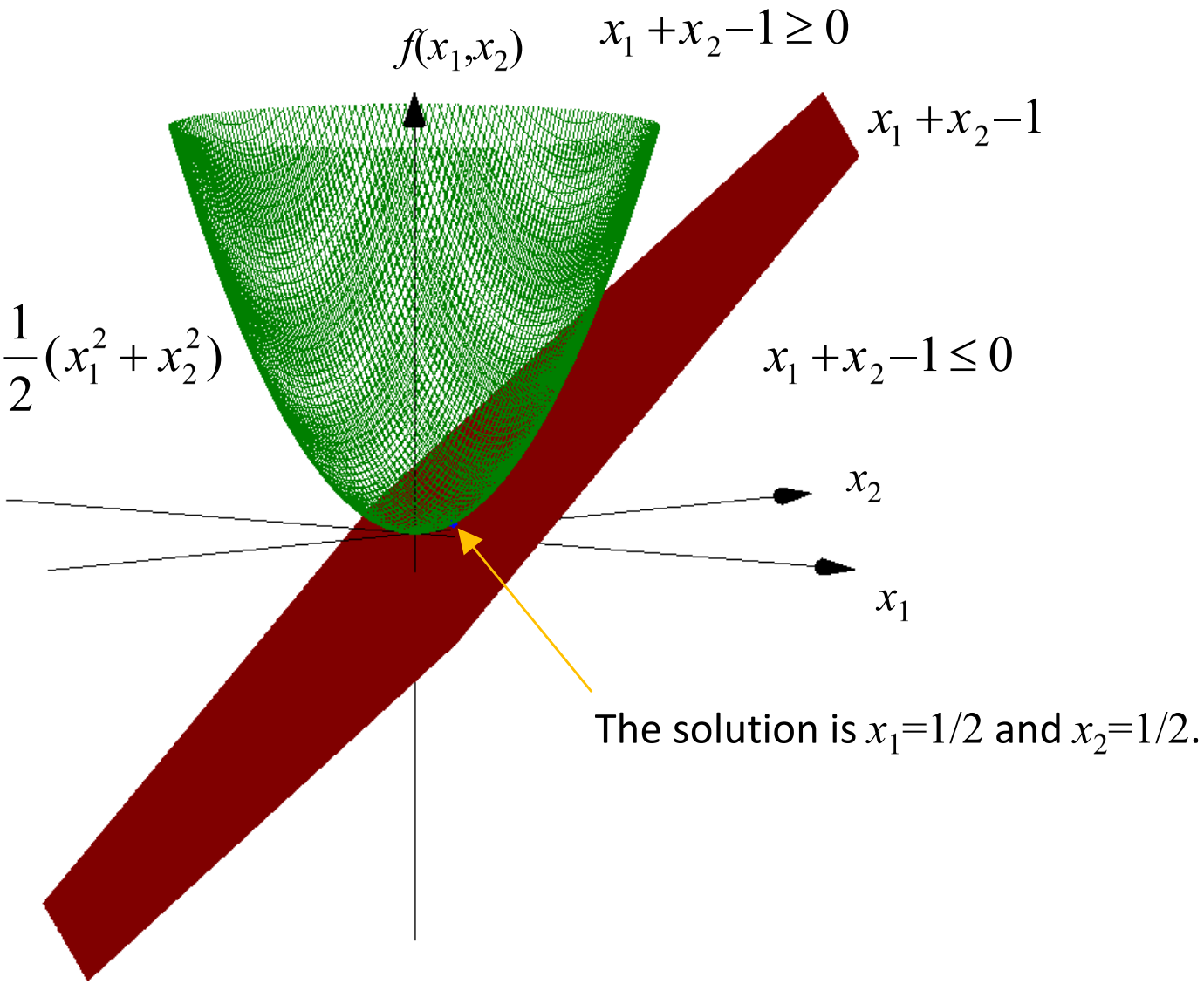
This is QP problem, and it is a convex QP as we will see later

We can rewrite it as:

Minimize $\frac{1}{2}(x_1^2 + x_2^2)$ subject to $x_1 + x_2 - 1 \geq 0$

quadratic objective *linear constraints*

Basics of optimization: Example QP problem

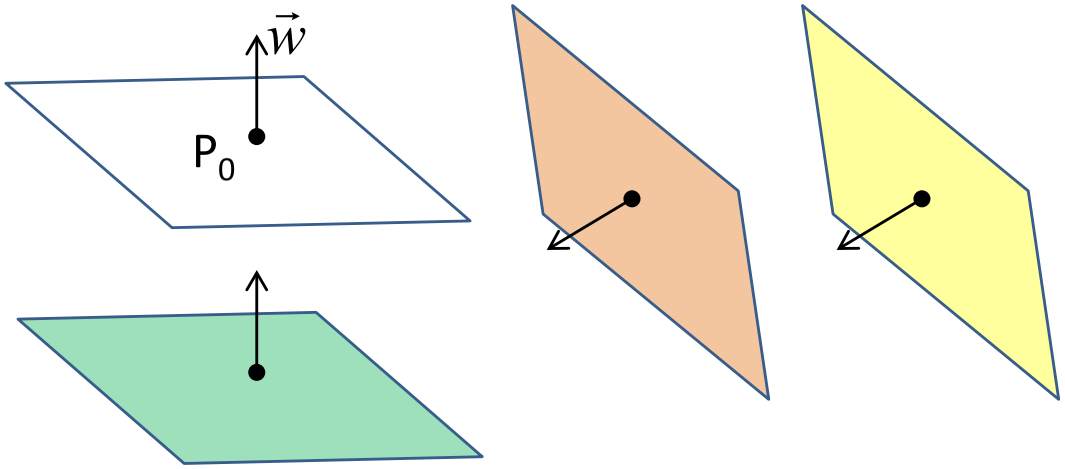


Congratulations! You have mastered all math elements needed to understand support vector machines.

Now, let us strengthen your knowledge by a quiz ☺

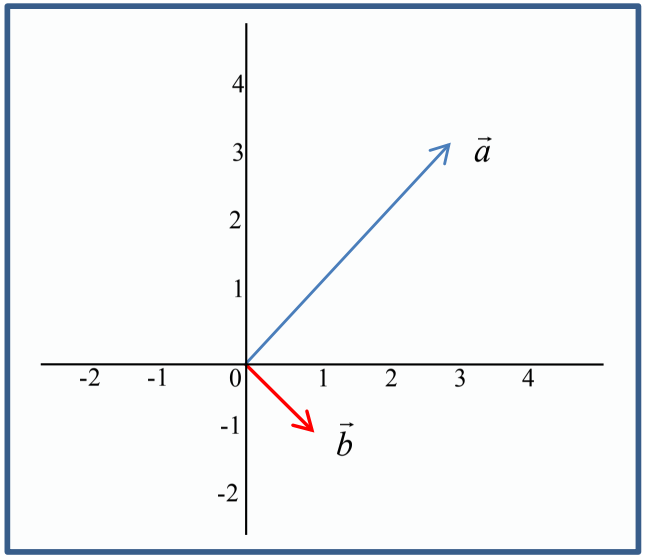
Quiz

1) Consider a hyperplane shown with white. It is defined by equation: $\vec{w} \cdot \vec{x} + 10 = 0$
Which of the three other hyperplanes can be defined by equation: $\vec{w} \cdot \vec{x} + 3 = 0$?



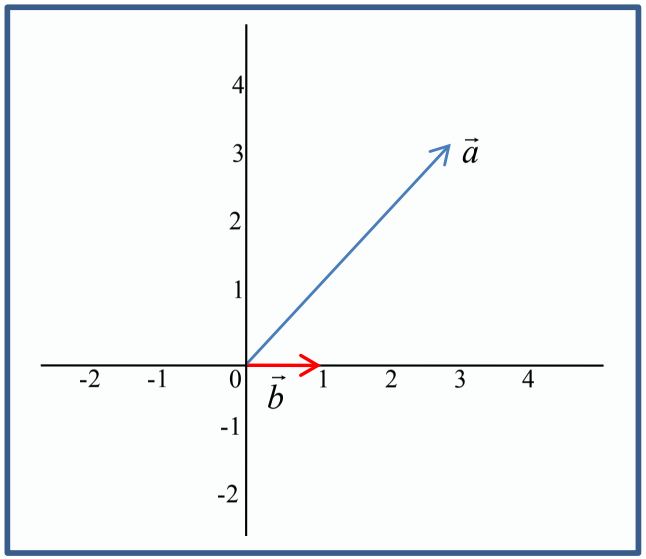
- Orange
- Green
- Yellow

2) What is the dot product between vectors $\vec{a} = (3,3)$ and $\vec{b} = (1,-1)$?

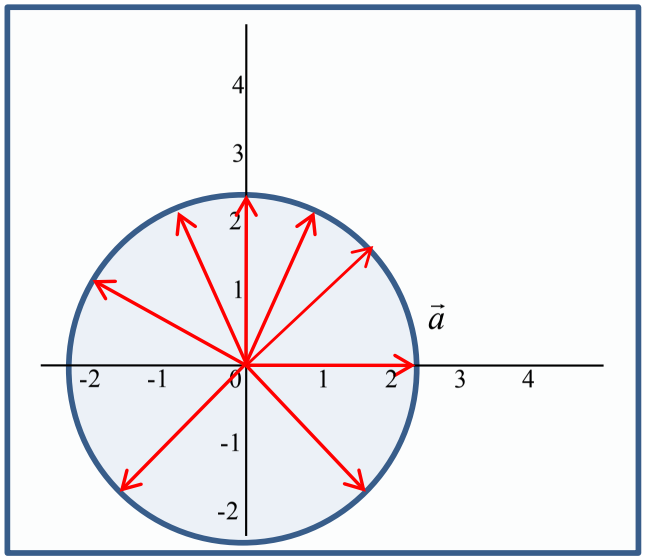


Quiz

3) What is the dot product between vectors $\vec{a} = (3,3)$ and $\vec{b} = (1,0)$?



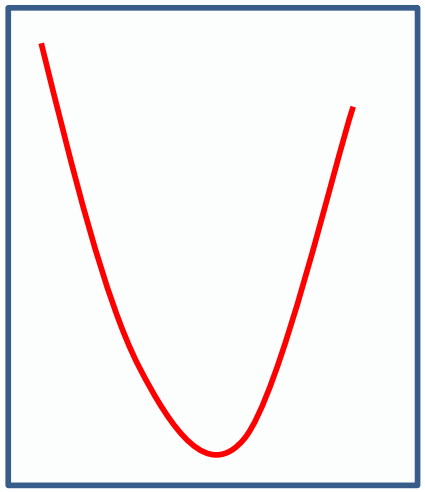
4) What is the length of a vector $\vec{a} = (2,0)$ and what is the length of all other red vectors in the figure?



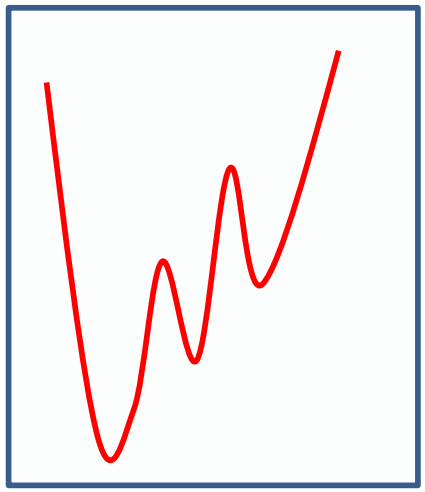
Quiz

5) Which of the four functions is/are convex?

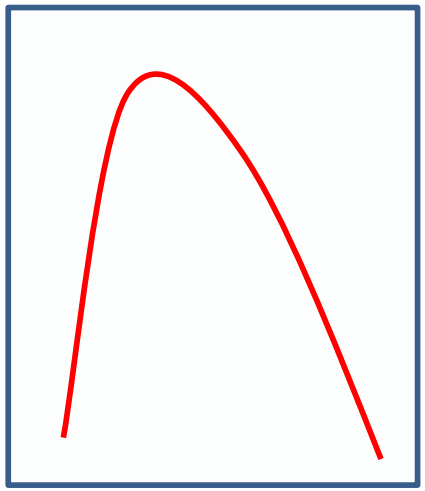
1



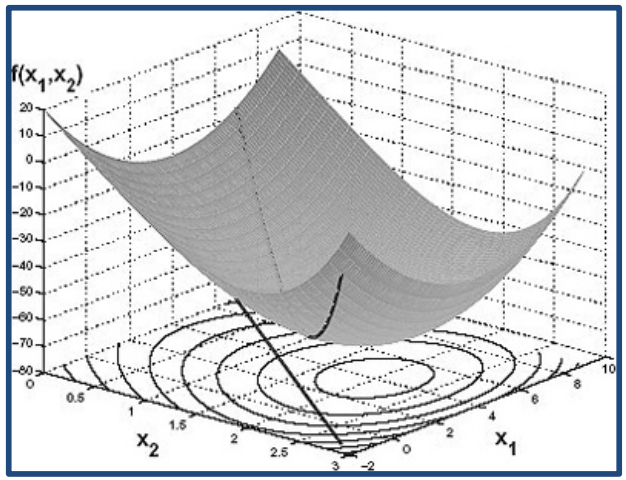
2



3



4

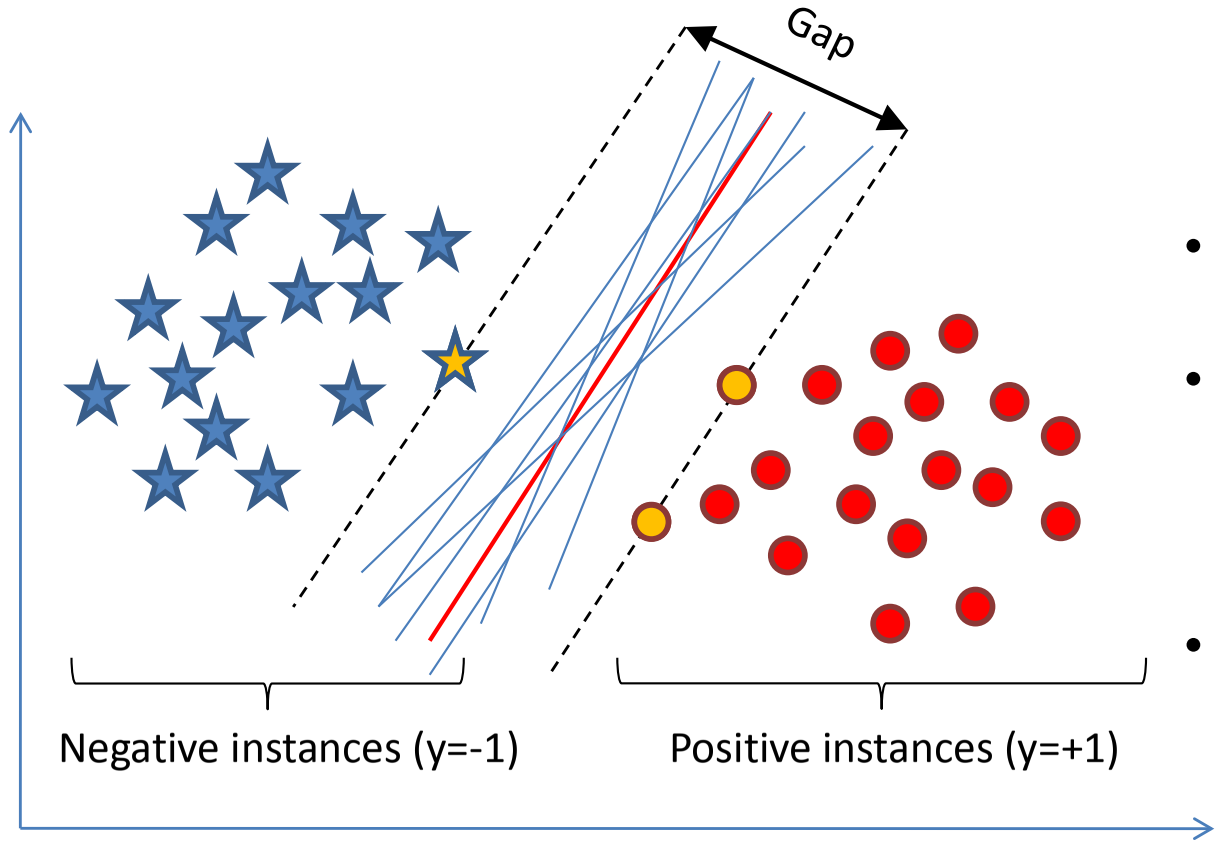


Support vector machines for binary classification: classical formulation

Case 1: Linearly separable data; "Hard-margin" linear SVM

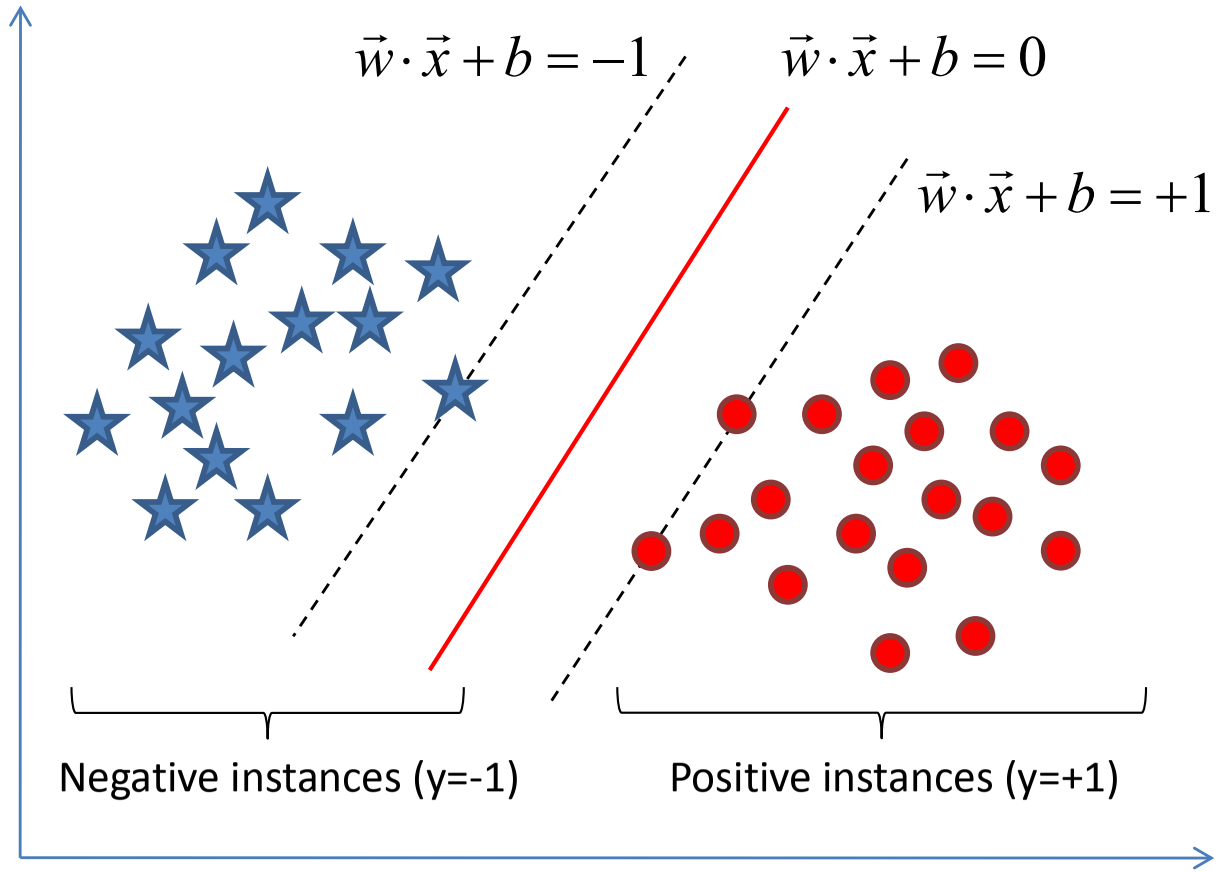
Given training data: $\vec{x}_1, \vec{x}_2, ..., \vec{x}_N \in R^n$

$y_1, y_2, ..., y_N \in \{-1, +1\}$



- Want to find a classifier (hyperplane) to separate negative instances from the positive ones.
- An infinite number of such hyperplanes exist.
- SVMs finds the hyperplane that maximizes the gap between data points on the boundaries (so-called "support vectors").
- If the points on the boundaries are not informative (e.g., due to noise), SVMs will not do well.

Statement of linear SVM classifier



The gap is distance between parallel hyperplanes:

$\vec{w}\cdot\vec{x}+b=-1$ and
$\vec{w}\cdot\vec{x}+b=+1$

Or equivalently:

$\vec{w}\cdot\vec{x}+(b+1)=0$
$\vec{w}\cdot\vec{x}+(b-1)=0$

We know that

$$D=\left|b_1-b_2\right|/\left\|\vec{w}\right\|$$

Therefore:

$$D=2/\left\|\vec{w}\right\|$$

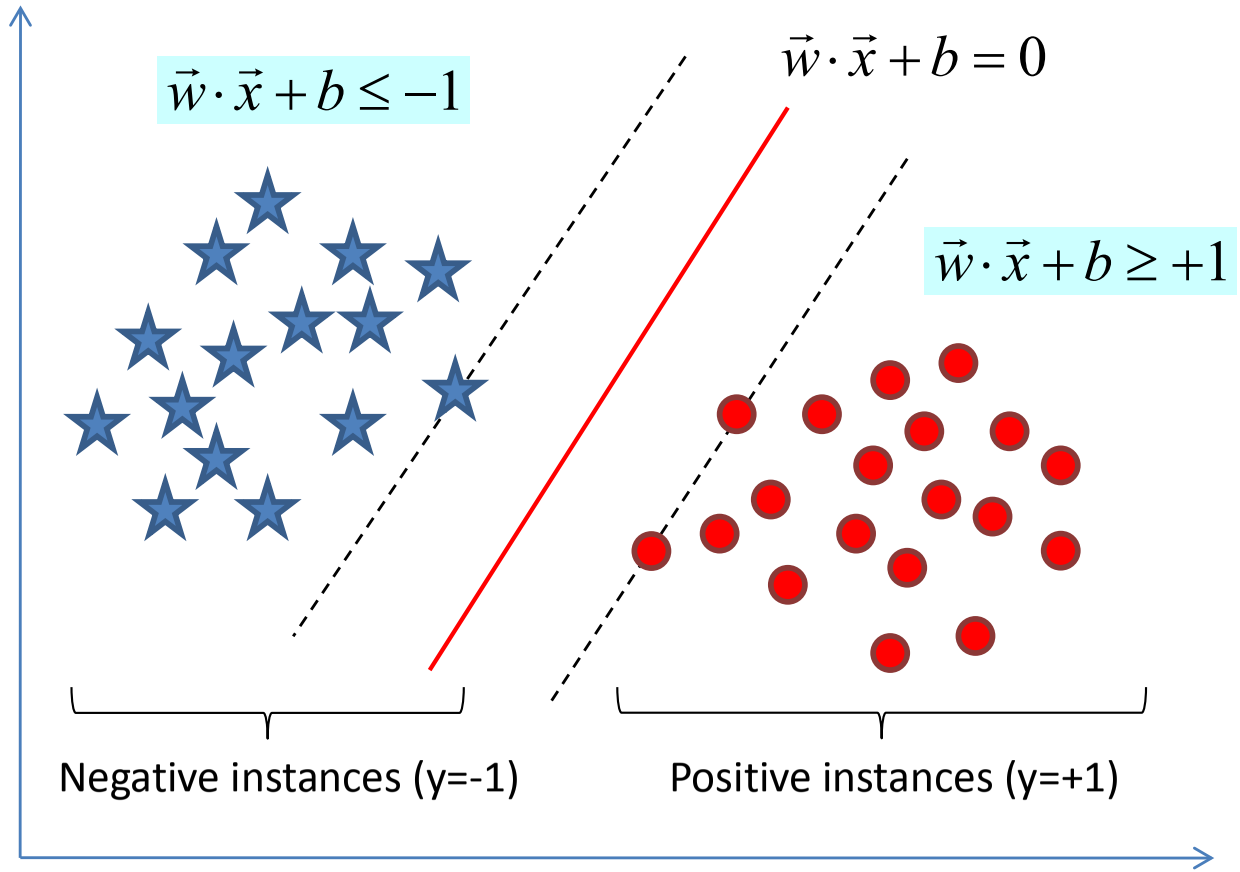
Since we want to maximize the gap,

we need to minimize $\left\|\vec{w}\right\|$

or equivalently minimize $\frac{1}{2}\left\|\vec{w}\right\|^2$

($\frac{1}{2}$ is convenient for taking derivative later on)

Statement of linear SVM classifier



In addition we need to impose constraints that all instances are correctly classified. In our case:

$\vec{w}\cdot\vec{x}_i+b\leq -1$ if $y_i=-1$

$\vec{w}\cdot\vec{x}_i+b\geq +1$ if $y_i=+1$

Equivalently:

$$y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1$$

In summary:

Want to minimize $\frac{1}{2}\left\|\vec{w}\right\|^2$ subject to $y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1$ for $i=1,\ldots,N$

Then given a new instance x, the classifier is $f(\vec{x})=sign(\vec{w}\cdot\vec{x}+b)$

SVM optimization problem: Primal formulation

Minimize $\frac{1}{2}\sum_{i=1}^{n} w_i^2$ subject to $y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0$ for $i = 1,\ldots,N$

Objective function

Constraints

- This is called "*primal formulation of linear SVMs*".
- It is a convex quadratic programming (QP) optimization problem with n variables (w_i, $i = 1,\ldots,n$), where n is the number of features in the dataset.

SVM optimization problem: Dual formulation

- The previous problem can be recast in the so-called "*dual form*" giving rise to "*dual formulation of linear SVMs*".
- It is also a convex quadratic programming problem but with *N* variables (α_i ,*i* = 1,…,*N*), where *N* is the number of samples.

Maximize $\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$ subject to $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$.

Objective function Constraints

Then the w-vector is defined in terms of α_i: $\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$

And the solution becomes: $f(\vec{x}) = sign(\sum_{i=1}^{N} \alpha_i y_i \vec{x}_i \cdot \vec{x} + b)$

SVM optimization problem: Benefits of using dual formulation

1) No need to access original data, need to access only dot products.

Objective function: $\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$

Solution: $f(\vec{x}) = sign(\sum_{i=1}^{N}\alpha_i y_i \vec{x}_i \cdot \vec{x} + b)$

2) Number of free parameters is bounded by the number of support vectors and not by the number of variables (beneficial for high-dimensional problems).

E.g., if a microarray dataset contains 20,000 genes and 100 patients, then need to find only up to 100 parameters!

(Derivation of dual formulation)

Minimize $\frac{1}{2}\sum_{i=1}^{n} w_i^2$ subject to $y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0$ for $i = 1,\ldots,N$

Objective function

Constraints

Apply the method of Lagrange multipliers.

Define Lagrangian $\Lambda_P(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\sum_{i=1}^{n} w_i^2 - \sum_{i=1}^{N} \alpha_i \left(y_i(\vec{w}\cdot\vec{x}_i + b) - 1\right)$

a vector with *n* elements

a vector with *N* elements

We need to minimize this Lagrangian with respect to $\vec{w}, b$ and simultaneously require that the derivative with respect to $\vec{\alpha}$ vanishes , all subject to the constraints that $\alpha_i \geq 0$.

(Derivation of dual formulation)

If we set the derivatives with respect to $\vec{w}, b$ to 0, we obtain:

$$\frac{\partial \Lambda_P(\vec{w}, b, \vec{\alpha})}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial \Lambda_P(\vec{w}, b, \vec{\alpha})}{\partial \vec{w}} = 0 \Rightarrow \vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$$

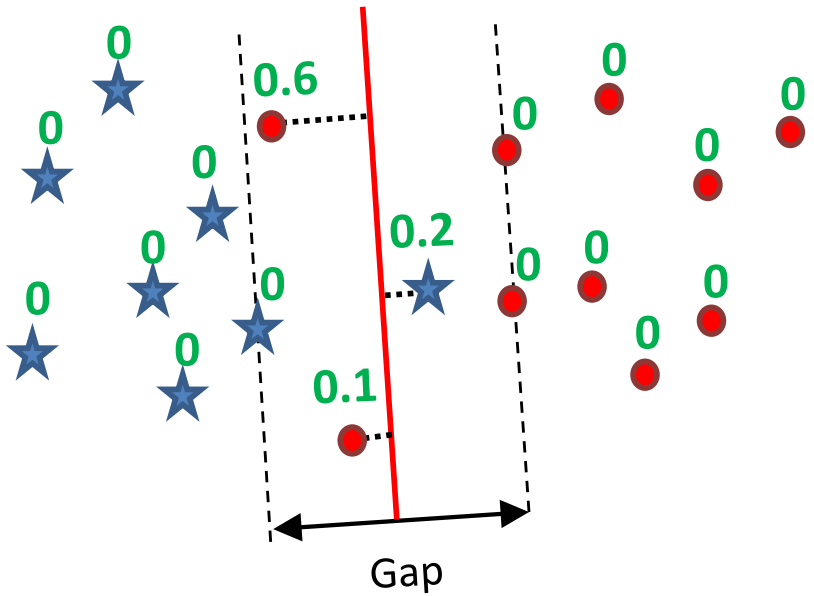
We substitute the above into the equation for $\Lambda_P(\vec{w}, b, \vec{\alpha})$ and obtain "*dual formulation of linear SVMs*":

$$\Lambda_D(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

We seek to maximize the above Lagrangian with respect to $\vec{\alpha}$, subject to the constraints that $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$.

Case 2: Not linearly separable data; "Soft-margin" linear SVM

What if the data is not linearly separable? E.g., there are outliers or noisy measurements, or the data is slightly non-linear.



Want to handle this case without changing the family of decision functions.

Approach:

Assign a "slack variable" to each instance $\xi_i \geq 0$, which can be thought of distance from the separating hyperplane if an instance is misclassified and 0 otherwise.

Want to minimize $\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\xi_i$ subject to $y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \xi_i$ for $i = 1,\ldots,N$

Then given a new instance x, the classifier is $f(x) = sign(\vec{w}\cdot\vec{x} + b)$

Two formulations of soft-margin linear SVM

Primal formulation:

Minimize $\frac{1}{2}\sum_{i=1}^{n} w_i^2 + C\sum_{i=1}^{N}\xi_i$ subject to $y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \xi_i$ for $i = 1,\ldots,N$

Objective function — Constraints

Dual formulation:

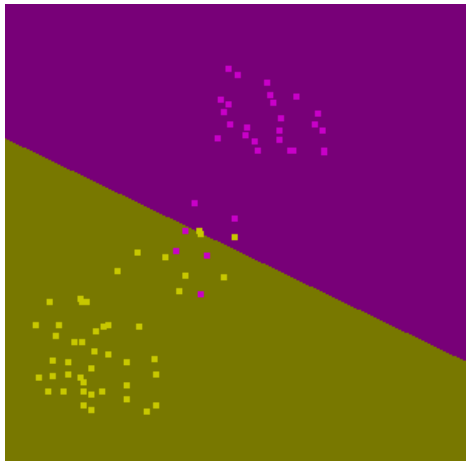
Minimize $\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \vec{x}_i\cdot\vec{x}_j$ subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{N}\alpha_i y_i = 0$

for $i = 1,\ldots,N$.

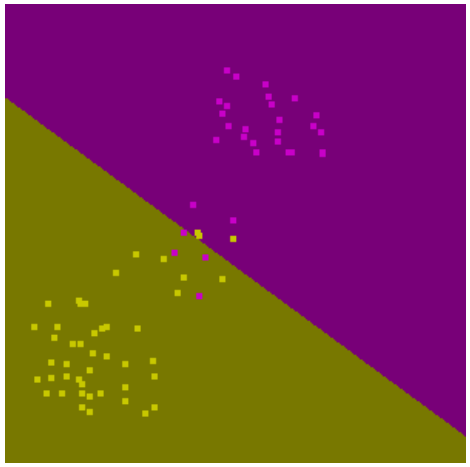
Objective function — Constraints

Parameter *C* in soft-margin SVM

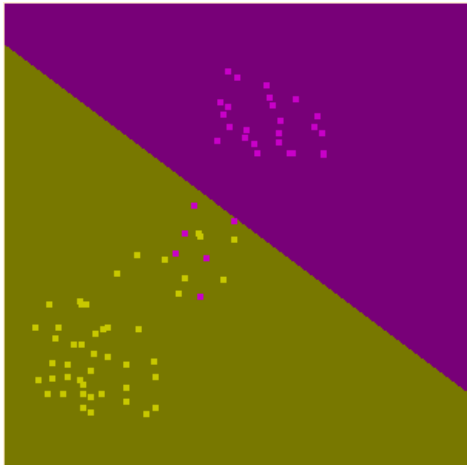
Minimize $\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\xi_i$ subject to $y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \xi_i$ for $i = 1,\ldots,N$



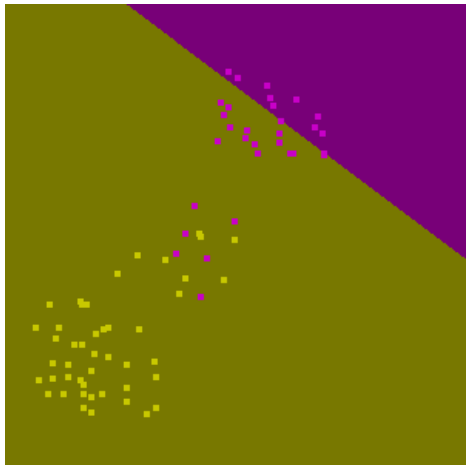
C=100



C=1



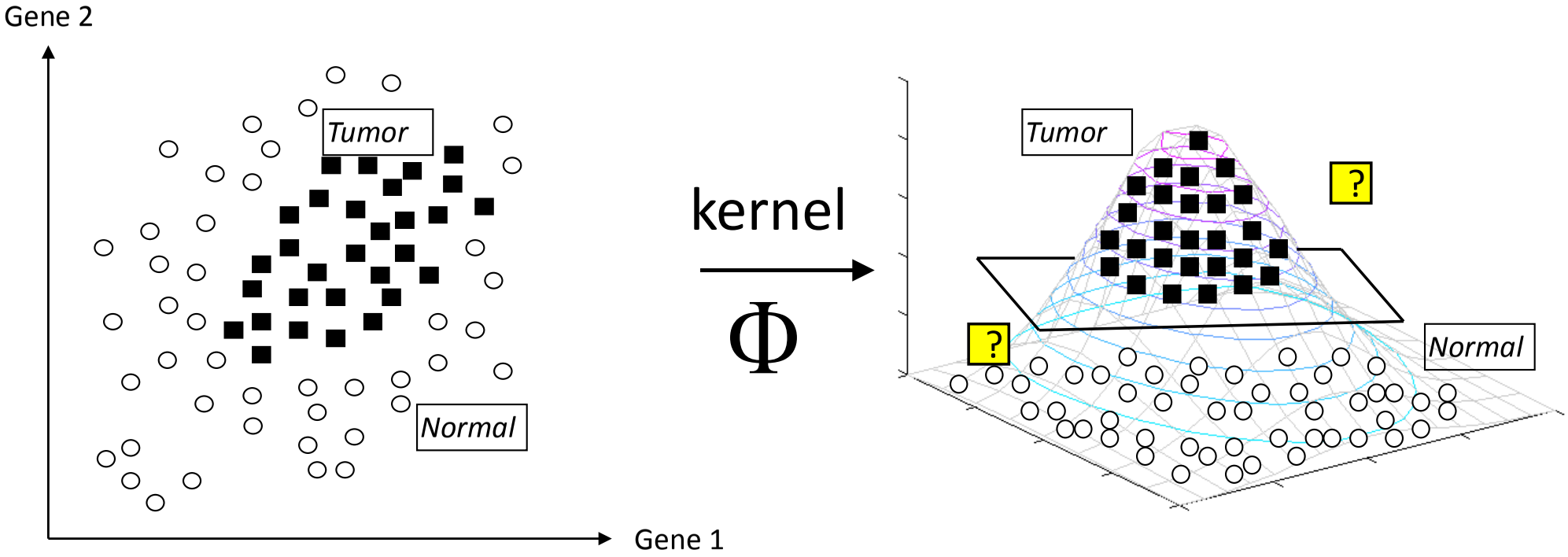
C=0.15



C=0.1

- When *C* is very large, the soft-margin SVM is equivalent to hard-margin SVM;
- When *C* is very small, we admit misclassifications in the training data at the expense of having w-vector with small norm;
- *C* has to be selected for the distribution at hand as it will be discussed later in this tutorial.

Case 3: Not linearly separable data; Kernel trick



Data is not linearly separable in the input space

Data is linearly separable in the feature space obtained by a kernel

$$\Phi : \mathbf{R}^N \rightarrow \mathbf{H}$$

Kernel trick

Original data $\vec{x}$ (in input space)

$$f(x) = sign(\vec{w} \cdot \vec{x} + b)$$

$$\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$$

Data in a higher dimensional feature space $\Phi(\vec{x})$

$$f(x) = sign(\vec{w} \cdot \Phi(\vec{x}) + b)$$

$$\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \Phi(\vec{x}_i)$$

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i \Phi(\vec{x}_i) \cdot \Phi(\vec{x}) + b)$$

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b)$$

Therefore, we do not need to know Φ explicitly, we just need to define function $K(\cdot, \cdot)$: $\mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$.

Not every function $\mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ can be a valid kernel; it has to satisfy so-called Mercer conditions. Otherwise, the underlying quadratic program may not be solvable.

Popular kernels

A kernel is a dot product in *some* feature space:

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$$

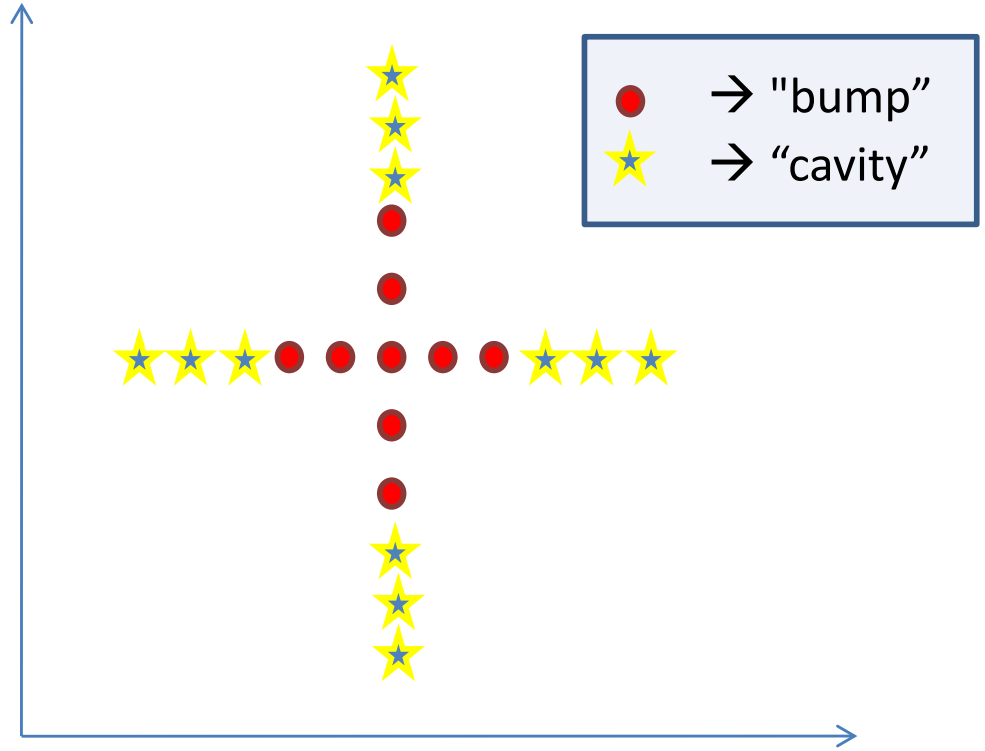
Examples:

$K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$	Linear kernel
$K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \left\| \vec{x}_i - \vec{x}_j \right\|^2)$	Gaussian kernel
$K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \left\| \vec{x}_i - \vec{x}_j \right\|)$	Exponential kernel
$K(\vec{x}_i, \vec{x}_j) = (p + \vec{x}_i \cdot \vec{x}_j)^q$	Polynomial kernel
$K(\vec{x}_i, \vec{x}_j) = (p + \vec{x}_i \cdot \vec{x}_j)^q \exp(-\gamma \left\| \vec{x}_i - \vec{x}_j \right\|^2)$	Hybrid kernel
$K(\vec{x}_i, \vec{x}_j) = \tanh(k\vec{x}_i \cdot \vec{x}_j - \delta)$	Sigmoidal

Understanding the Gaussian kernel

Consider Gaussian kernel: $K(\vec{x}, \vec{x}_j) = \exp(-\gamma \left\| \vec{x} - \vec{x}_j \right\|^2)$

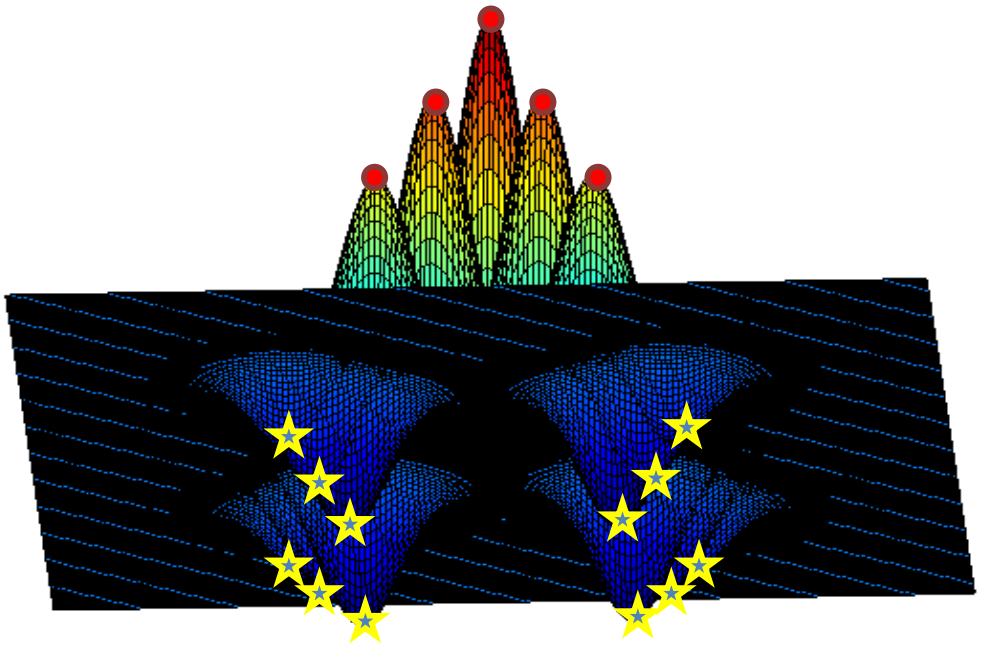
Geometrically, this is a "bump" or "cavity" centered at the training data point $\vec{x}_j$:

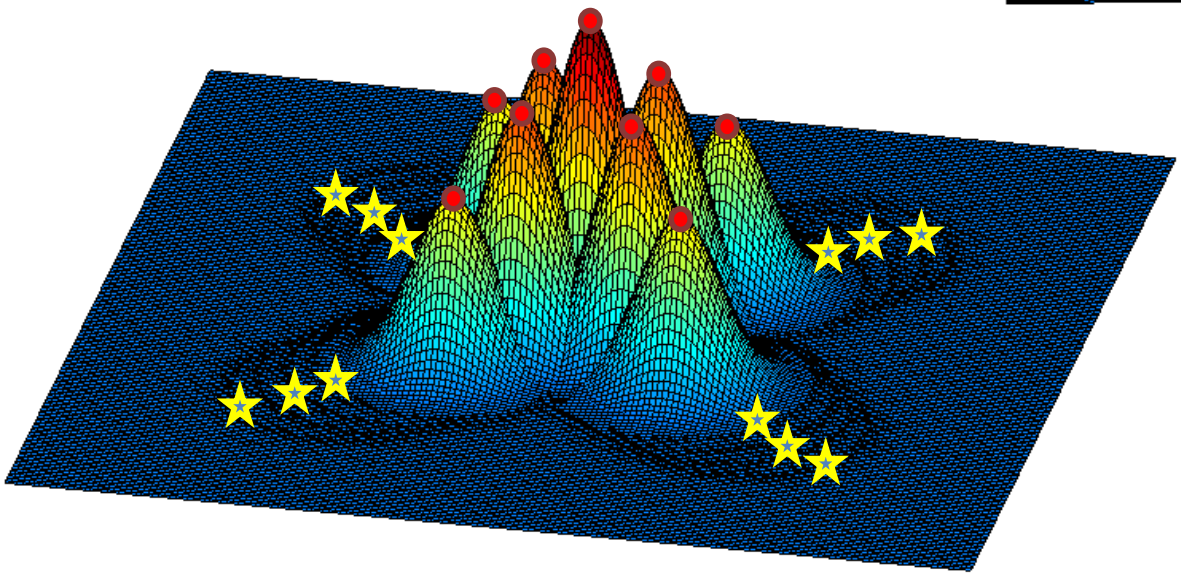


The resulting mapping function is a **combination** of bumps and cavities.

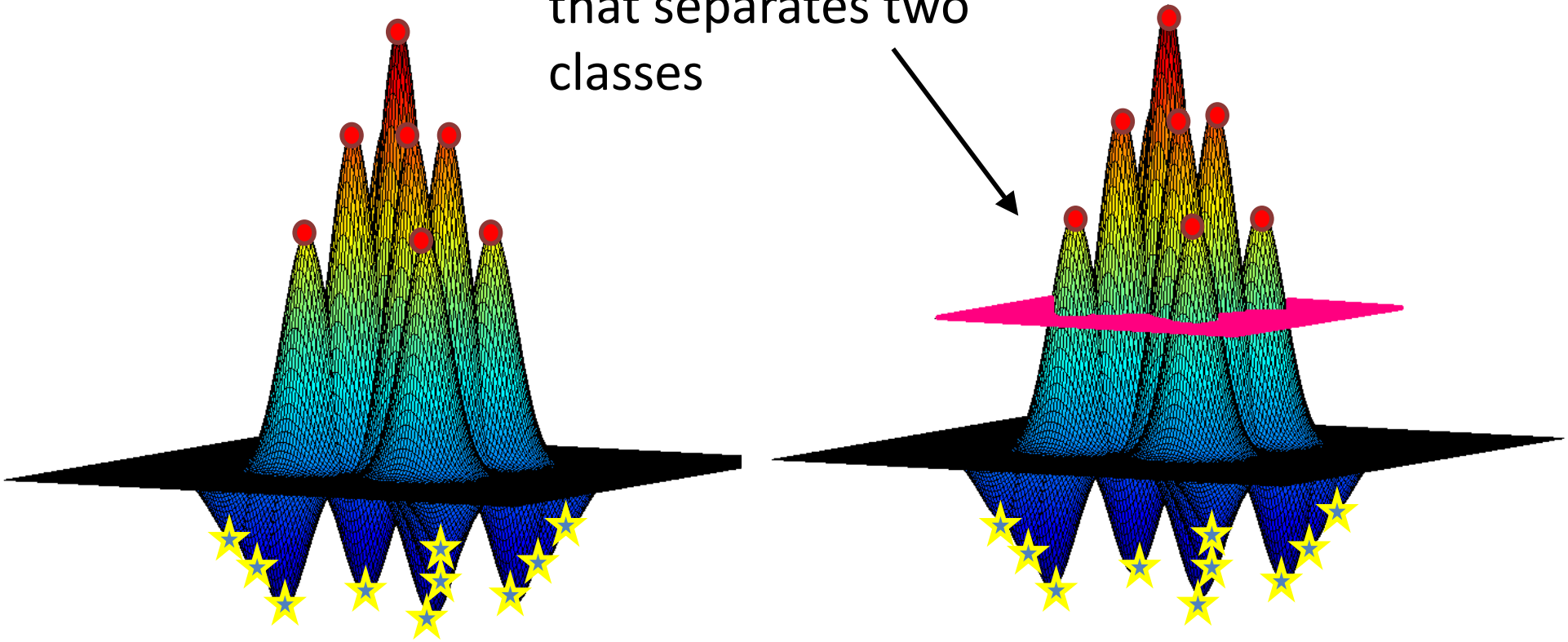
Understanding the Gaussian kernel

Several more views of the data is mapped to the feature space by Gaussian kernel





Understanding the Gaussian kernel



Understanding the polynomial kernel

Consider polynomial kernel: $K(\vec{x}_i, \vec{x}_j) = (1 + \vec{x}_i \cdot \vec{x}_j)^3$

Assume that we are dealing with 2-dimensional data (i.e., in $\mathbb{R}^2$). Where will this kernel map the data?

2-dimensional space

$x_{(1)} \quad x_{(2)}$

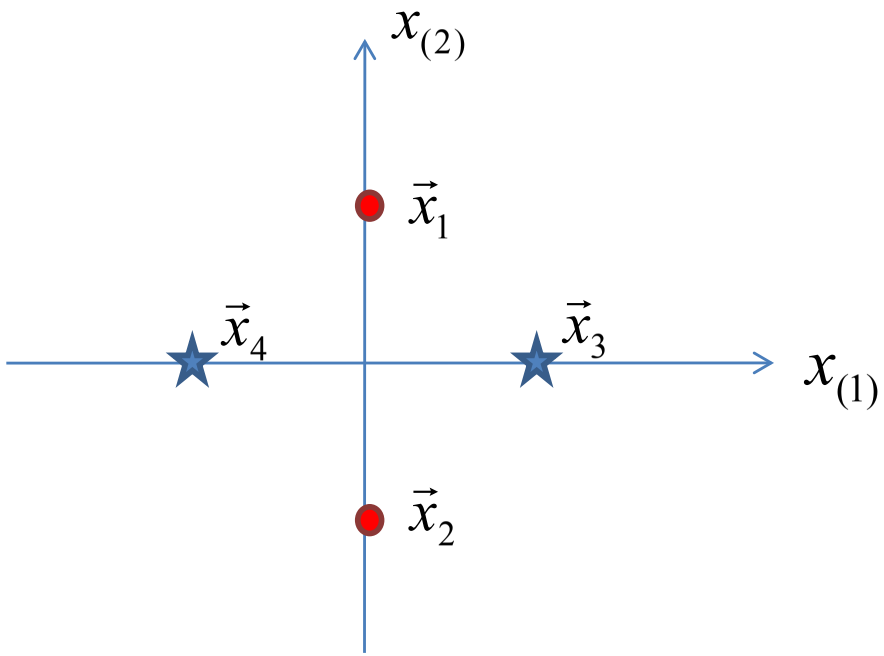
kernel

▲

10-dimensional space

$1 \quad x_{(1)} \quad x_{(2)} \quad x_{(1)}^2 \quad x_{(2)}^2 \quad x_{(1)}x_{(2)} \quad x_{(1)}^3 \quad x_{(2)}^3 \quad x_{(1)}x_{(2)}^2 \quad x_{(1)}^2 x_{(2)}$

Example of benefits of using a kernel



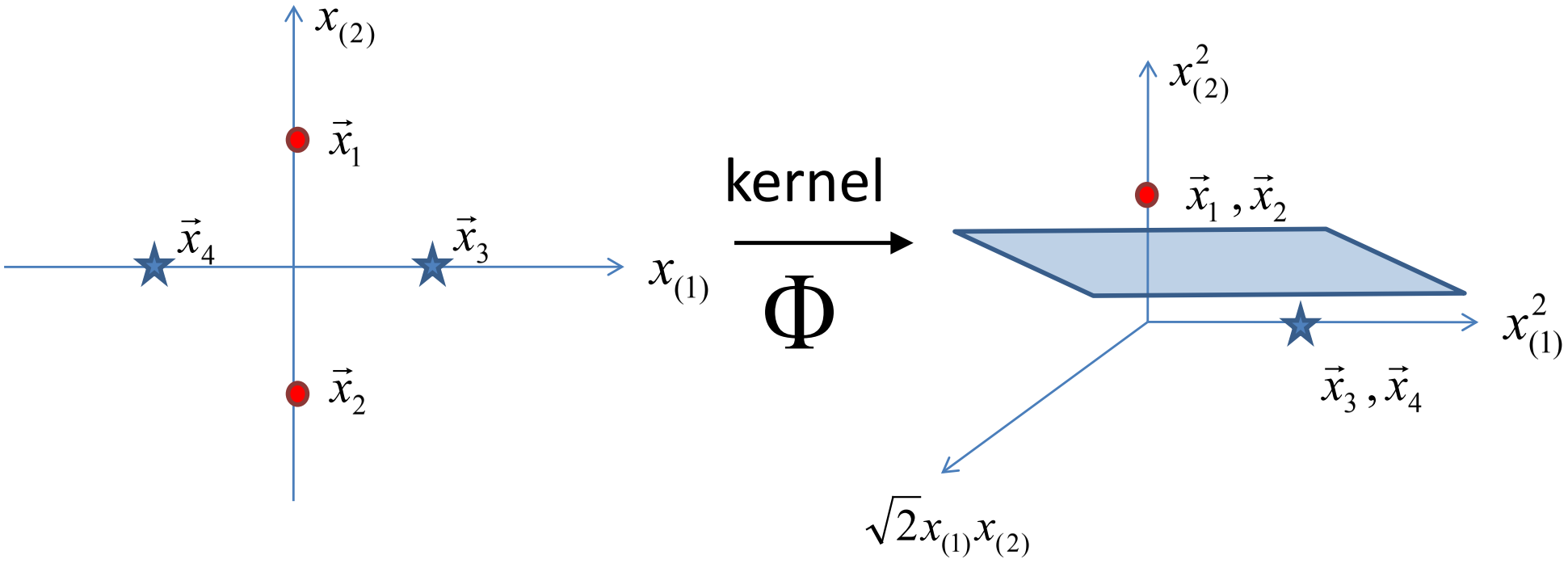
- Data is not linearly separable in the input space ($\mathbb{R}^2$).
- Apply kernel $K(\vec{x},\vec{z}) = (\vec{x}\cdot\vec{z})^2$ to map data to a higher dimensional space (3-dimensional) where it is linearly separable.

$$K(\vec{x},\vec{z}) = (\vec{x}\cdot\vec{z})^2 = \left[\begin{pmatrix} x_{(1)} \\ x_{(2)} \end{pmatrix}\cdot\begin{pmatrix} z_{(1)} \\ z_{(2)} \end{pmatrix}\right]^2 = \left[x_{(1)}z_{(1)} + x_{(2)}z_{(2)}\right]^2 =$$

$$= x_{(1)}^2 z_{(1)}^2 + 2x_{(1)}z_{(1)}x_{(2)}z_{(2)} + x_{(2)}^2 z_{(2)}^2 = \begin{pmatrix} x_{(1)}^2 \\ \sqrt{2}x_{(1)}x_{(2)} \\ x_{(2)}^2 \end{pmatrix}\cdot\begin{pmatrix} z_{(1)}^2 \\ \sqrt{2}z_{(1)}z_{(2)} \\ z_{(2)}^2 \end{pmatrix} = \Phi(\vec{x})\cdot\Phi(\vec{z})$$

Example of benefits of using a kernel

Therefore, the explicit mapping is $\Phi(\vec{x}) = \begin{pmatrix} x_{(1)}^2 \\ \sqrt{2}x_{(1)}x_{(2)} \\ x_{(2)}^2 \end{pmatrix}$



Comparison with methods from classical statistics & regression

- Need ≥ 5 samples for each parameter of the regression model to be estimated:

Number of variables	Polynomial degree	Number of parameters	Required sample
2	3	10	50
10	3	286	1,430
10	5	3,003	15,015
100	3	176,851	884,255
100	5	96,560,646	482,803,230

- SVMs do not have such requirement & often require much less sample than the number of variables, even when a high-degree polynomial kernel is used.

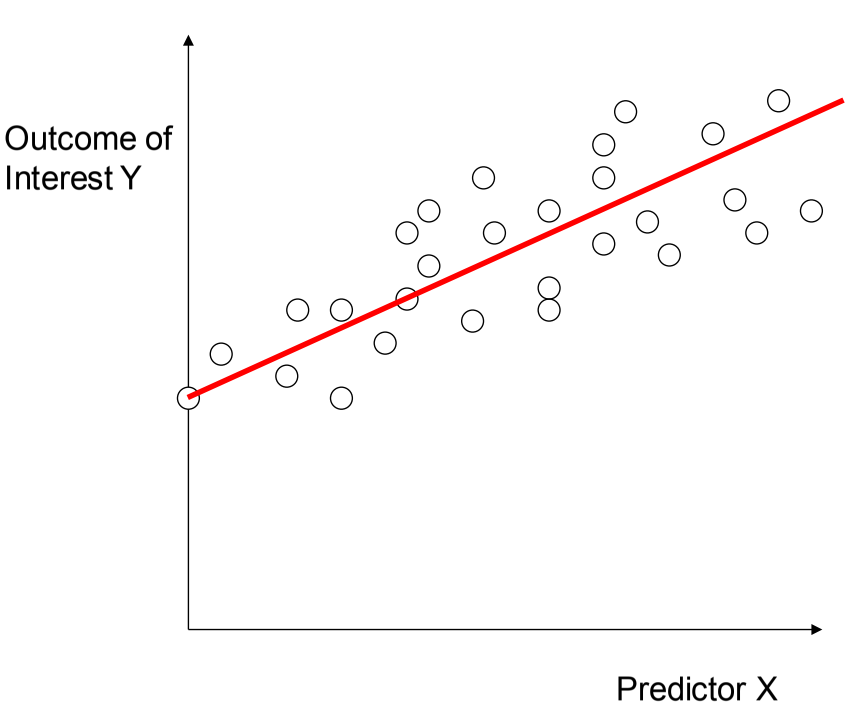
Basic principles of statistical machine learning

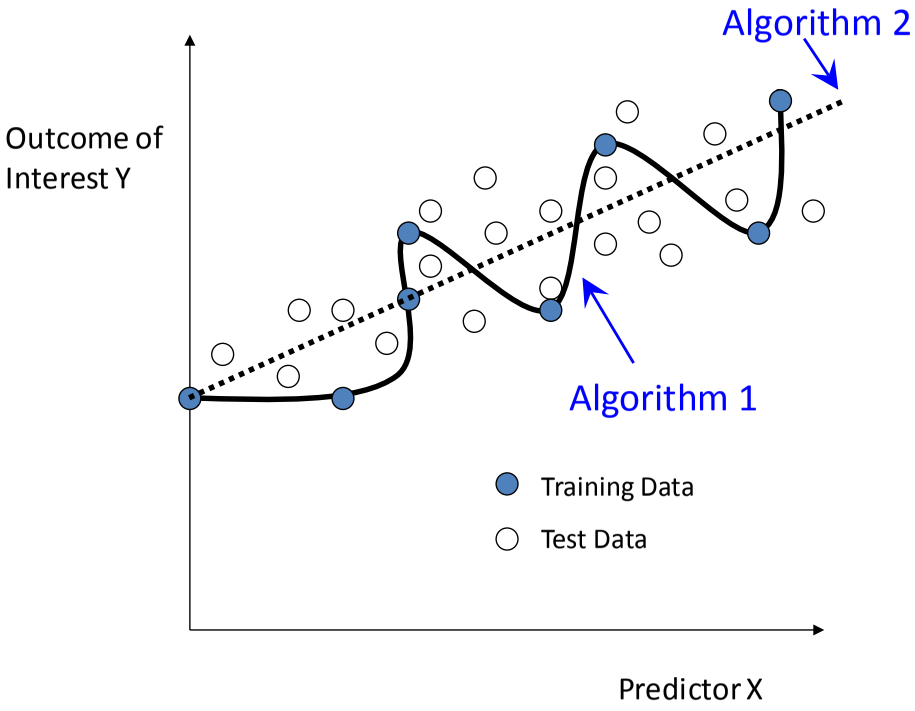
Generalization and overfitting

- **Generalization:** A classifier or a regression algorithm learns to correctly predict output from given inputs not only in previously seen samples but also in previously unseen samples.
- **Overfitting:** A classifier or a regression algorithm learns to correctly predict output from given inputs in previously seen samples but fails to do so in previously unseen samples.
- **Overfitting → Poor generalization**.

Example of overfitting and generalization

There is a linear relationship between predictor and outcome (plus some Gaussian noise).





- Algorithm 1 learned non-reproducible peculiarities of the specific sample available for learning but did not learn the general characteristics of the function that generated the data. Thus, it is overfitted and has poor generalization.
- Algorithm 2 learned general characteristics of the function that produced the data. Thus, it generalizes.

"Loss + penalty" paradigm for learning to avoid overfitting and ensure generalization

- Many statistical learning algorithms (including SVMs) search for a decision function by solving the following optimization problem:

Minimize (*Loss* + λ *Penalty*)

- ***Loss*** measures error of fitting the data
- ***Penalty*** penalizes complexity of the learned function
- **λ** is regularization parameter that balances ***Loss*** and ***Penalty***

SVMs in "loss + penalty" form

SVMs build the following classifiers: $f(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b)$

Consider soft-margin linear SVM formulation:

Find $\vec{w}$ and b that

Minimize $\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N} \xi_i$ subject to $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i$ for $i = 1,\ldots,N$

This can also be stated as:

Find $\vec{w}$ and b that

Minimize $\sum_{i=1}^{N} [1 - y_i f(\vec{x}_i)]_+ + \lambda \|\vec{w}\|_2^2$

Loss (*"hinge loss"*) *Penalty*

(in fact, one can show that $\lambda = 1/(2C)$).

Meaning of SVM loss function

Consider loss function: $\sum_{i=1}^{N}[1-y_i f(\vec{x}_i)]_+$

- Recall that $[\ldots]_+$ indicates the positive part
- For a given sample/patient *i*, the loss is non-zero if $1-y_i f(\vec{x}_i)>0$
- In other words, $y_i f(\vec{x}_i)<1$
- Since $y_i=\{-1,+1\}$, this means that the loss is non-zero if

  $f(\vec{x}_i)<1$ for $y_i=+1$

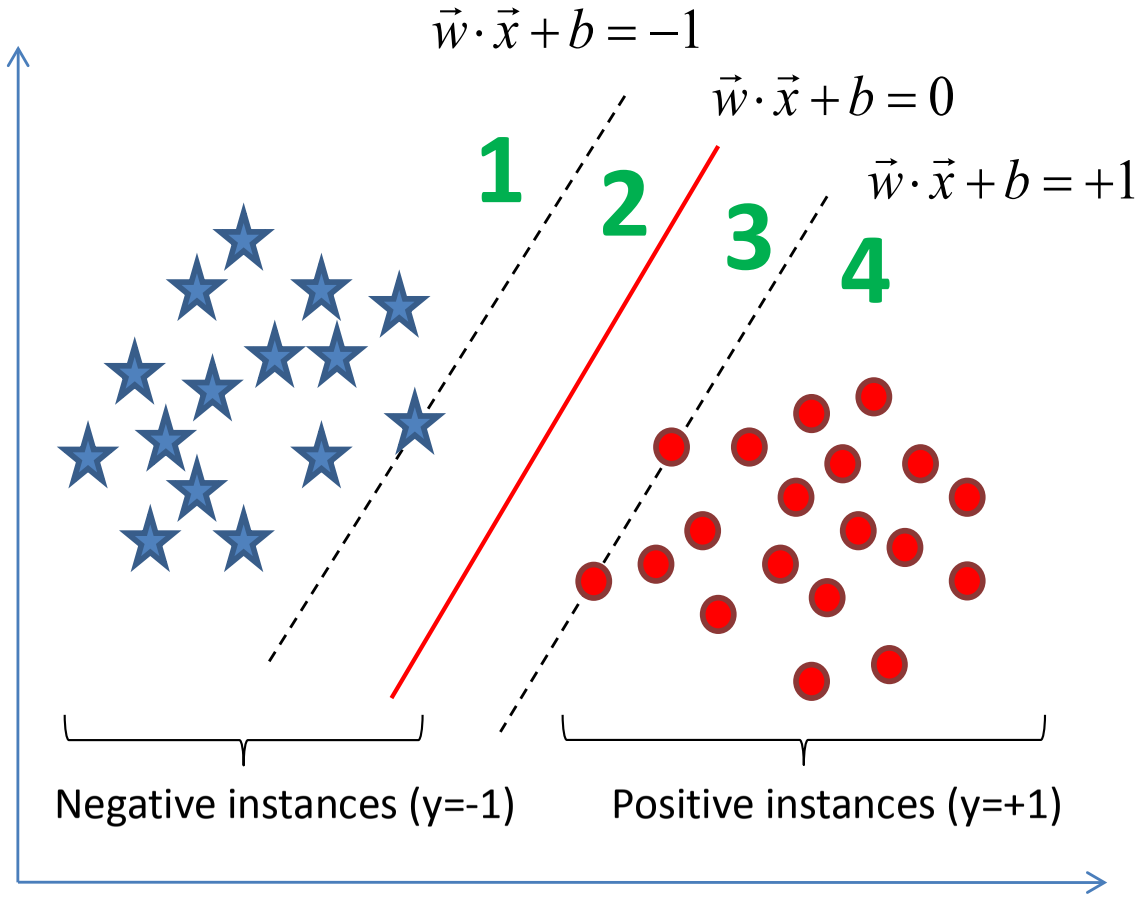
  $f(\vec{x}_i)>-1$ for y_i= -1
- In other words, the loss is non-zero if

  $\vec{w}\cdot\vec{x}_i+b<1$ for $y_i=+1$

  $\vec{w}\cdot\vec{x}_i+b>-1$ for y_i= -1

Meaning of SVM loss function

- If the instance is negative, it is penalized only in regions 2,3,4
- If the instance is positive, it is penalized only in regions 1,2,3



Flexibility of "loss + penalty" framework

Minimize (*Loss* + λ *Penalty*)

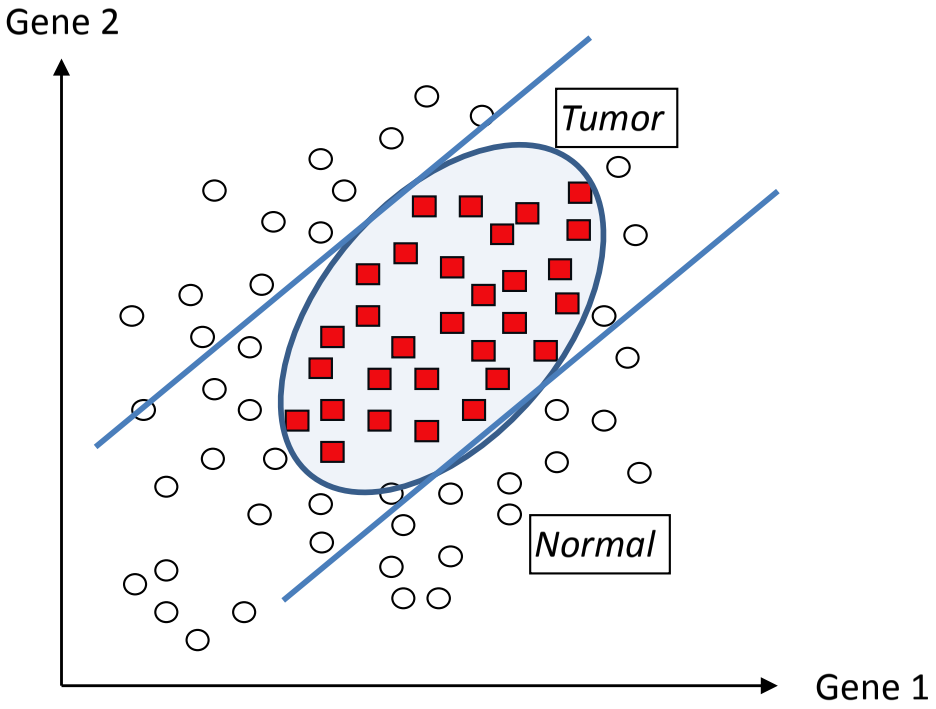
Loss function	Penalty function	Resulting algorithm
Hinge loss: $\sum_{i=1}^{N}[1-y_i f(\vec{x}_i)]_+$	$\lambda\|\vec{w}\|_2^2$	SVMs
Mean squared error: $\sum_{i=1}^{N}(y_i - f(\vec{x}_i))^2$	$\lambda\|\vec{w}\|_2^2$	Ridge regression
Mean squared error: $\sum_{i=1}^{N}(y_i - f(\vec{x}_i))^2$	$\lambda\|\vec{w}\|_1$	Lasso
Mean squared error: $\sum_{i=1}^{N}(y_i - f(\vec{x}_i))^2$	$\lambda_1\|\vec{w}\|_1 + \lambda_2\|\vec{w}\|_2^2$	Elastic net
Hinge loss: $\sum_{i=1}^{N}[1-y_i f(\vec{x}_i)]_+$	$\lambda\|\vec{w}\|_1$	1-norm SVM

Part 2

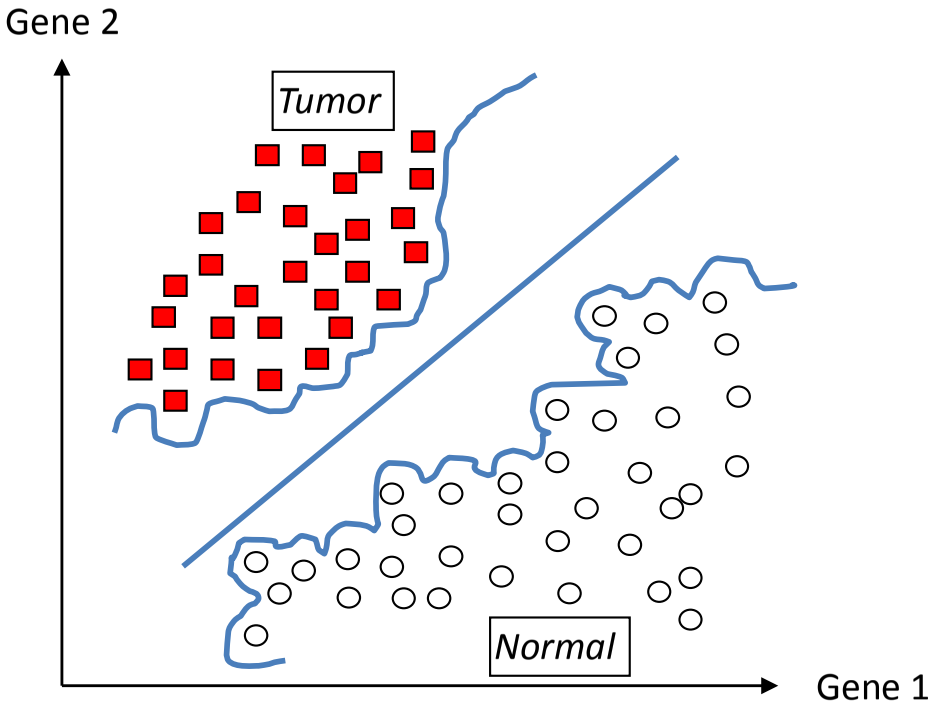
- Model selection for SVMs
- Extensions to the basic SVM model:
  1. SVMs for multicategory data
  2. Support vector regression
  3. Novelty detection with SVM-based methods
  4. Support vector clustering
  5. SVM-based variable selection
  6. Computing posterior class probabilities for SVM classifiers

Model selection for SVMs

Need for model selection for SVMs



- It is impossible to find a linear SVM classifier that separates tumors from normals!
- Need a non-linear SVM classifier, e.g. SVM with polynomial kernel of degree 2 solves this problem without errors.



- We should not apply a non-linear SVM classifier while we can perfectly solve this problem using a linear SVM classifier!

A data-driven approach for model selection for SVMs

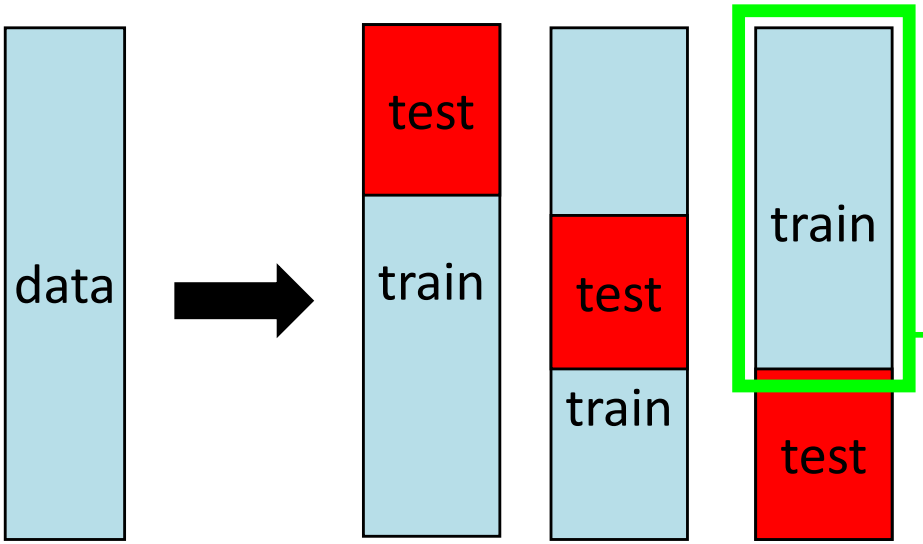
- Do not know *a priori* what type of SVM kernel and what kernel parameter(s) to use for a given dataset?
- Need to examine various combinations of parameters, e.g. consider searching the following grid:

	Polynomial degree *d*				
Parameter *C*	(0.1, 1)	(1, 1)	(10, 1)	(100, 1)	(1000, 1)
	(0.1, 2)	(1, 2)	(10, 2)	(100, 2)	(1000, 2)
	(0.1, 3)	(1, 3)	(10, 3)	(100, 3)	(1000, 3)
	(0.1, 4)	(1, 4)	(10, 4)	(100, 4)	(1000, 4)
	(0.1, 5)	(1, 5)	(10, 5)	(100, 5)	(1000, 5)

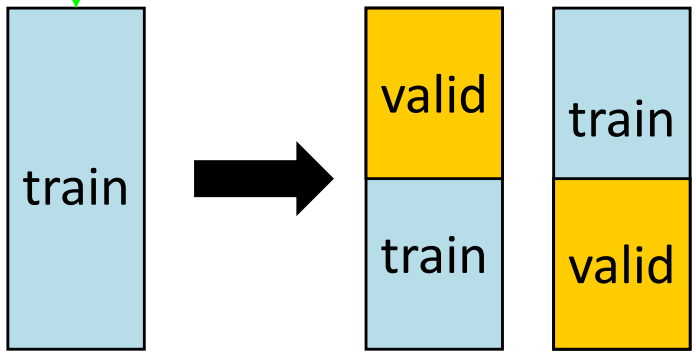
- How to search this grid while producing an unbiased estimate of classification performance?

Nested cross-validation

Recall the main idea of cross-validation:



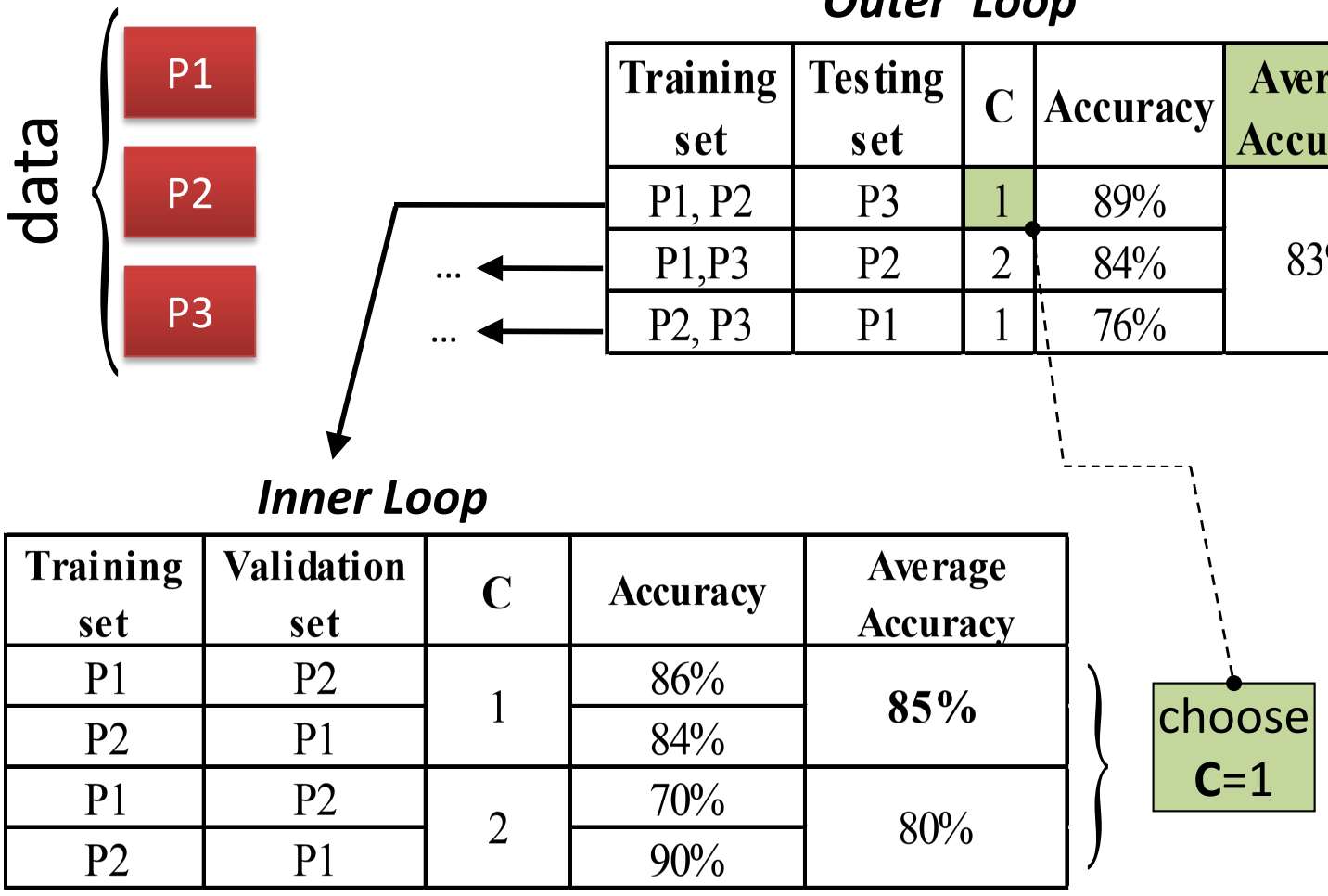
What combination of SVM parameters to apply on training data?



Perform “grid search” using another nested loop of cross-validation.

Example of nested cross-validation

Consider that we use 3-fold cross-validation and we want to optimize parameter C that takes values “1” and “2”.

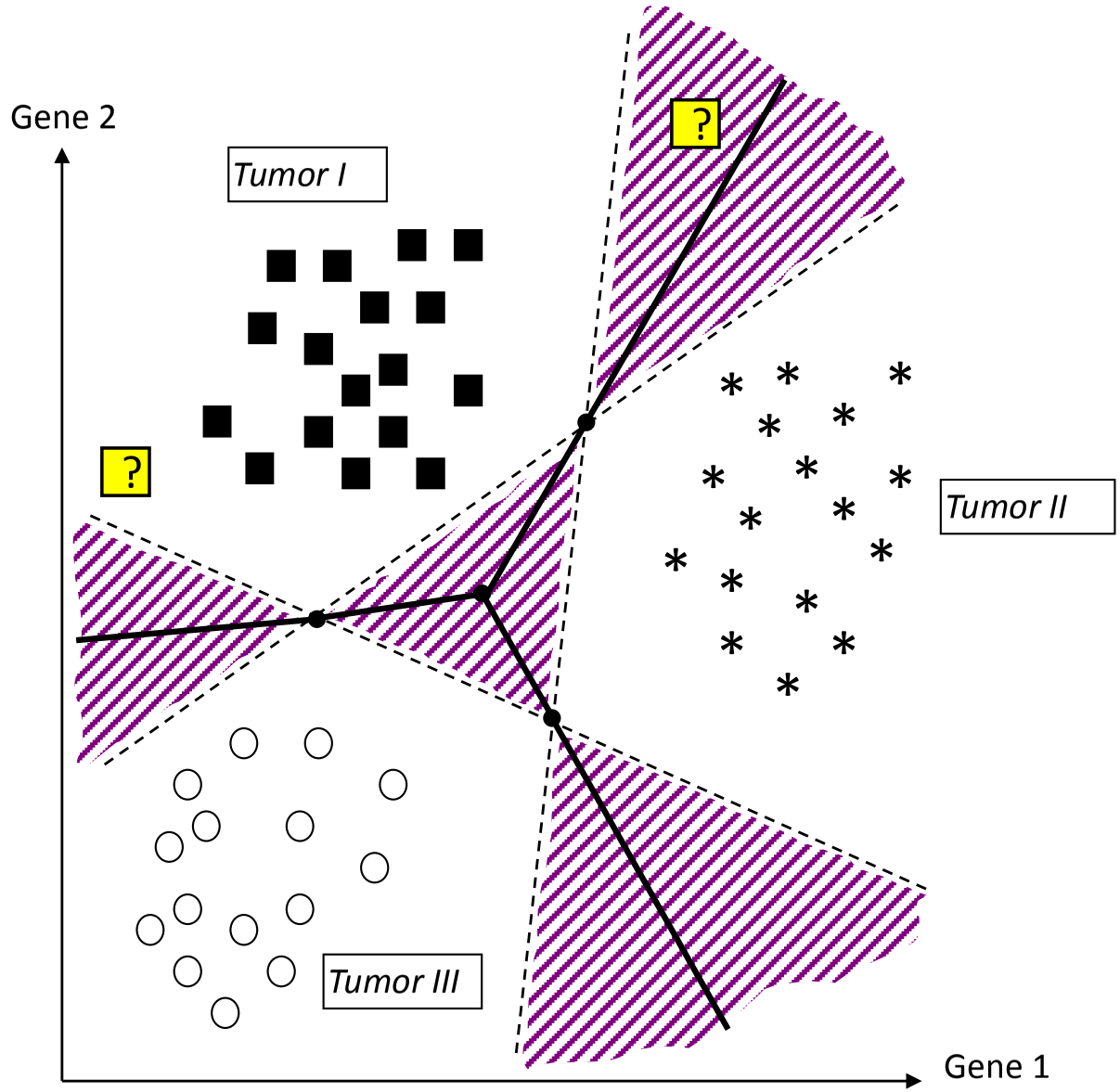


On use of cross-validation

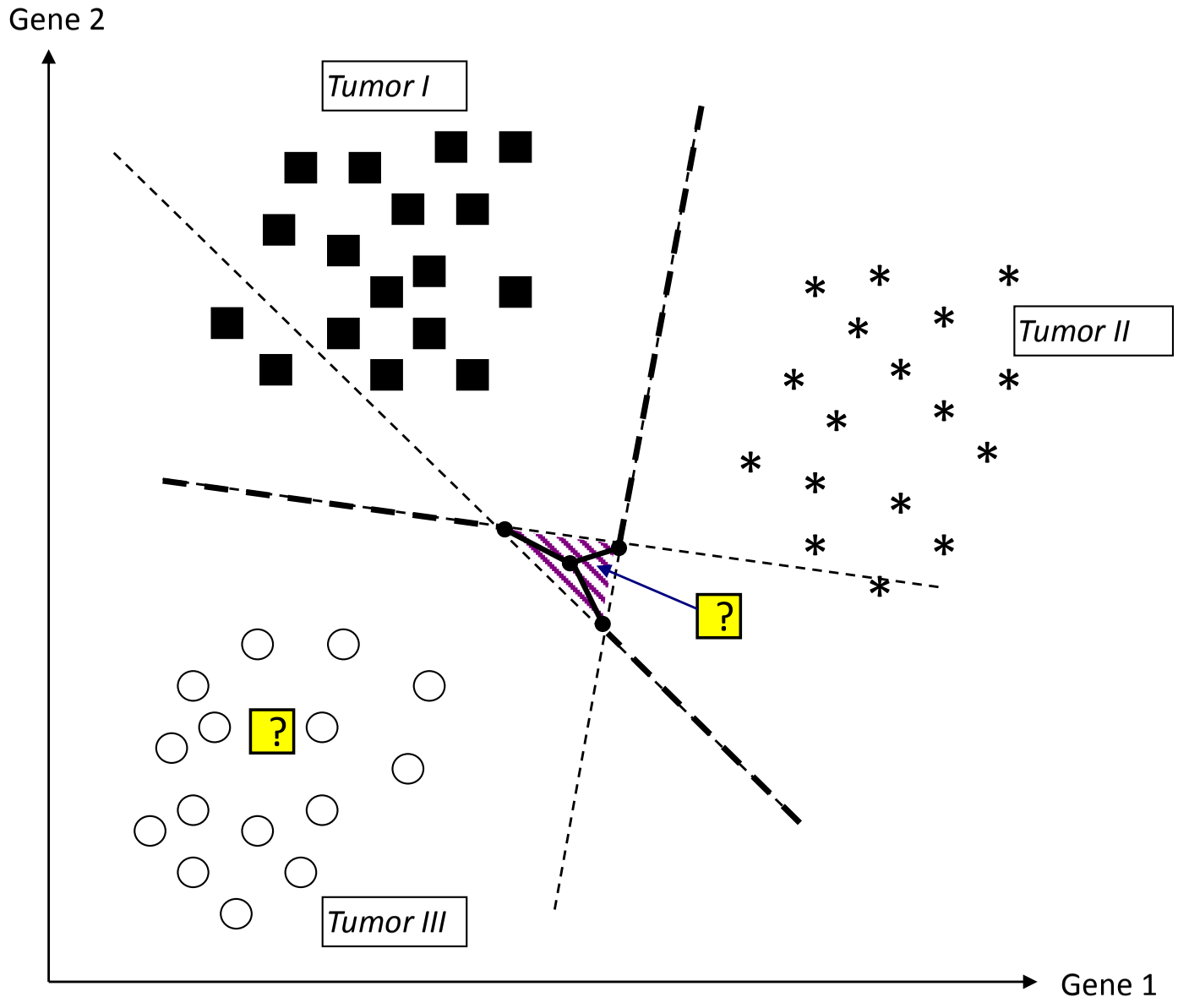
- Empirically we found that cross-validation works well for model selection for SVMs in many problem domains;
- Many other approaches that can be used for model selection for SVMs exist, e.g.:
  - Generalized cross-validation
  - Bayesian information criterion (BIC)
  - Minimum description length (MDL)
  - Vapnik-Chernovenkis (VC) dimension
  - Bootstrap

SVMs for multicategory data

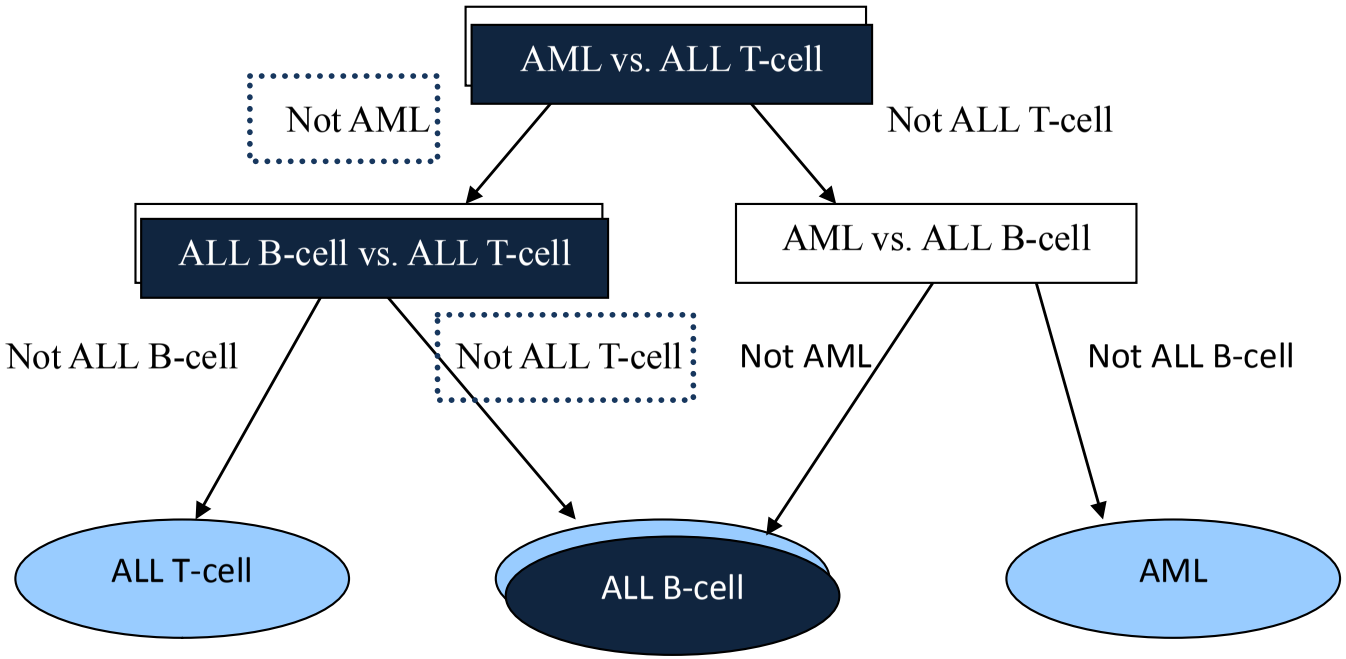
One-versus-rest multicategory SVM method



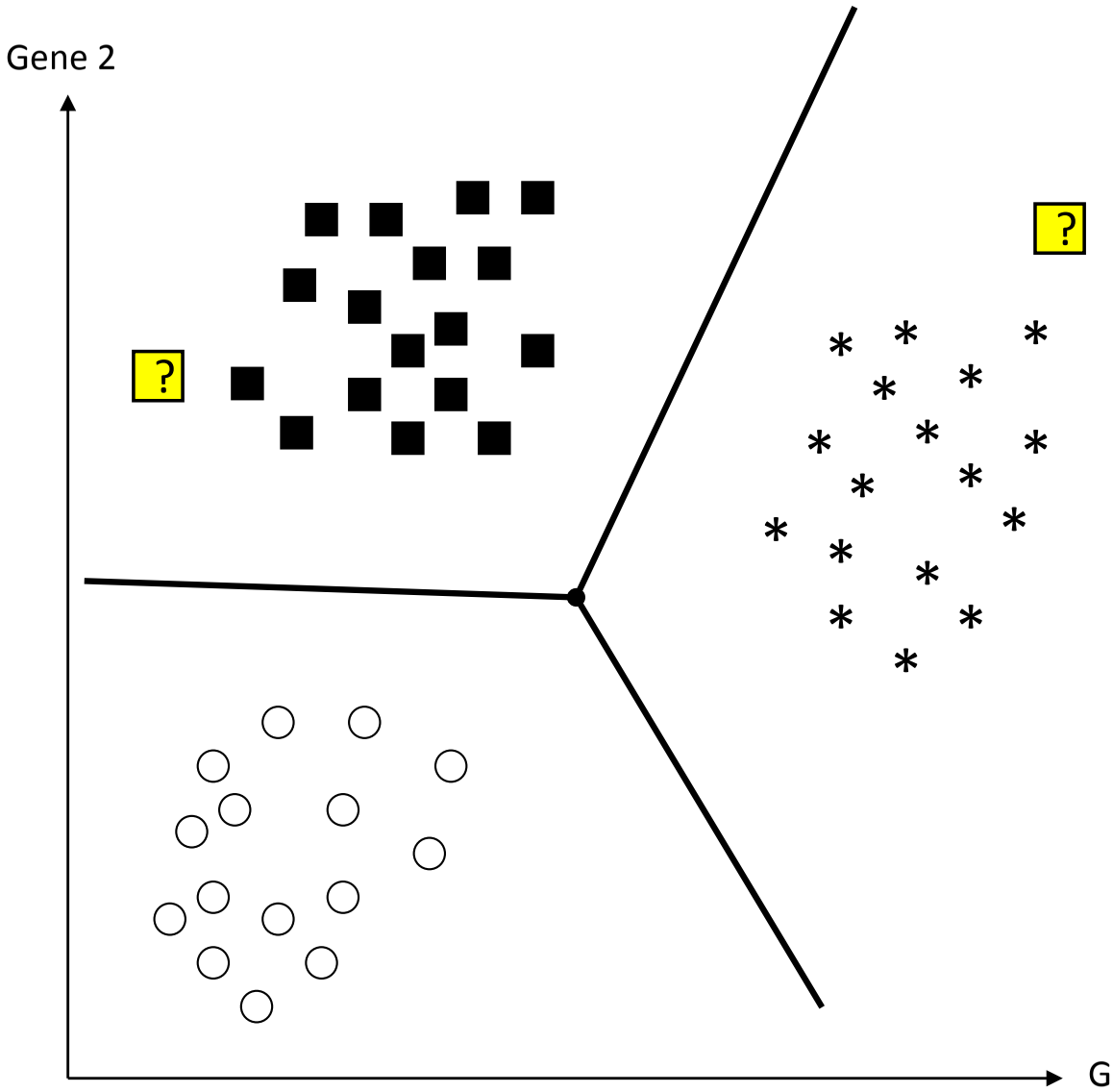
One-versus-one multicategory SVM method



DAGSVM multicategory SVM method



SVM multicategory methods by Weston and Watkins and by Crammer and Singer

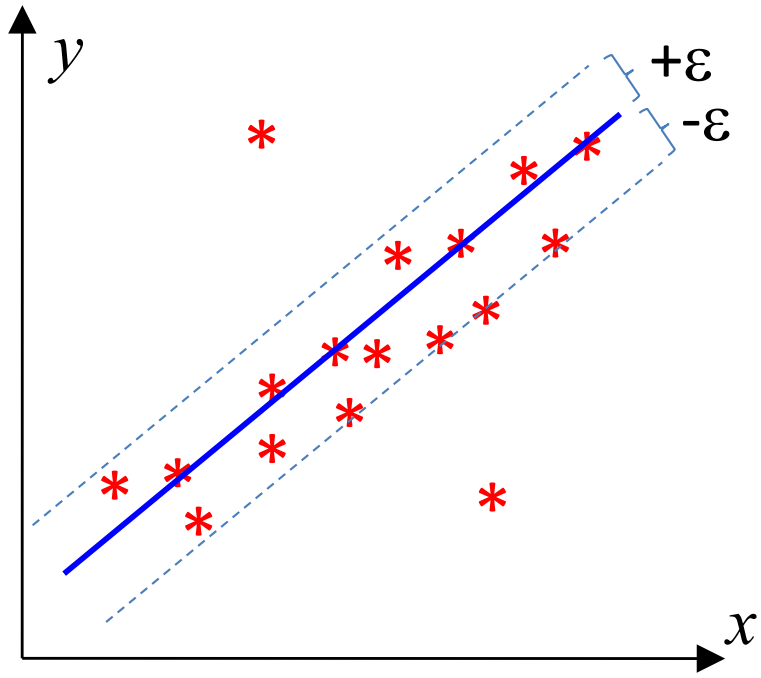


Support vector regression

ε-Support vector regression (ε-SVR)

Given training data: $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \in R^n$

$y_1, y_2, \ldots, y_N \in R$



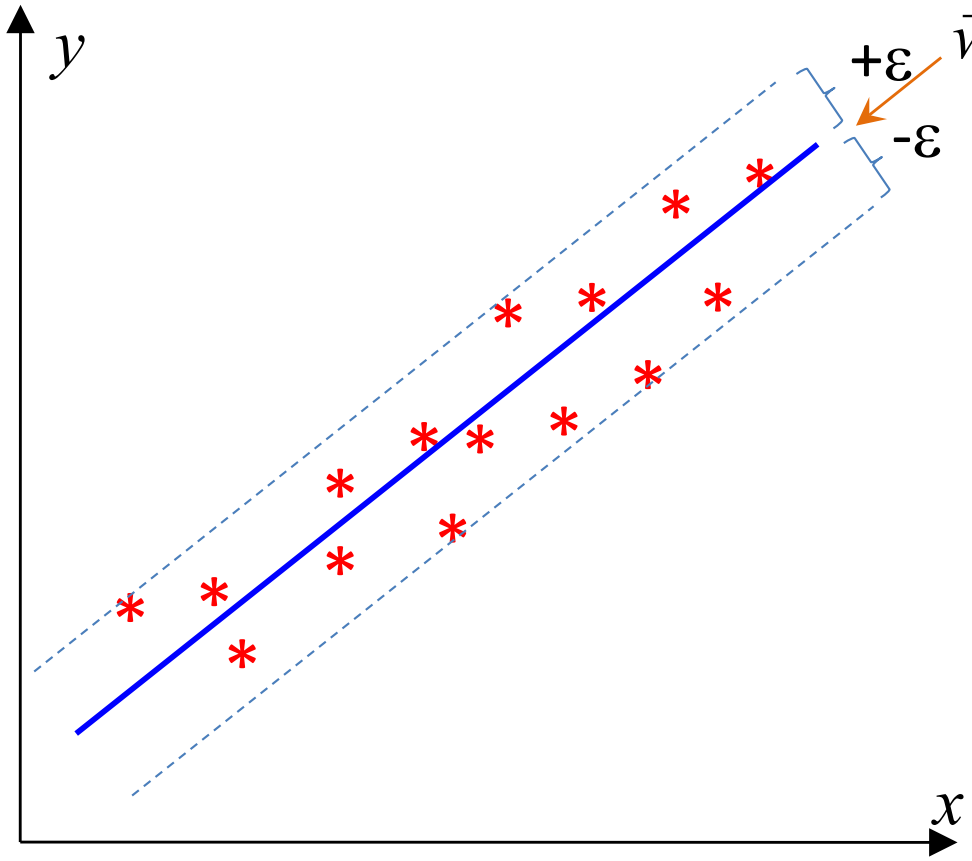
Main idea:

Find a function $f(\vec{x}) = \vec{w} \cdot \vec{x} + b$ that approximates $y_1, \ldots, y_N$:

- it has at most ε derivation from the true values y_i
- it is as "flat" as possible (to avoid overfitting)

E.g., build a model to predict survival of cancer patients that can admit a one month error (= ε).

Formulation of "hard-margin" ε-SVR



Find $f(\vec{x})=\vec{w}\cdot\vec{x}+b$

by minimizing $\frac{1}{2}\|\vec{w}\|^2$ subject to constraints:

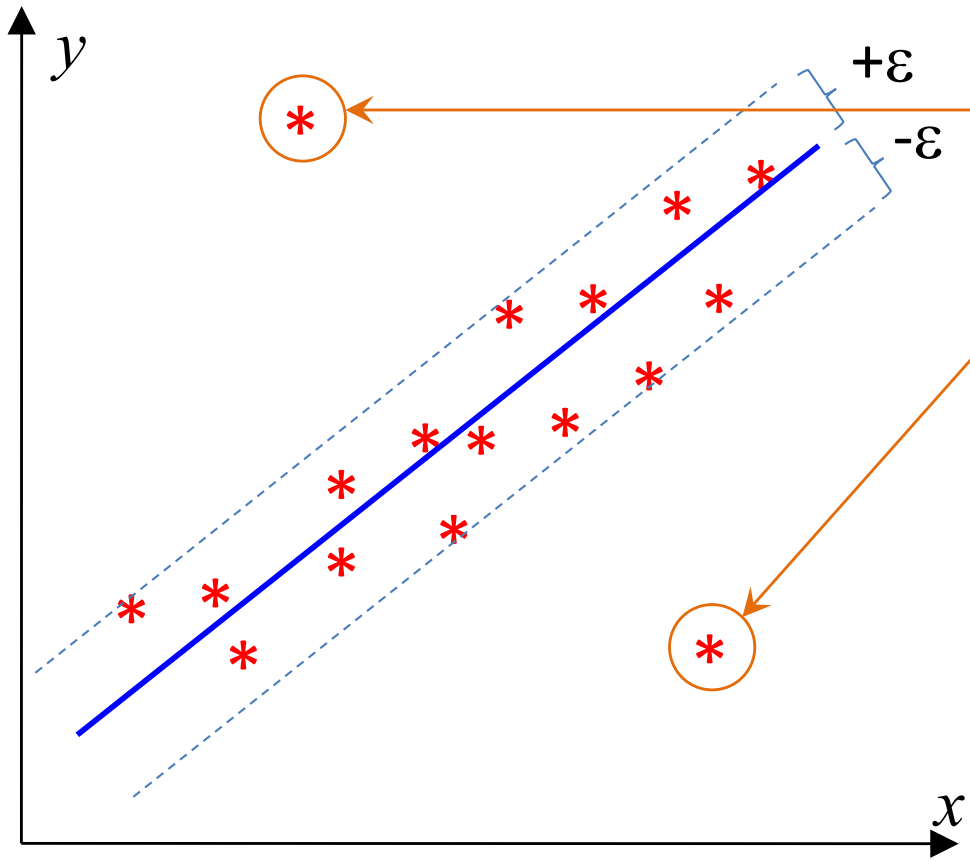
$$y_i-(\vec{w}\cdot\vec{x}+b)\le\varepsilon$$

$$y_i-(\vec{w}\cdot\vec{x}+b)\ge-\varepsilon$$

for $i=1,\ldots,N.$

I.e., difference between y_i and the fitted function should be smaller than ε and larger than -ε ⇔ all points y_i should be in the "ε-ribbon" around the fitted function.

Formulation of "soft-margin" ε-SVR

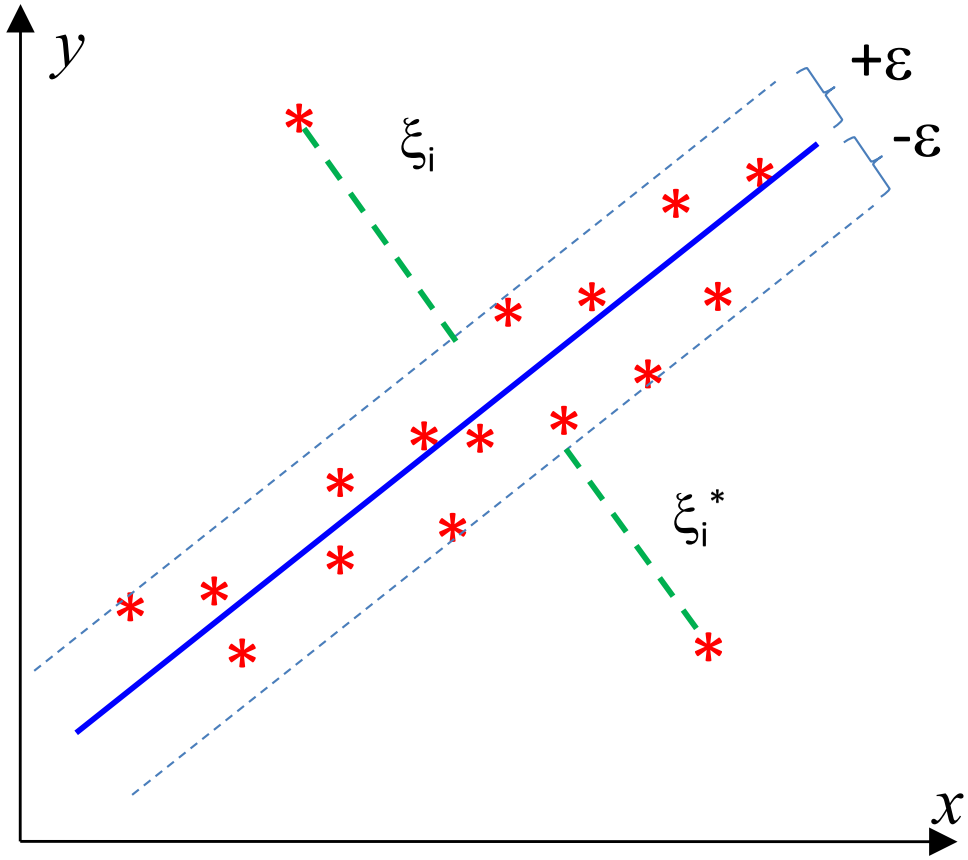


If we have points like this (e.g., outliers or noise) we can either:

a) increase ε to ensure that these points are within the new ε-ribbon, or

b) assign a penalty ("slack" variable) to each of this points (as was done for "soft-margin" SVMs)

Formulation of "soft-margin" ε-SVR



Find $f(\vec{x}) = \vec{w} \cdot \vec{x} + b$

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

subject to constraints:

$$y_i - (\vec{w} \cdot \vec{x} + b) \le \varepsilon + \xi_i$$

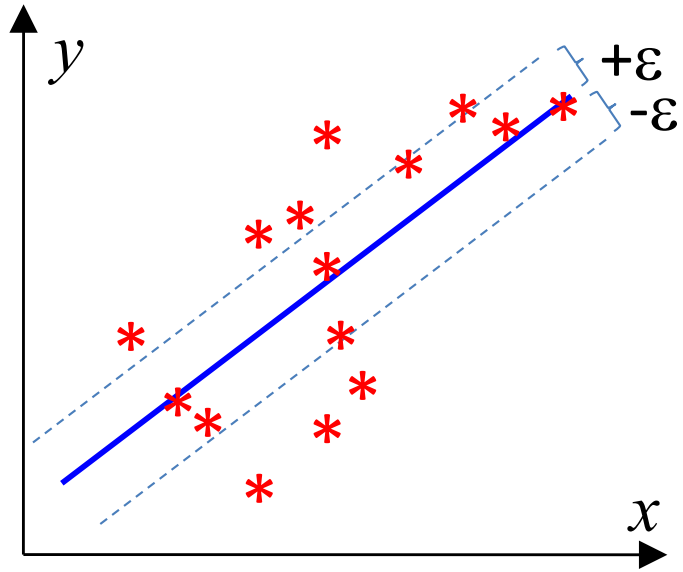
$$y_i - (\vec{w} \cdot \vec{x} + b) \ge -\varepsilon - \xi_i^*$$

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

for $i = 1,\ldots,N.$

Notice that only points outside ε-ribbon are penalized!

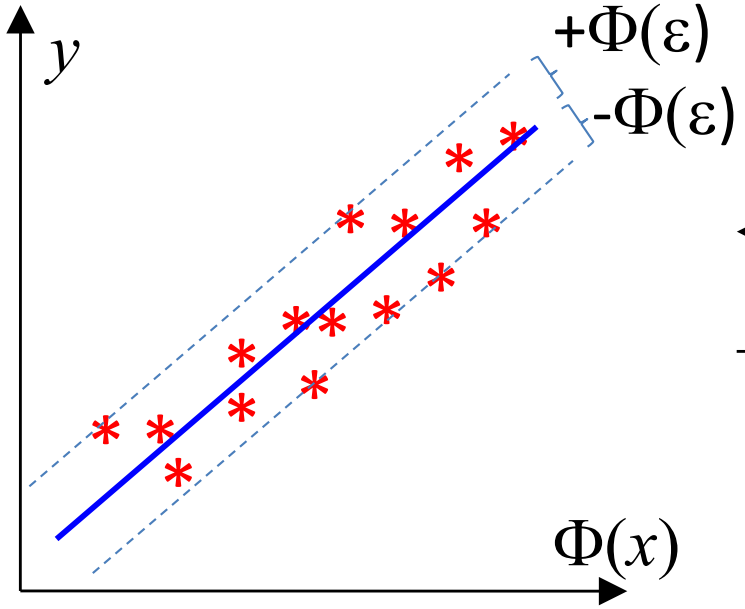
Nonlinear ε-SVR



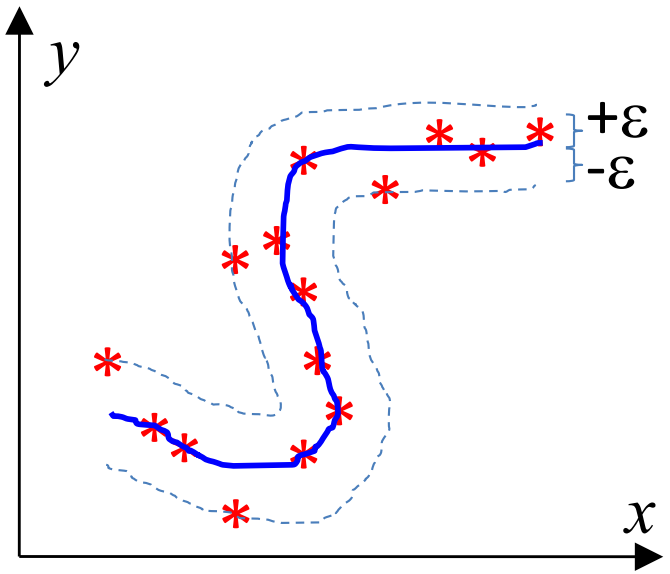
Cannot approximate well this function with small ε!

kernel

Φ



Φ^{-1}



ε-Support vector regression in "loss + penalty" form

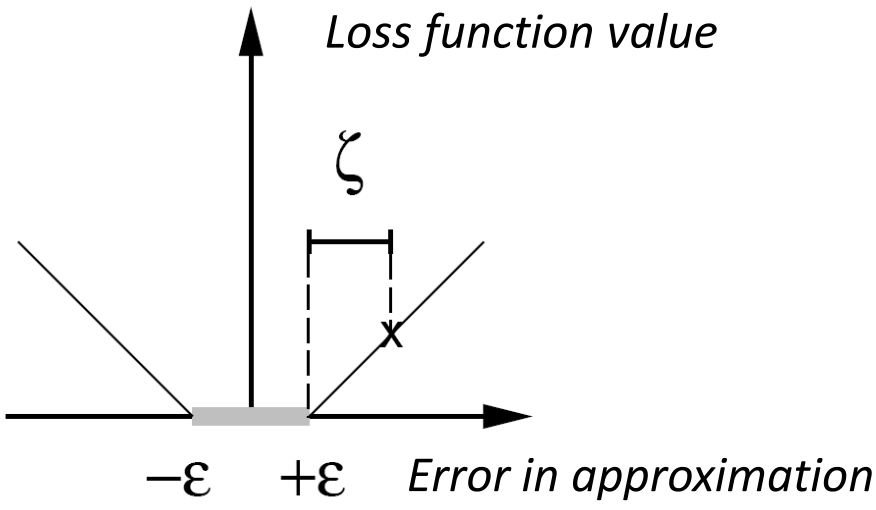
Build decision function of the form: $f(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Find $\vec{w}$ and b that

Minimize $\sum_{i=1}^{N} \max(0, |\, y_i - f(\vec{x}_i)\,| - \varepsilon) + \lambda \|\vec{w}\|_2^2$

Loss *Penalty*

("linear ε-insensitive loss*")*

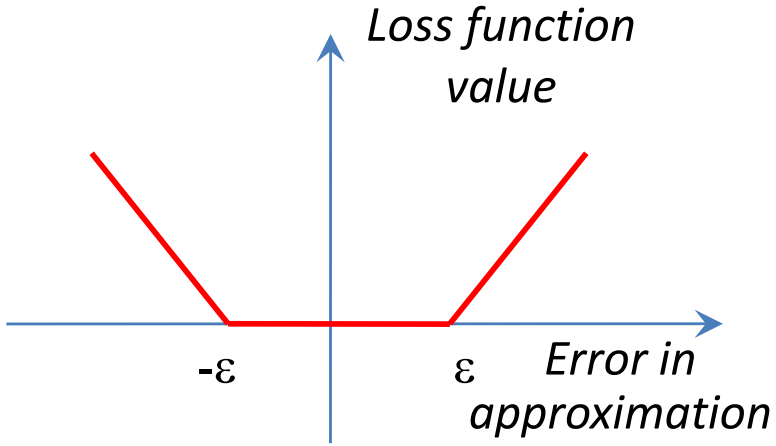


Comparing ε-SVR with popular regression methods

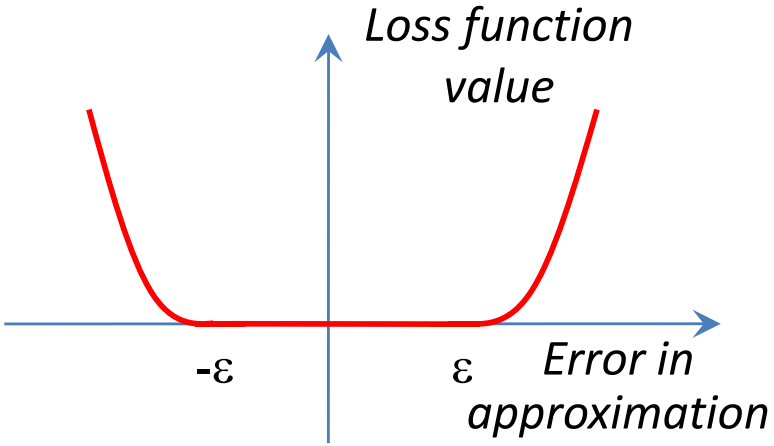
Loss function	Penalty function	Resulting algorithm
Linear ε-insensitive loss: $\sum_{i=1}^{N} \max(0, \lvert y_i - f(\vec{x}_i) \rvert - \varepsilon)$	$\lambda \lVert \vec{w} \rVert_2^2$	ε-SVR
Quadratic ε-insensitive loss: $\sum_{i=1}^{N} \max(0, (y_i - f(\vec{x}_i))^2 - \varepsilon)$	$\lambda \lVert \vec{w} \rVert_2^2$	Another variant of ε-SVR
Mean squared error: $\sum_{i=1}^{N} (y_i - f(\vec{x}_i))^2$	$\lambda \lVert \vec{w} \rVert_2^2$	Ridge regression
Mean linear error: $\sum_{i=1}^{N} \lvert y_i - f(\vec{x}_i) \rvert$	$\lambda \lVert \vec{w} \rVert_2^2$	Another variant of ridge regression

Comparing loss functions of regression methods

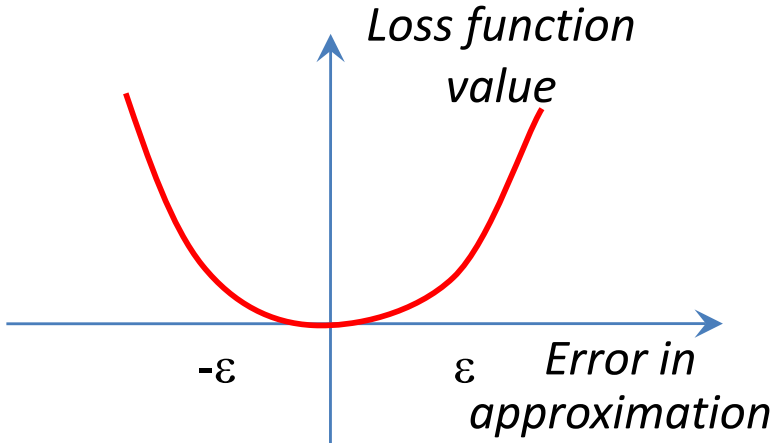
Linear ε-insensitive loss



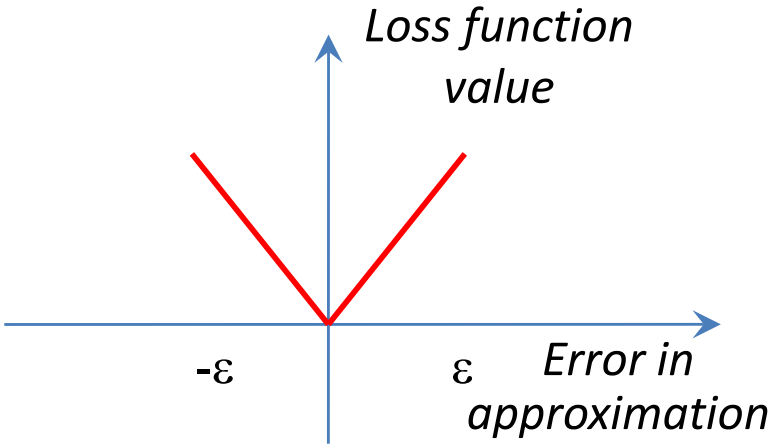
Quadratic ε-insensitive loss



Mean squared error



Mean linear error



Applying ε-SVR to real data

In the absence of domain knowledge about decision functions, it is recommended to optimize the following parameters (e.g., by cross-validation using grid-search):

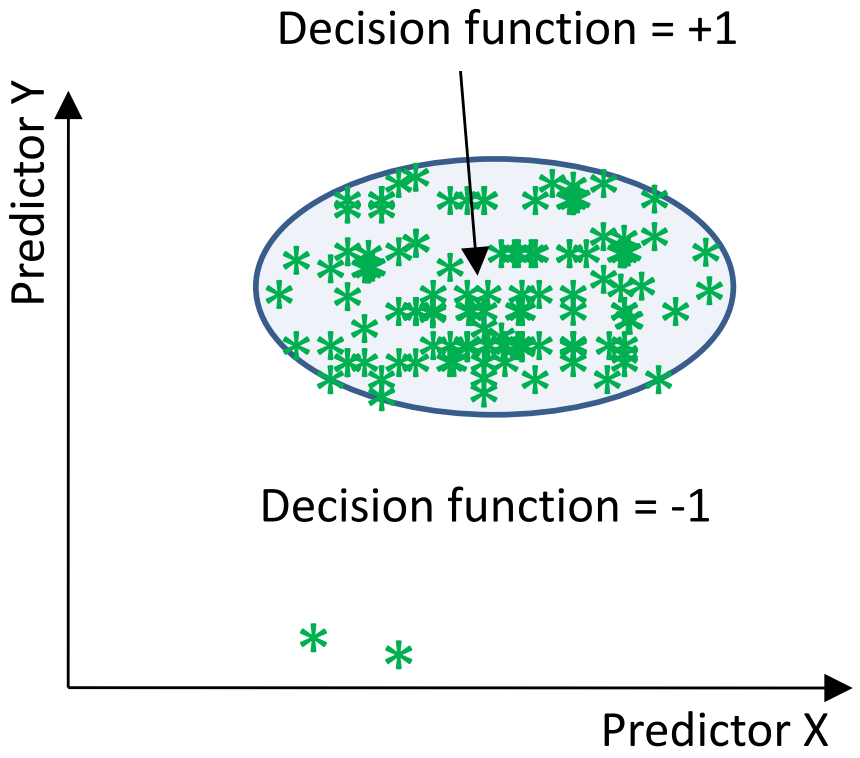
- parameter *C*
- parameter ε
- kernel parameters (e.g., degree of polynomial)

Notice that parameter ε depends on the ranges of variables in the dataset; therefore it is recommended to normalize/re-scale data prior to applying ε-SVR.

Novelty detection with SVM-based methods

What is it about?

- Find the simplest and most compact region in the space of predictors where the majority of data samples “live” (i.e., with the highest density of samples).
- Build a decision function that takes value +1 in this region and -1 elsewhere.
- Once we have such a decision function, we can identify novel or outlier samples/patients in the data.

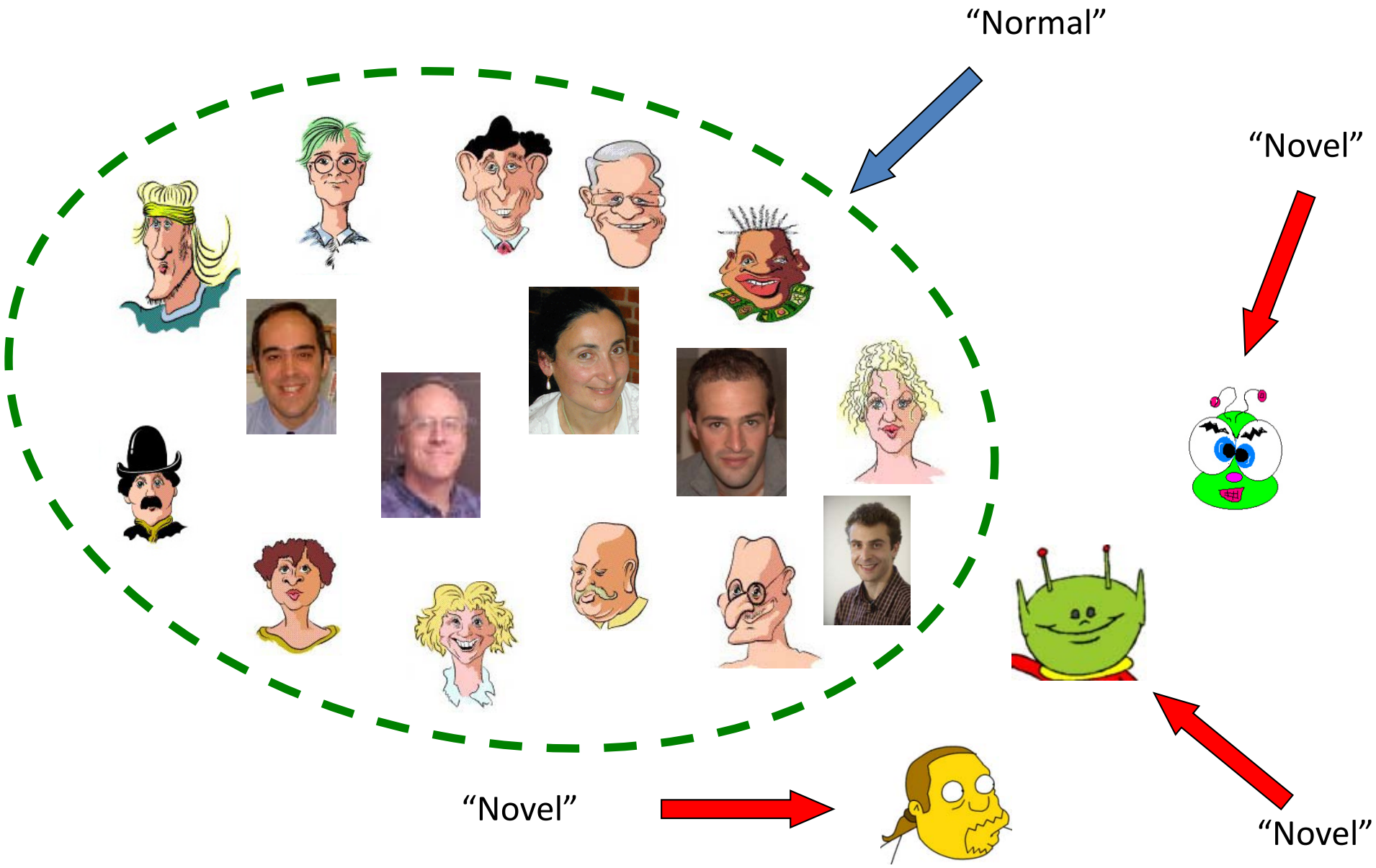


Key assumptions

- We do not know classes/labels of samples (positive or negative) in the data available for learning
  ➔ this is not a classification problem
- All positive samples are similar but each negative sample can be different in its own way

Thus, do not need to collect data for negative samples!

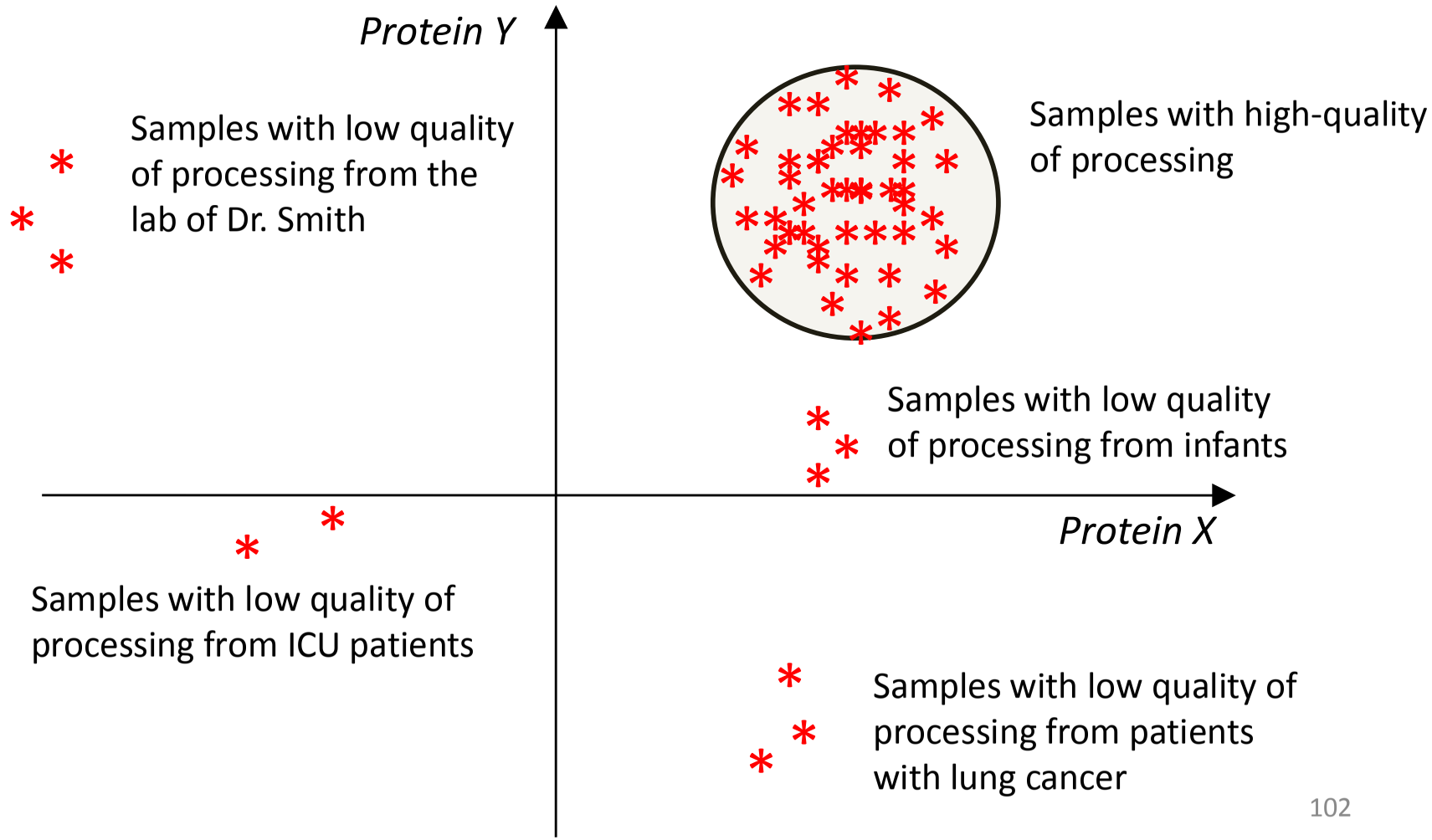
Sample applications



Modified from: www.cs.huji.ac.il/course/2004/learns/NoveltyDetection.ppt

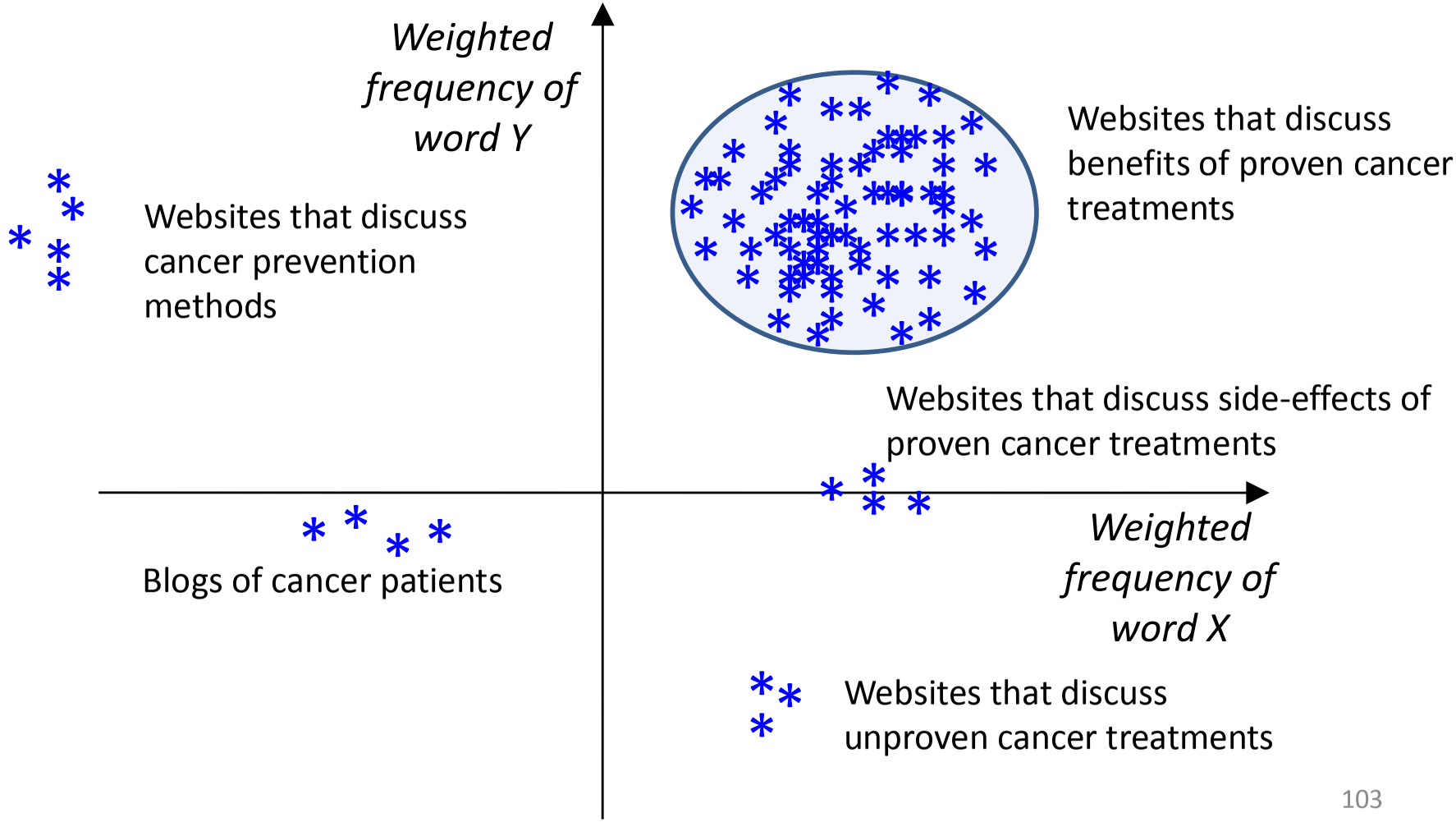
Sample applications

Discover deviations in sample handling protocol when doing quality control of assays.



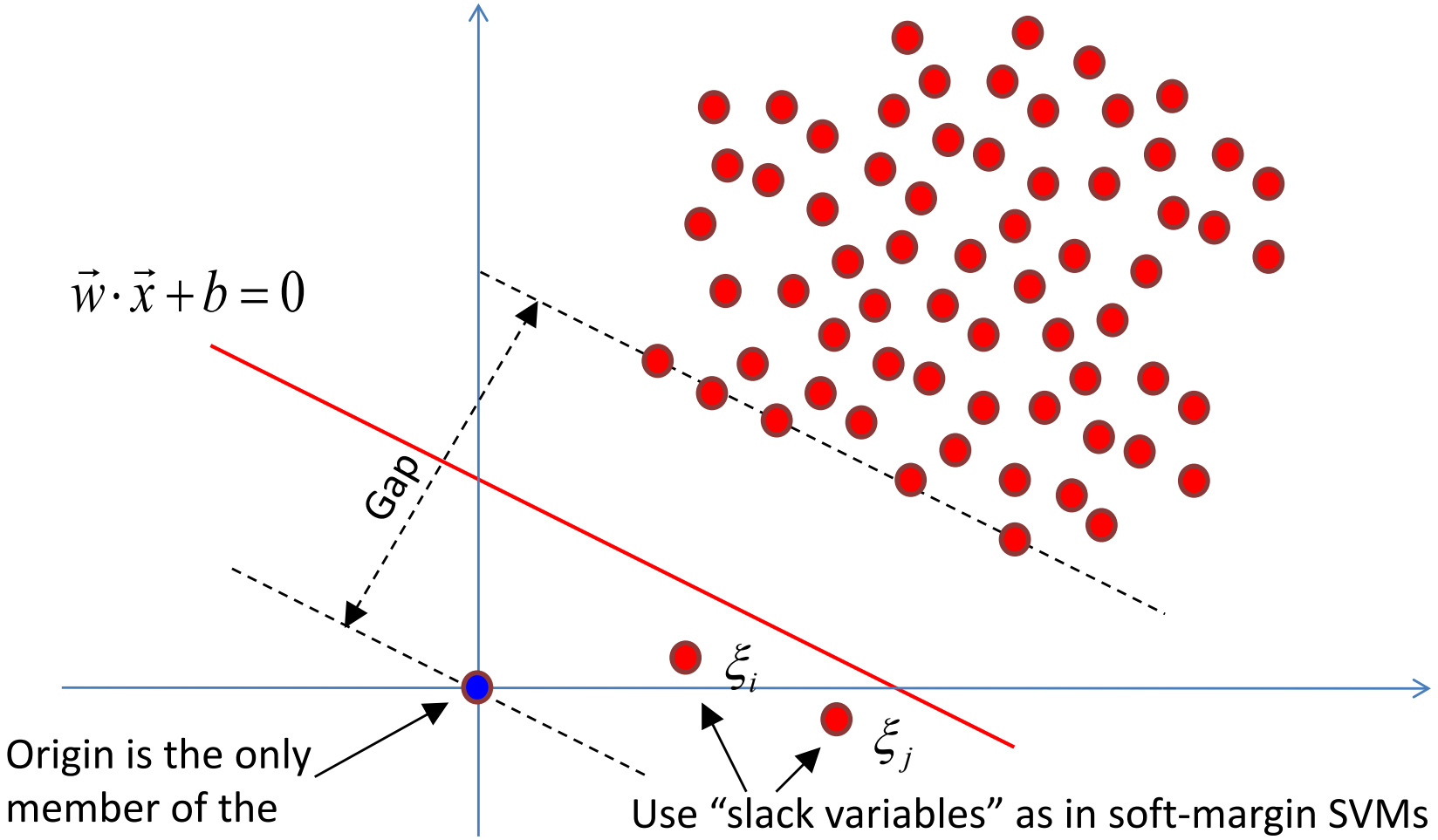
Sample applications

Identify websites that discuss benefits of proven cancer treatments.



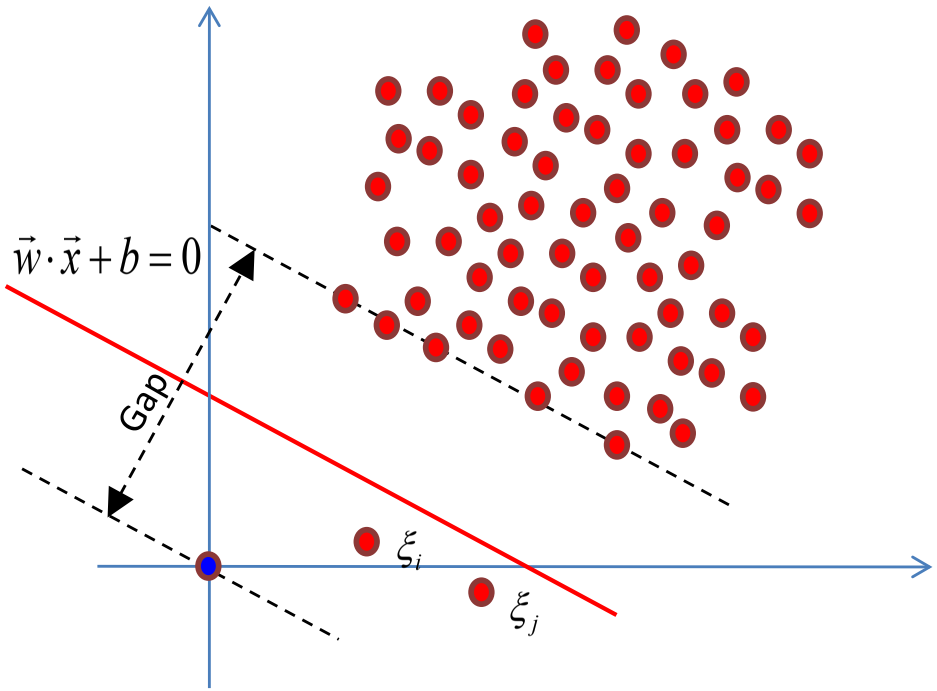
One-class SVM

Main idea: Find the maximal gap hyperplane that separates data from the origin (i.e., the only member of the second class is the origin).



Formulation of one-class SVM: linear case

Given training data: $\vec{x}_1, \vec{x}_2, ..., \vec{x}_N \in R^n$



Find $f(\vec{x}) = sign(\vec{w}\cdot\vec{x}+b)$

by minimizing $\frac{1}{2}\|\vec{w}\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i + b$

subject to constraints:

$$\vec{w}\cdot\vec{x}+b \geq -\xi_i$$

$$\xi_i \geq 0$$

for $i = 1,\ldots,N.$

upper bound on the fraction of outliers (i.e., points outside decision surface) allowed in the data

i.e., the decision function should be positive in all training samples except for small deviations

Formulation of one-class SVM: linear and non-linear cases

Linear case

Find $f(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b)$

by minimizing $\frac{1}{2}\|\vec{w}\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i + b$

subject to constraints:

$$\vec{w} \cdot \vec{x} + b \geq -\xi_i$$

$$\xi_i \geq 0$$

for $i = 1,\ldots,N.$

Non-linear case (use "kernel trick")

Find $f(\vec{x}) = sign(\vec{w} \cdot \Phi(\vec{x}) + b)$

by minimizing $\frac{1}{2}\|\vec{w}\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i + b$

subject to constraints:

$$\vec{w} \cdot \Phi(\vec{x}) + b \geq -\xi_i$$

$$\xi_i \geq 0$$

for $i = 1,\ldots,N.$

More about one-class SVM

- One-class SVMs inherit most of properties of SVMs for binary classification (e.g., “kernel trick”, sample efficiency, ease of finding of a solution by efficient optimization method, etc.);
- The choice of other parameter ν significantly affects the resulting decision surface.
- The choice of origin is arbitrary and also significantly affects the decision surface returned by the algorithm.

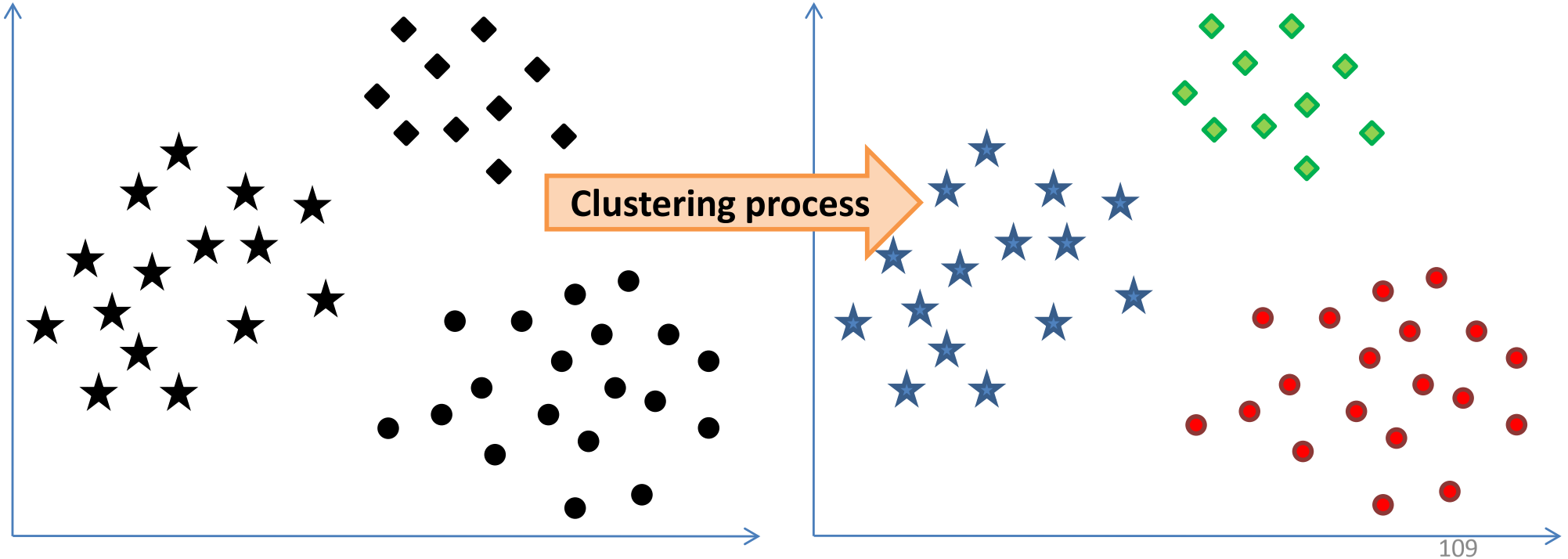
Support vector clustering

Contributed by **Nikita Lytkin**

Goal of clustering (aka class discovery)

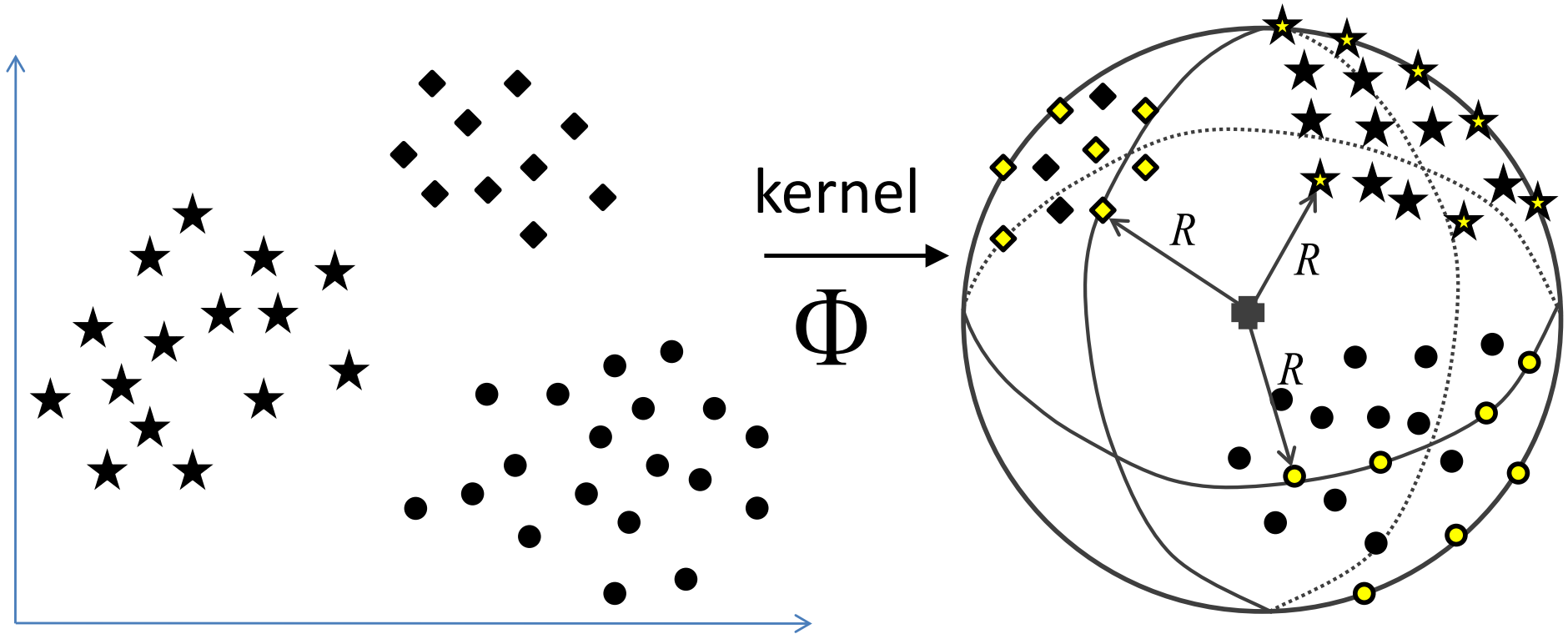
Given a heterogeneous set of data points $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \in R^n$

Assign labels $y_1, y_2, \ldots, y_N \in \{1,2,\ldots,K\}$ such that points with the same label are highly "similar" to each other and are distinctly different from the rest



Support vector domain description

- Support Vector Domain Description (SVDD) of the data is a set of vectors lying on the surface of the smallest hyper-sphere enclosing all data points *in a feature space*
  - These surface points are called *Support Vectors*



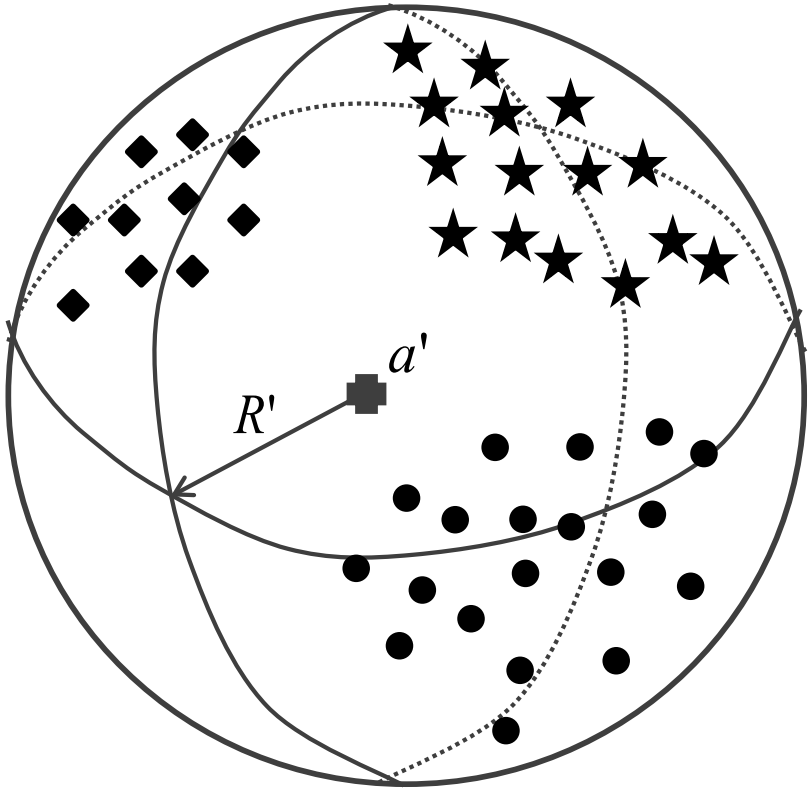
SVDD optimization criterion

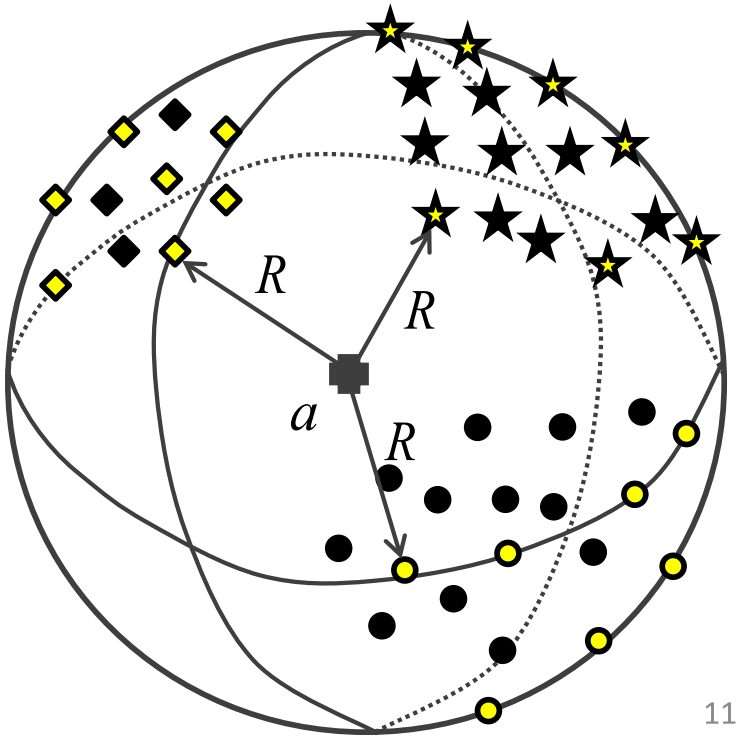
Formulation with hard constraints:

Minimize R^2 subject to $\| \Phi(x_i) - a \|^2 \leq R^2$ for $i = 1,\ldots,N$

Squared radius of the sphere

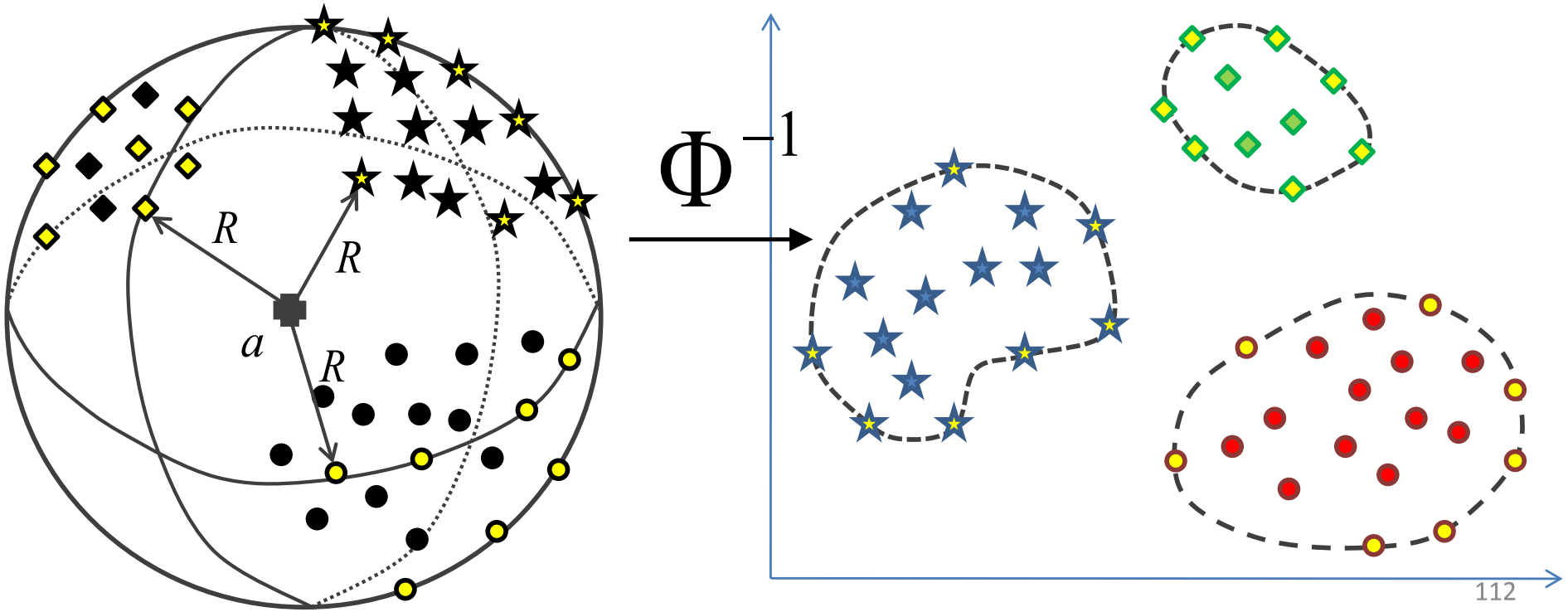
Constraints





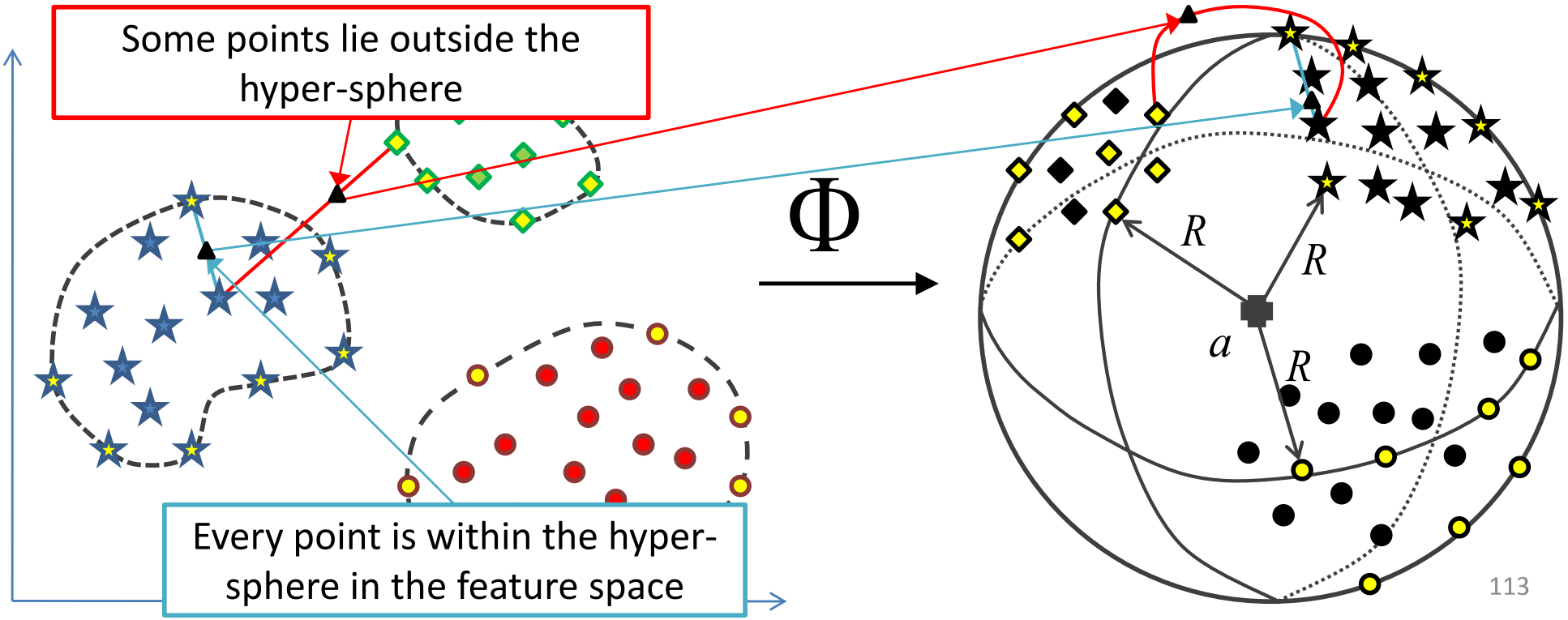
Main idea behind Support Vector Clustering

- Cluster boundaries in the input space are formed by the set of points that when mapped from the input space to the feature space fall exactly on the surface of the minimal enclosing hyper-sphere
  - SVs identified by SVDD are a subset of the cluster boundary points



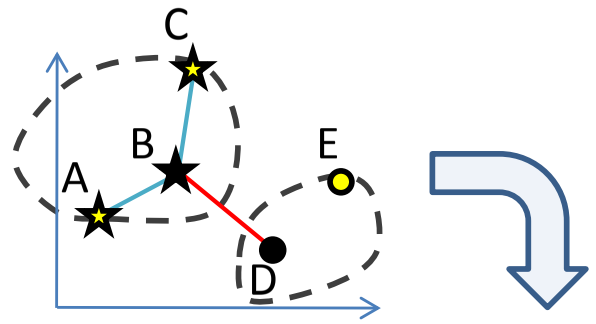
Cluster assignment in SVC

- Two points x_i and x_j belong to the same cluster (i.e., have the same label) if every point of the line segment (x_i, x_j) projected to the feature space lies within the hyper-sphere

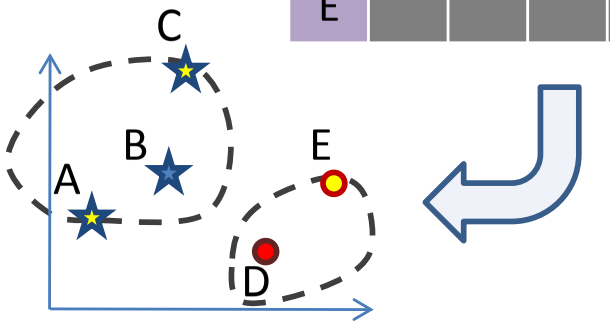


Cluster assignment in SVC (continued)

- Point-wise adjacency matrix is constructed by testing the line segments between every pair of points
- Connected components are extracted
- Points belonging to the same connected component are assigned the same label

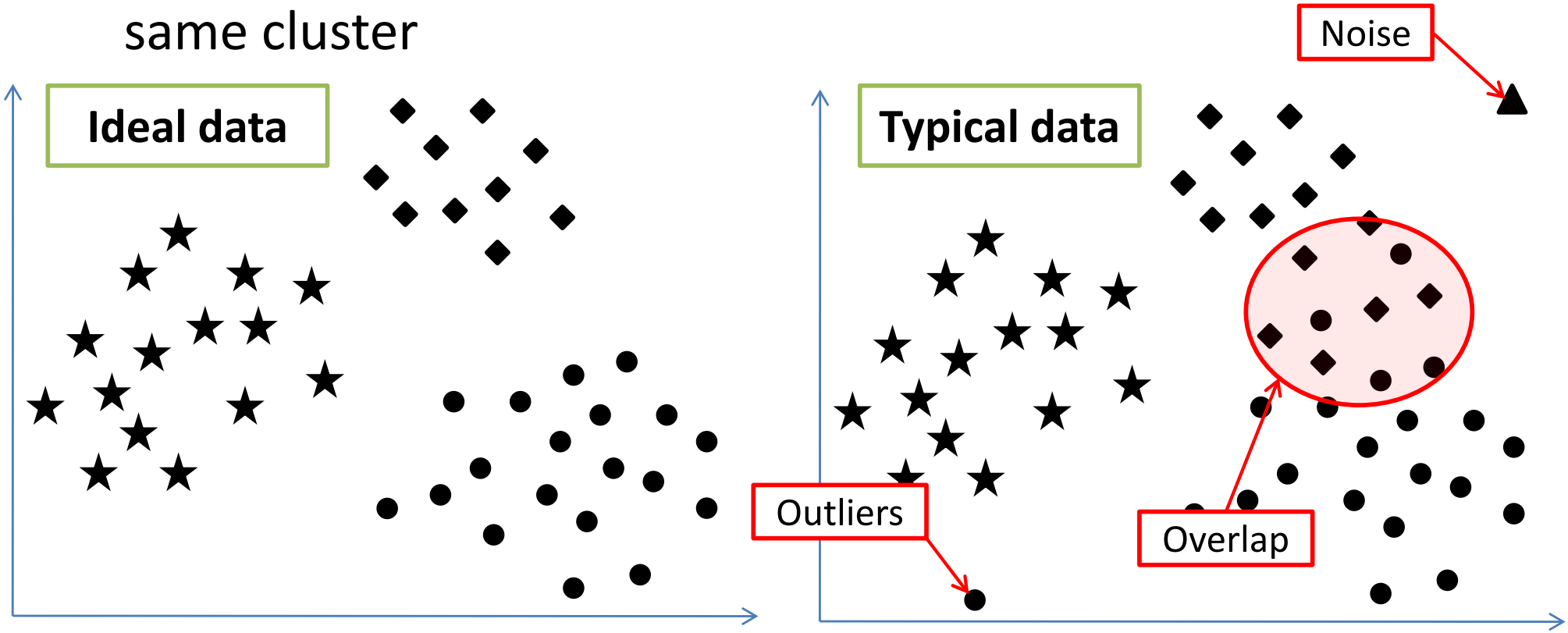


	A	B	C	D	E
A	1	1	1	0	0
B		1	1	0	0
C			1	0	0
D				1	1
E					1



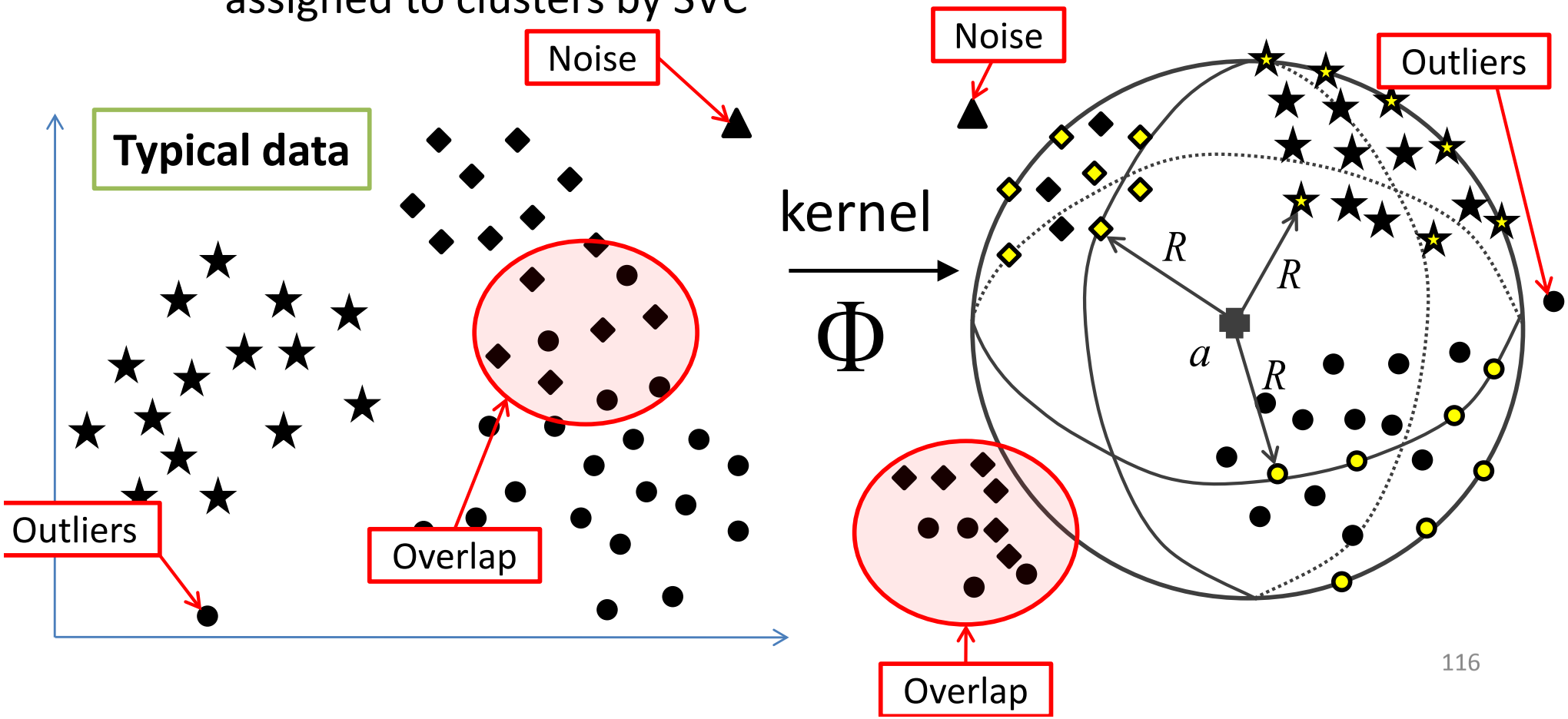
Effects of noise and cluster overlap

- In practice, data often contains noise, outlier points and overlapping clusters, which would prevent contour separation and result in all points being assigned to the same cluster



SVDD with soft constraints

- SVC can be used on noisy data by allowing a fraction of points, called Bounded SVs (BSV), to lie outside the hyper-sphere
  - BSVs are not considered as cluster boundary points and are not assigned to clusters by SVC



Soft SVDD optimization criterion

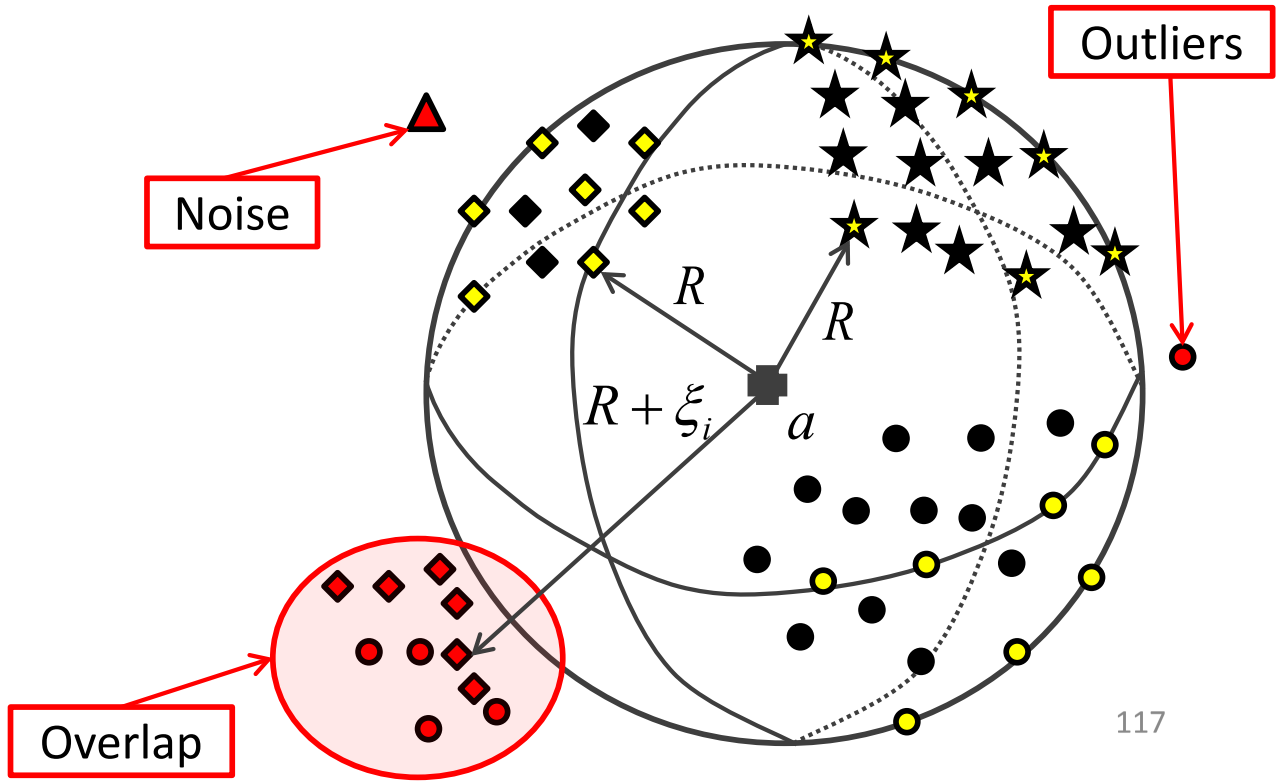
Primal formulation with soft constraints:

Minimize R^2 subject to $\| \Phi(x_i) - a \|^2 \leq R^2 + \xi_i$ $\xi_i \geq 0$ for $i = 1,\ldots,N$

Squared radius of the sphere

Soft constraints

Introduction of slack variables ξ_i mitigates the influence of noise and overlap on the clustering process



Dual formulation of soft SVDD

Minimize $W = \sum_i \beta_i K(x_i, x_i) - \sum_{i,j} \beta_i \beta_j K(x_i, x_j)$

subject to $0 \le \beta_i \le C$ for $i = 1, \ldots, N$

Constraints

- As before, $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ denotes a kernel function
- Parameter $0 < C \le 1$ gives a trade-off between volume of the sphere and the number of errors (*C*=1 corresponds to hard constraints)
- Gaussian kernel $K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \left\| \vec{x}_i - \vec{x}_j \right\|^2)$ tends to yield tighter contour representations of clusters than the polynomial kernel
- The Gaussian kernel width parameter $\gamma > 0$ influences tightness of cluster boundaries, number of SVs and the number of clusters
  - Increasing γ causes an increase in the number of clusters

SVM-based variable selection

Understanding the weight vector w

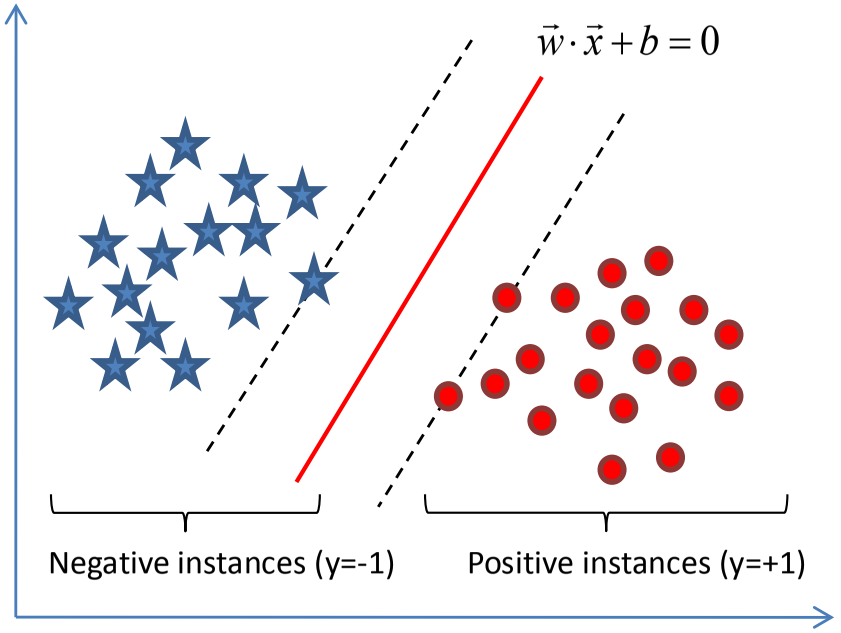
Recall standard SVM formulation:

Find $\vec{w}$ and b that minimize

$\frac{1}{2}\|\vec{w}\|^2$ subject to $y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1$

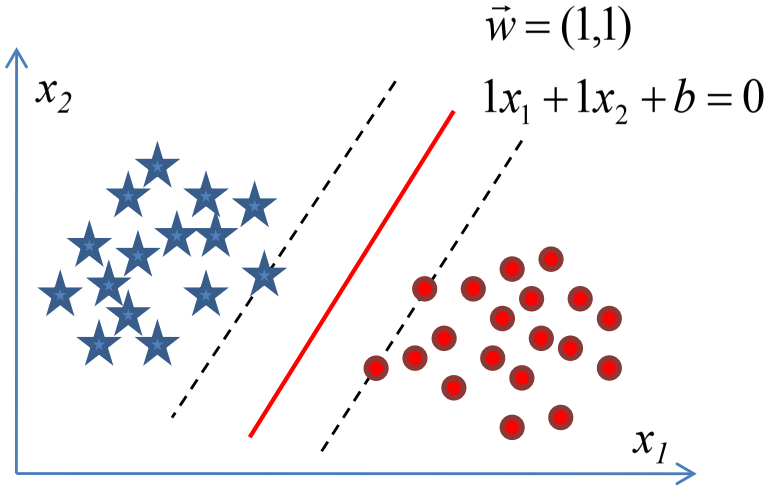
for $i = 1,\ldots,N.$

Use classifier: $f(\vec{x}) = sign(\vec{w}\cdot\vec{x}+b)$

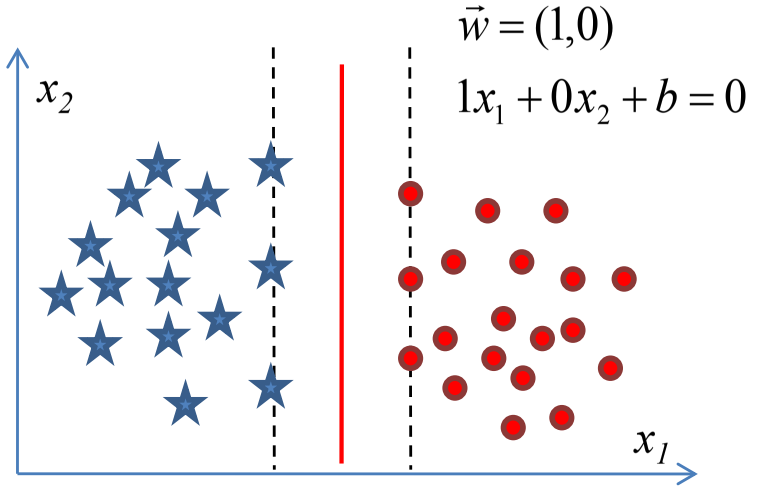


- The weight vector $\vec{w}$ contains as many elements as there are input variables in the dataset, i.e. $\vec{w}\in\mathbf{R}^n$.
- The magnitude of each element denotes importance of the corresponding variable for classification task.

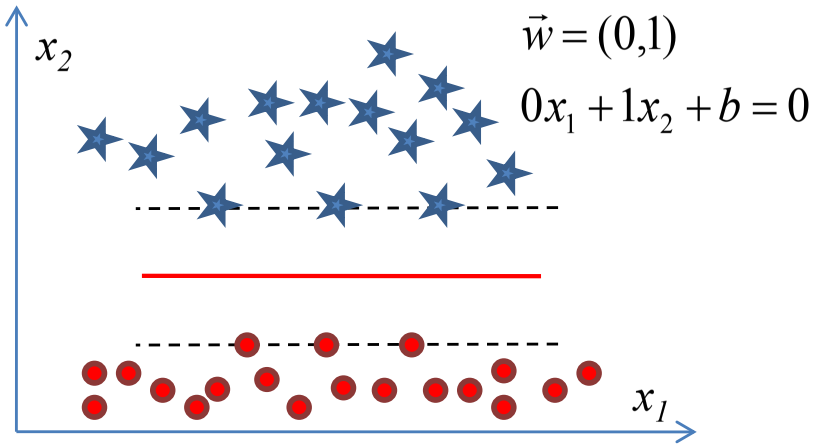
Understanding the weight vector w



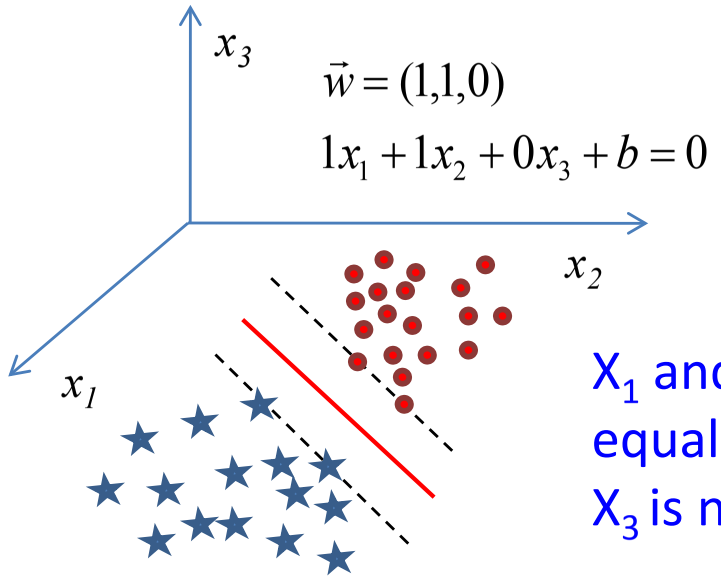
X_1 and X_2 are equally important



X_1 is important, X_2 is not

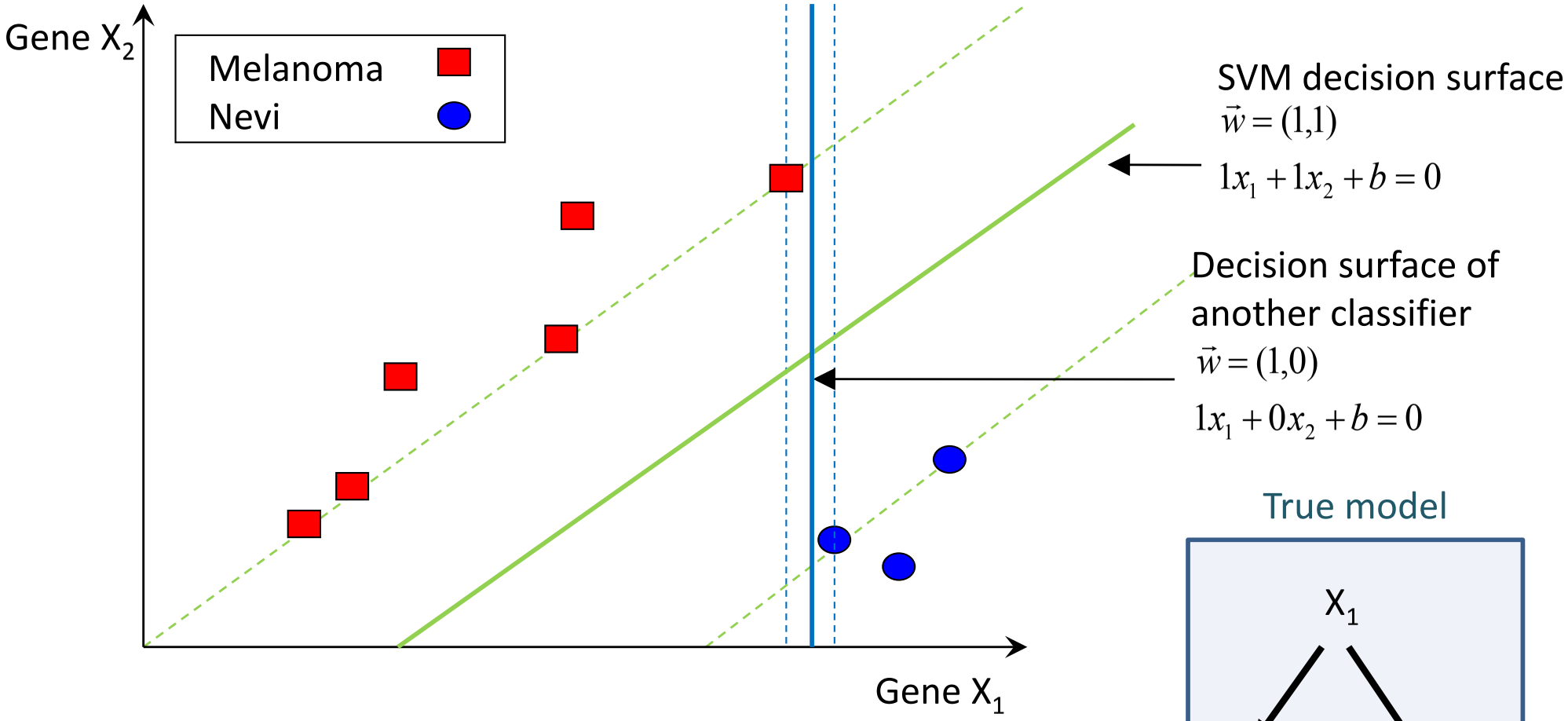


X_2 is important, X_1 is not



X_1 and X_2 are equally important, X_3 is not

Understanding the weight vector w



- In the true model, X_1 is causal and X_2 is redundant
- SVM decision surface implies that X_1 and X_2 are equally important; thus it is locally causally inconsistent
- There exists a causally consistent decision surface for this example
- Causal discovery algorithms can identify that X_1 is causal and X_2 is redundant

Simple SVM-based variable selection algorithm

Algorithm:

1. Train SVM classifier using data for all variables to estimate vector $\vec{w}$
2. Rank each variable based on the magnitude of the corresponding element in vector $\vec{w}$
3. Using the above ranking of variables, select the smallest nested subset of variables that achieves the best SVM prediction accuracy.

Simple SVM-based variable selection algorithm

Consider that we have 7 variables: $X_1, X_2, X_3, X_4, X_5, X_6, X_7$

The vector $\vec{w}$ is: (0.1, 0.3, 0.4, 0.01, 0.9, -0.99, 0.2)

The ranking of variables is: $X_6, X_5, X_3, X_2, X_7, X_1, X_4$

Subset of variables							Classification accuracy
X_6	X_5	X_3	X_2	X_7	X_1	X_4	0.920
X_6	X_5	X_3	X_2	X_7	X_1		0.920
X_6	X_5	X_3	X_2	X_7			0.919
X_6	X_5	X_3	X_2				0.852
X_6	X_5	X_3					0.843
X_6	X_5						0.832
X_6							0.821

Best classification accuracy

Classification accuracy that is statistically indistinguishable from the best one

➔ Select the following variable subset: X_6, X_5, X_3, X_2, X_7

Simple SVM-based variable selection algorithm

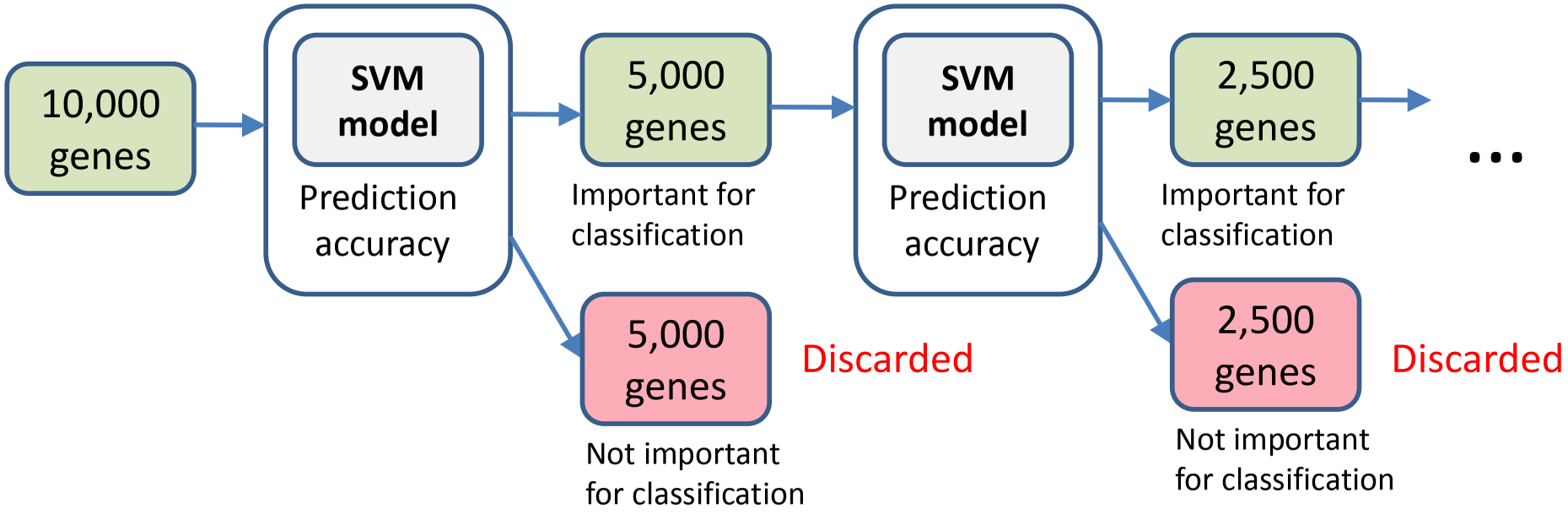
- SVM weights are not locally causally consistent → we may end up with a variable subset that is not causal and not necessarily the most compact one.
- The magnitude of a variable in vector $\vec{w}$ estimates the effect of removing that variable on the objective function of SVM (e.g., function that we want to minimize). However, this algorithm becomes sub-optimal when considering effect of removing several variables at a time… This pitfall is addressed in the SVM-RFE algorithm that is presented next.

SVM-RFE variable selection algorithm

Algorithm:

1. Initialize **V** to all variables in the data
2. Repeat
3. Train SVM classifier using data for variables in **V** to estimate vector $\vec{w}$
4. Estimate prediction accuracy of variables in **V** using the above SVM classifier (e.g., by cross-validation)
5. Remove from **V** a variable (or a subset of variables) with the smallest magnitude of the corresponding element in vector $\vec{w}$
6. Until there are no variables in **V**
7. Select the smallest subset of variables with the best prediction accuracy

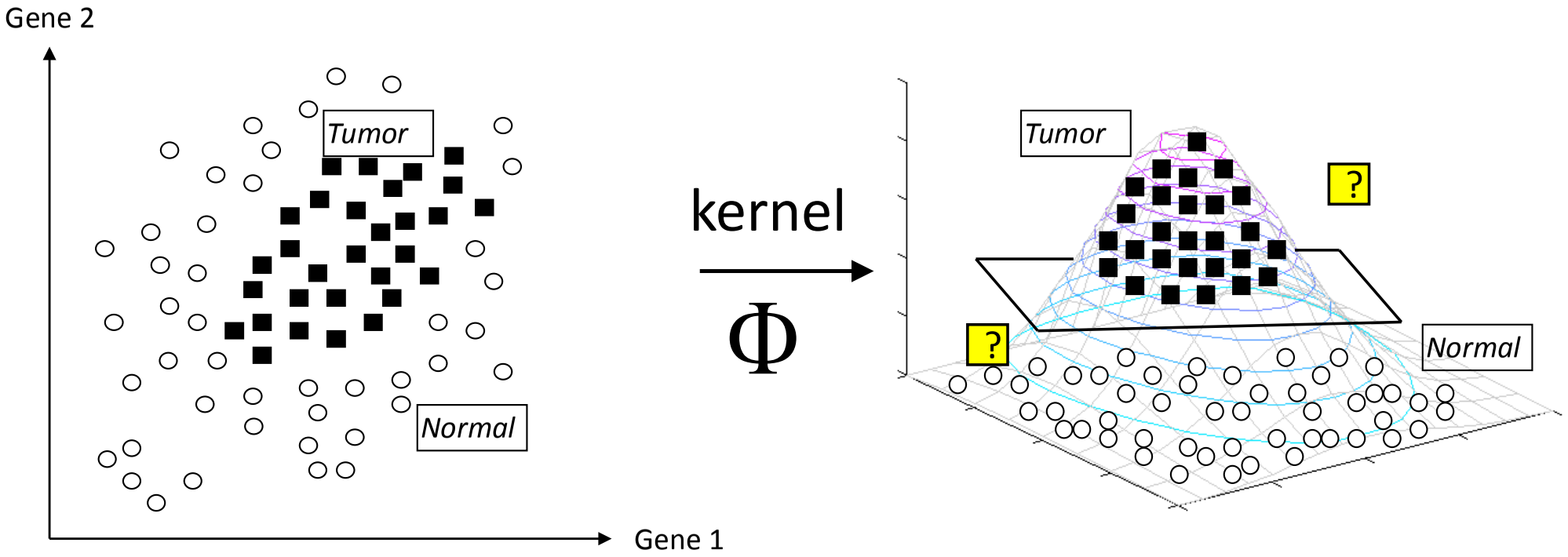
SVM-RFE variable selection algorithm



- Unlike simple SVM-based variable selection algorithm, SVM-RFE estimates vector $\vec{w}$ many times to establish ranking of the variables.
- Notice that the prediction accuracy should be estimated at each step in an unbiased fashion, e.g. by cross-validation.

SVM variable selection in feature space

The real power of SVMs comes with application of the kernel trick that maps data to a much higher dimensional space ("feature space") where the data is linearly separable.



input space

feature space

SVM variable selection in feature space

- We have data for 100 SNPs ($X_1,\ldots,X_{100}$) and some phenotype.
- We allow up to 3rd order interactions, e.g. we consider:
  - $X_1,\ldots,X_{100}$
  - $X_1^2, X_1X_2, X_1X_3,\ldots,X_1X_{100},\ldots,X_{100}^2$
  - $X_1^3, X_1X_2X_3, X_1X_2X_4,\ldots,X_1X_{99}X_{100},\ldots,X_{100}^3$
- ***Task***: find the smallest subset of features (either SNPs or their interactions) that achieves the best predictive accuracy of the phenotype.
- ***Challenge***: If we have limited sample, we cannot explicitly construct and evaluate all SNPs and their interactions (176,851 features in total) as it is done in classical statistics.

SVM variable selection in feature space

Heuristic solution: Apply algorithm SVM-FSMB that:

1. Uses SVMs with polynomial kernel of degree 3 and selects M features (not necessarily input variables!) that have largest weights in the feature space.

   E.g., the algorithm can select features like: X_{10}, (X_1X_2), $(X_9X_2X_{22})$, $(X_7^2X_{98})$, and so on.

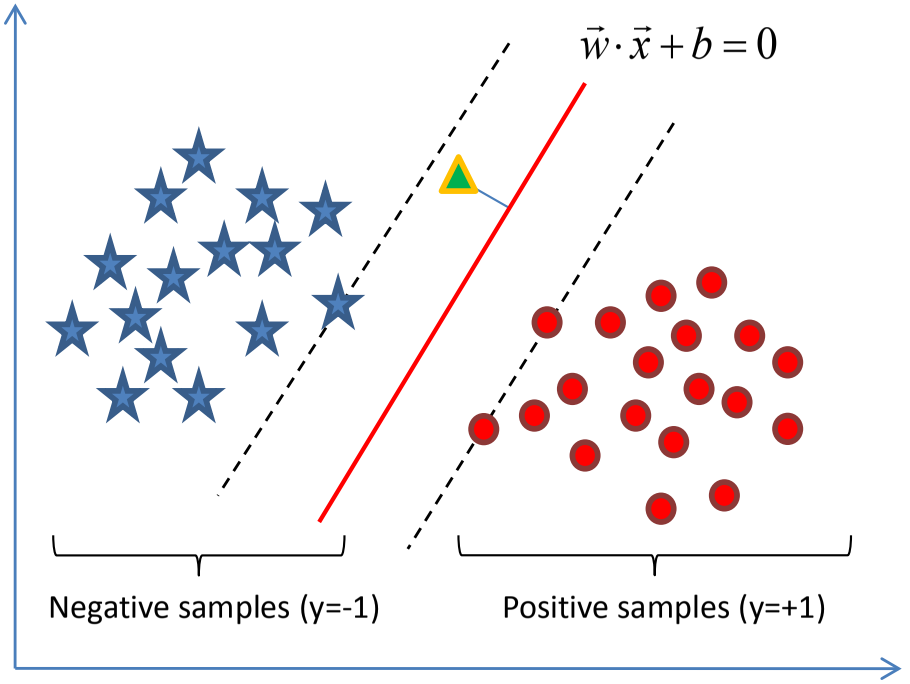
2. Apply HITON-MB Markov blanket algorithm to find the Markov blanket of the phenotype using M features from step 1.

Computing posterior class probabilities for SVM classifiers

Output of SVM classifier

1. SVMs output a class label (positive or negative) for each sample: $sign(\vec{w} \cdot \vec{x} + b)$

2. One can also compute distance from the hyperplane that separates classes, e.g. $\vec{w} \cdot \vec{x} + b$. These distances can be used to compute performance metrics like area under ROC curve.



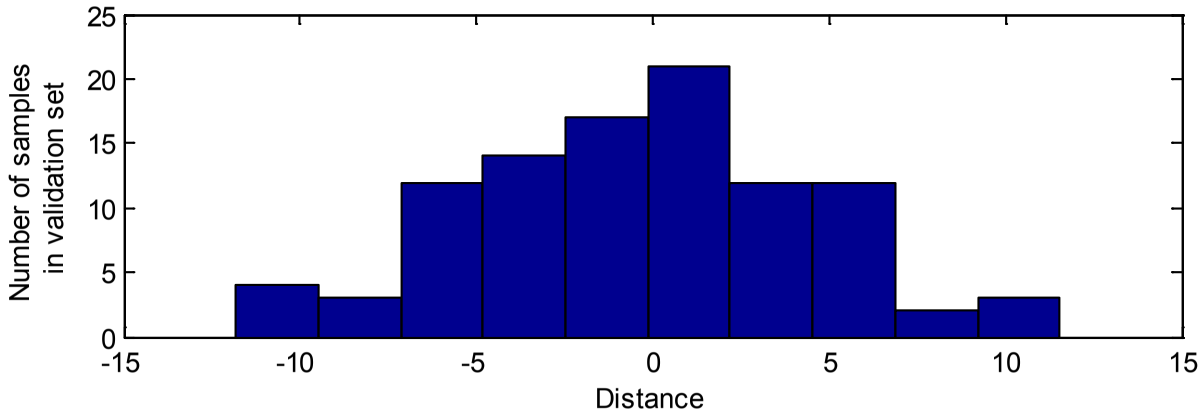
Question: How can one use SVMs to estimate posterior class probabilities, i.e., P(class positive | sample *x*)?

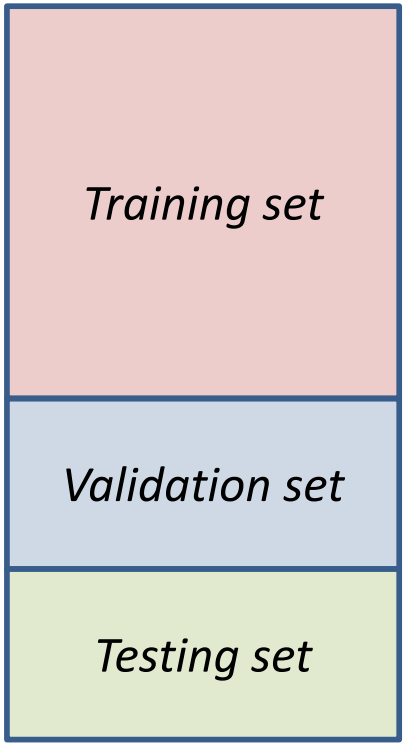
Simple binning method

1. Train SVM classifier in the *Training set*.
2. Apply it to the *Validation set* and compute distances from the hyperplane to each sample.

Sample #	1	2	3	4	5	…	98	99	100
Distance	2	-1	8	3	4		-2	0.3	0.8

3. Create a histogram with Q (e.g., say 10) bins using the above distances. Each bin has an upper and lower value in terms of distance.

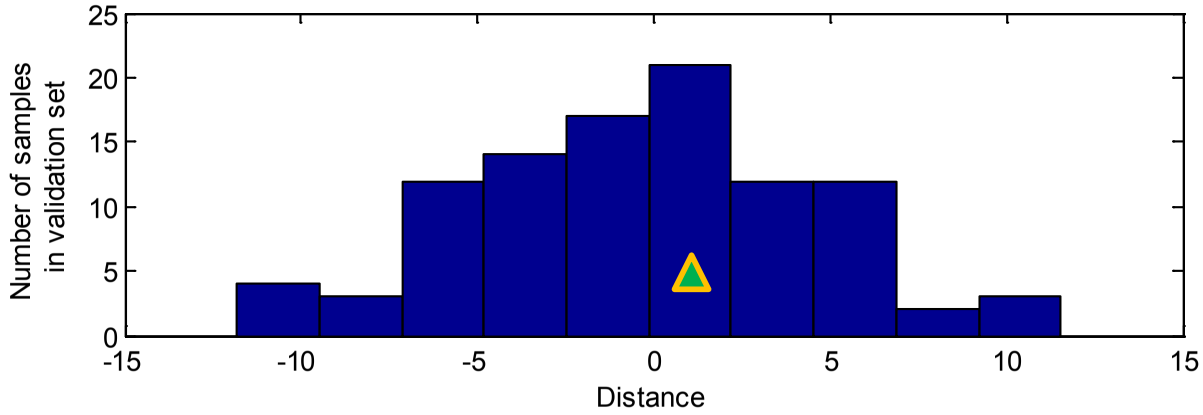


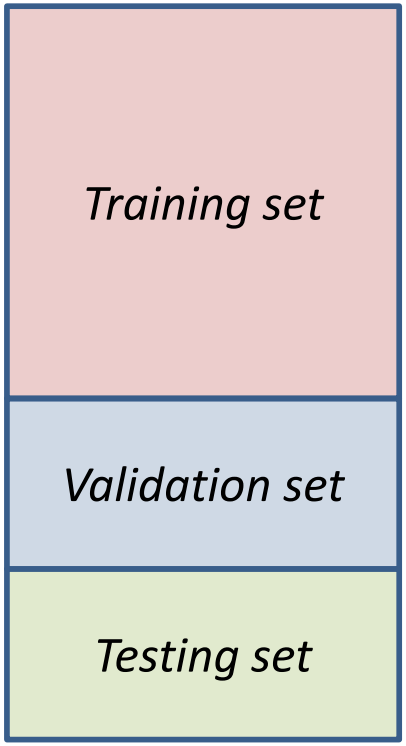


Simple binning method

4. Given a new sample from the *Testing set*, place it in the corresponding bin.

   E.g., sample #382 has distance to hyperplane = 1, so it is placed in the bin [0, 2.5]





5. Compute probability P(positive class | sample #382) as a fraction of true positives in this bin.

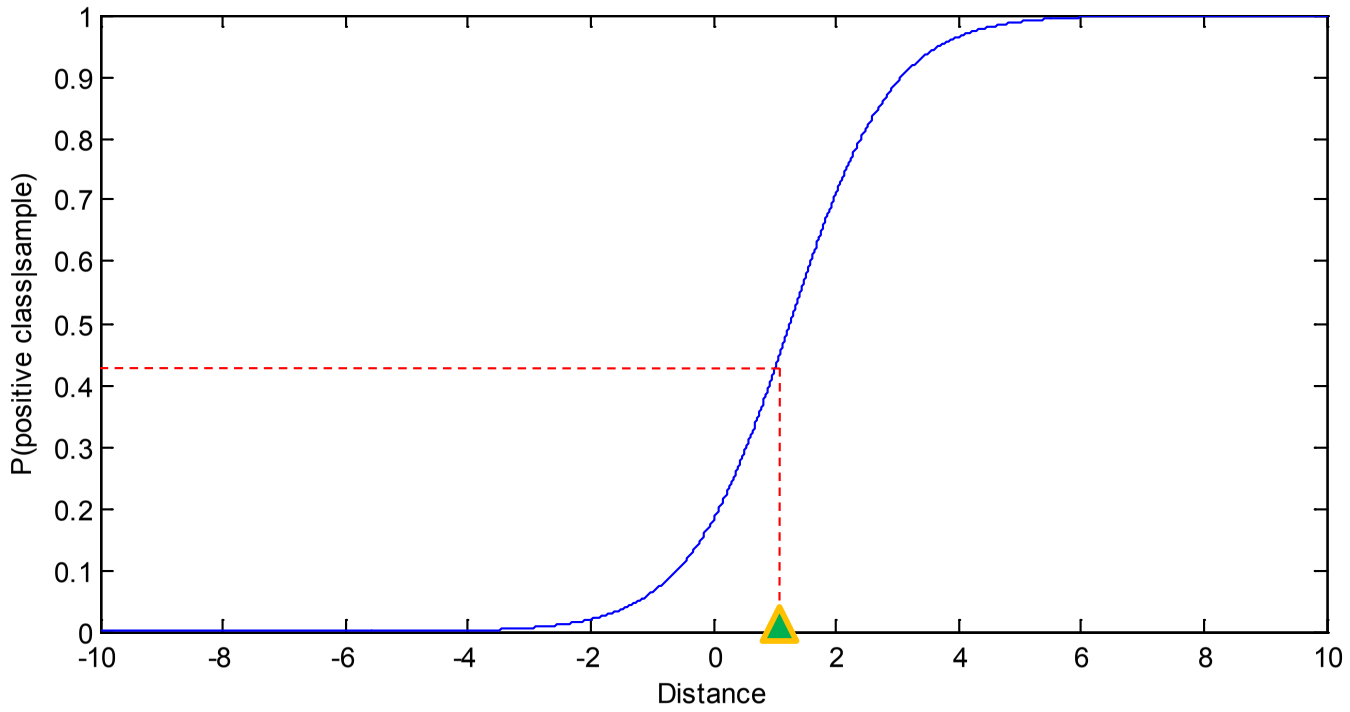
   E.g., this bin has 22 samples (from the *Validation set*), out of which 17 are true positive ones , so we compute P(positive class | sample #382) = 17/22 = 0.77

Platt's method

Convert distances output by SVM to probabilities by passing them through the sigmoid filter:

$$P(\mathit{positive\ class} \mid \mathit{sample}) = \frac{1}{1+\exp(Ad+B)}$$

where d is the distance from hyperplane and A and B are parameters.



Platt's method

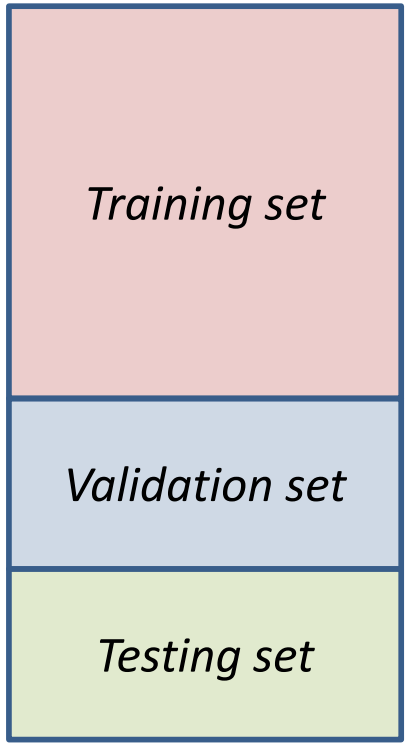
1. Train SVM classifier in the *Training set*.

2. Apply it to the *Validation set* and compute distances from the hyperplane to each sample.

Sample #	1	2	3	4	5	…	98	99	100
Distance	2	-1	8	3	4		-2	0.3	0.8

3. Determine parameters A and B of the sigmoid function by minimizing the negative log likelihood of the data from the *Validation set*.

4. Given a new sample from the *Testing set*, compute its posterior probability using sigmoid function.



Part 3

- Case studies (taken from our research)
  1. Classification of cancer gene expression microarray data
  2. Text categorization in biomedicine
  3. Prediction of clinical laboratory values
  4. Modeling clinical judgment
  5. Using SVMs for feature selection
  6. Outlier detection in ovarian cancer proteomics data
- Software
- Conclusions
- Bibliography

1. Classification of cancer gene expression microarray data

Comprehensive evaluation of algorithms for classification of cancer microarray data

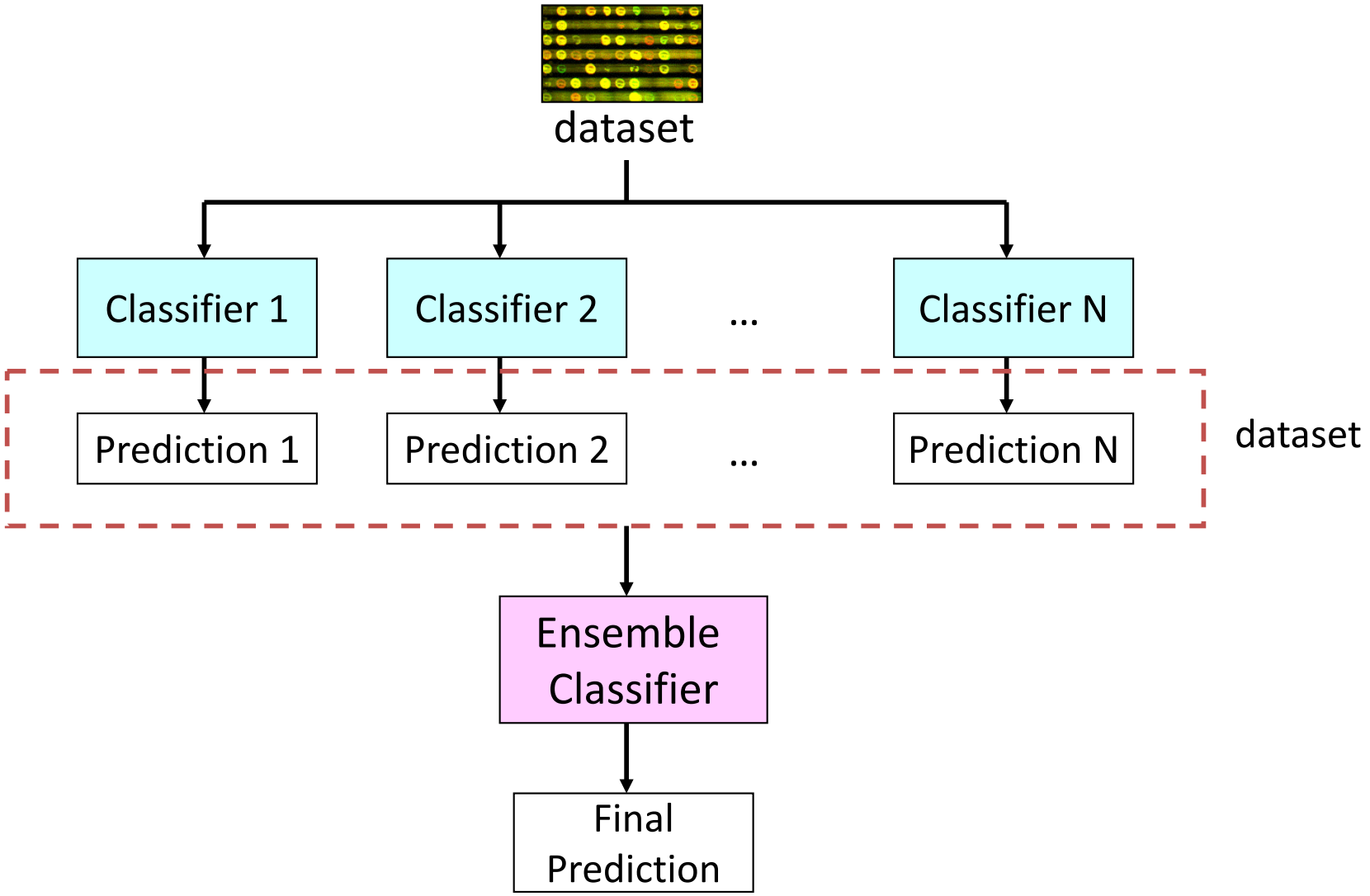
Main goals:

- Find the best performing decision support algorithms for cancer diagnosis from microarray gene expression data;
- Investigate benefits of using gene selection and ensemble classification methods.

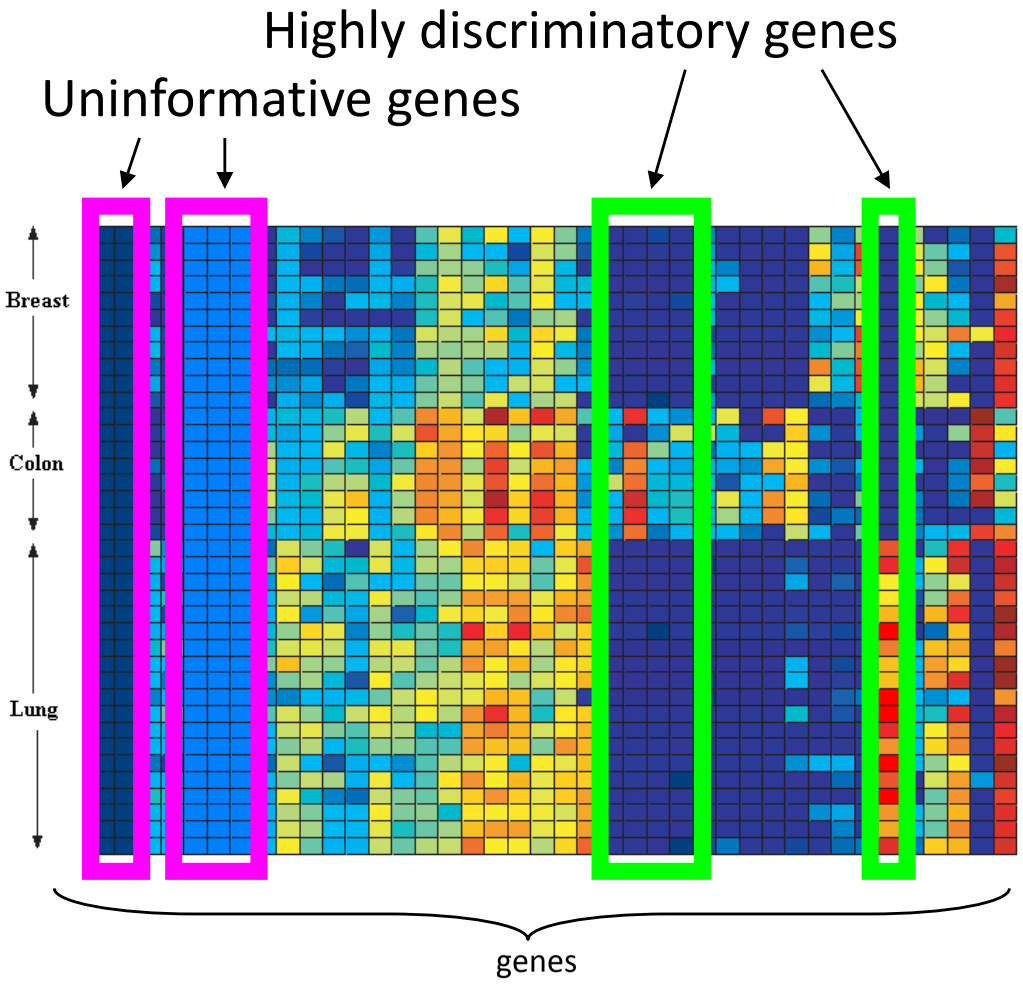
Classifiers

- K-Nearest Neighbors (KNN) — instance-based
- Backpropagation Neural Networks (NN)
- Probabilistic Neural Networks (PNN) — neural networks
- Multi-Class SVM: One-Versus-Rest (OVR)
- Multi-Class SVM: One-Versus-One (OVO)
- Multi-Class SVM: DAGSVM
- Multi-Class SVM by Weston & Watkins (WW)
- Multi-Class SVM by Crammer & Singer (CS) — kernel-based
- Weighted Voting: One-Versus-Rest
- Weighted Voting: One-Versus-One — voting
- Decision Trees: CART — decision trees

Ensemble classifiers



Gene selection methods



1. Signal-to-noise (S2N) ratio in one-versus-rest (OVR) fashion;
2. Signal-to-noise (S2N) ratio in one-versus-one (OVO) fashion;
3. Kruskal-Wallis nonparametric one-way ANOVA (KW);
4. Ratio of genes between-categories to within-category sum of squares (BW).

Performance metrics and statistical comparison

1. Accuracy
   + can compare to previous studies
   + easy to interpret & simplifies statistical comparison

2. Relative classifier information (RCI)
   + easy to interpret & simplifies statistical comparison
   + not sensitive to distribution of classes
   + accounts for difficulty of a decision problem

- Randomized permutation testing to compare accuracies of the classifiers (α=0.05)

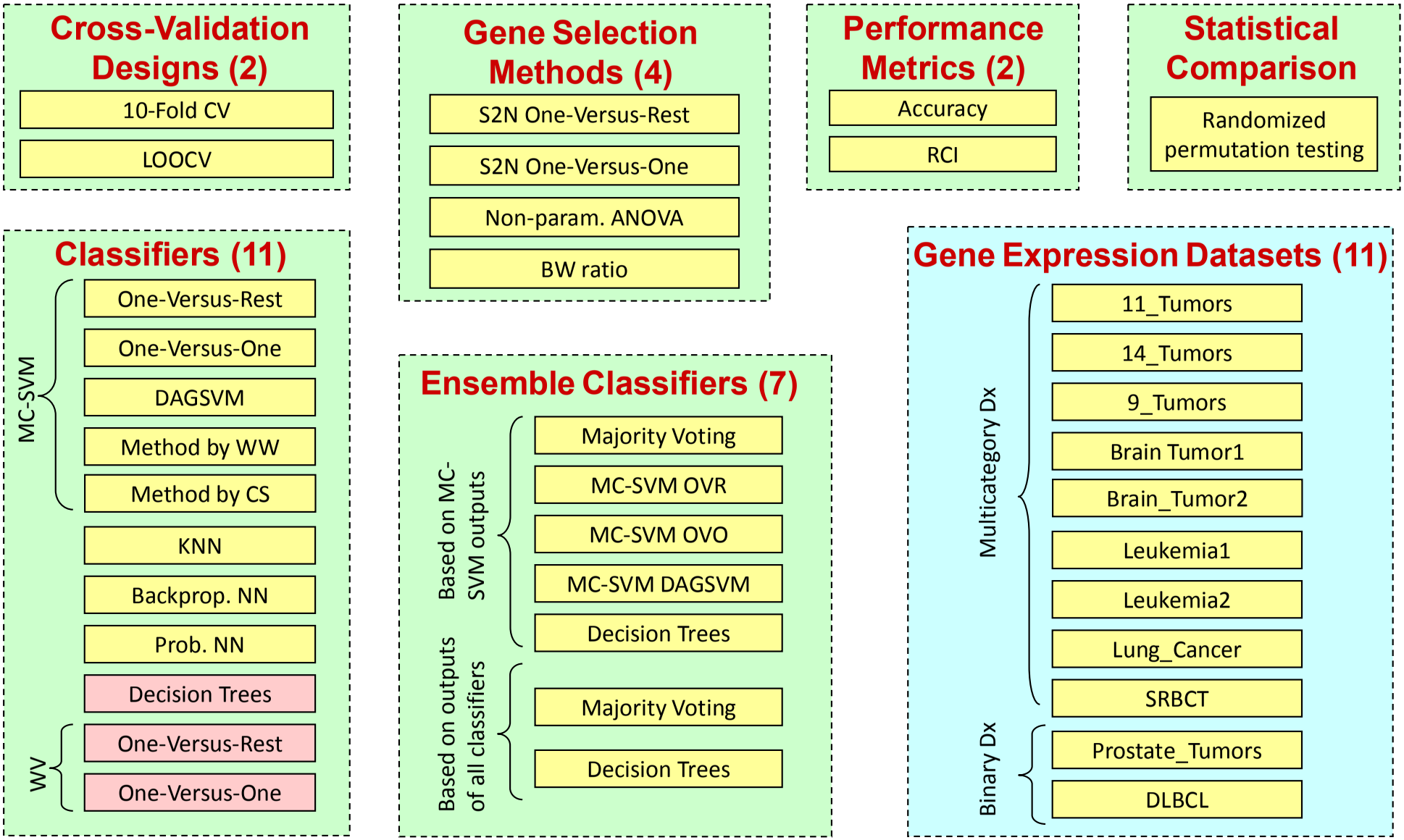
Microarray datasets

Dataset name	Number of			Reference
	Sam-ples	Variables (genes)	Cate-gories	
11_Tumors	174	12533	11	Su, 2001
14_Tumors	308	15009	26	Ramaswamy, 2001
9_Tumors	60	5726	9	Staunton, 2001
Brain_Tumor1	90	5920	5	Pomeroy, 2002
Brain_Tumor2	50	10367	4	Nutt, 2003
Leukemia1	72	5327	3	Golub, 1999
Leukemia2	72	11225	3	Armstrong, 2002
Lung_Cancer	203	12600	5	Bhattacherjee, 2001
SRBCT	83	2308	4	Khan, 2001
Prostate_Tumor	102	10509	2	Singh, 2002
DLBCL	77	5469	2	Shipp, 2002

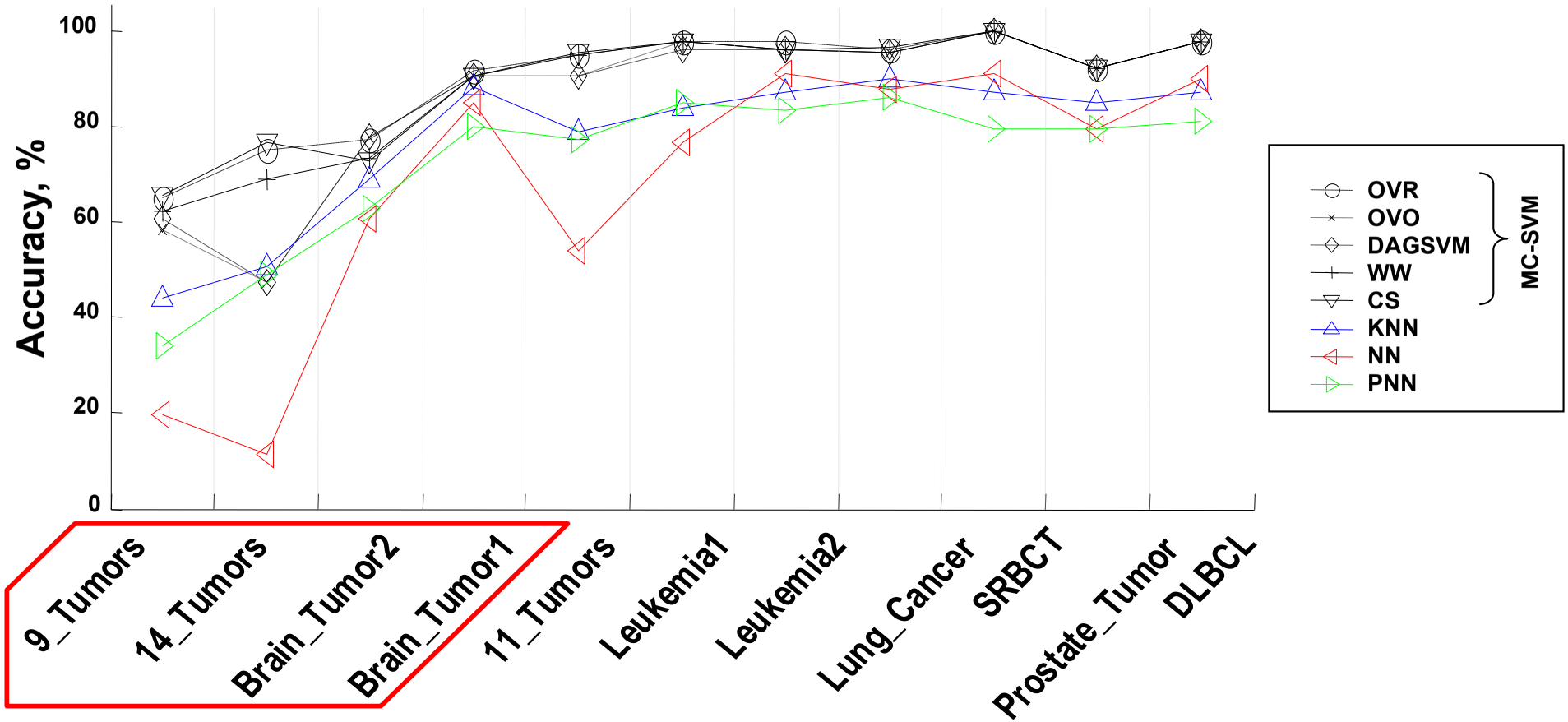
Total:

- ~1300 samples
- 74 diagnostic categories
- 41 cancer types and 12 normal tissue types

Summary of methods and datasets

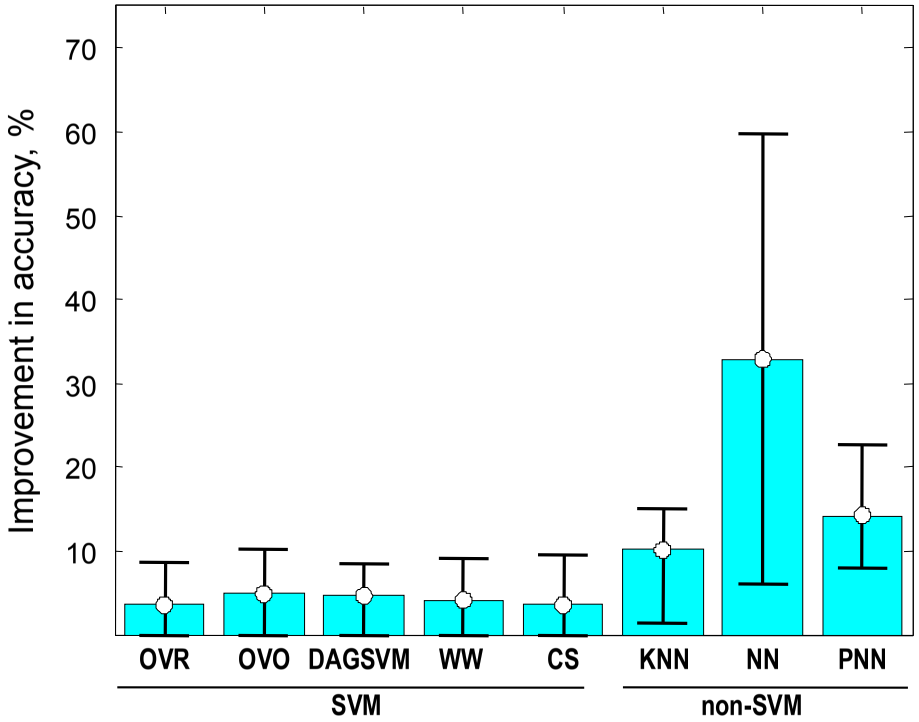


Results without gene selection

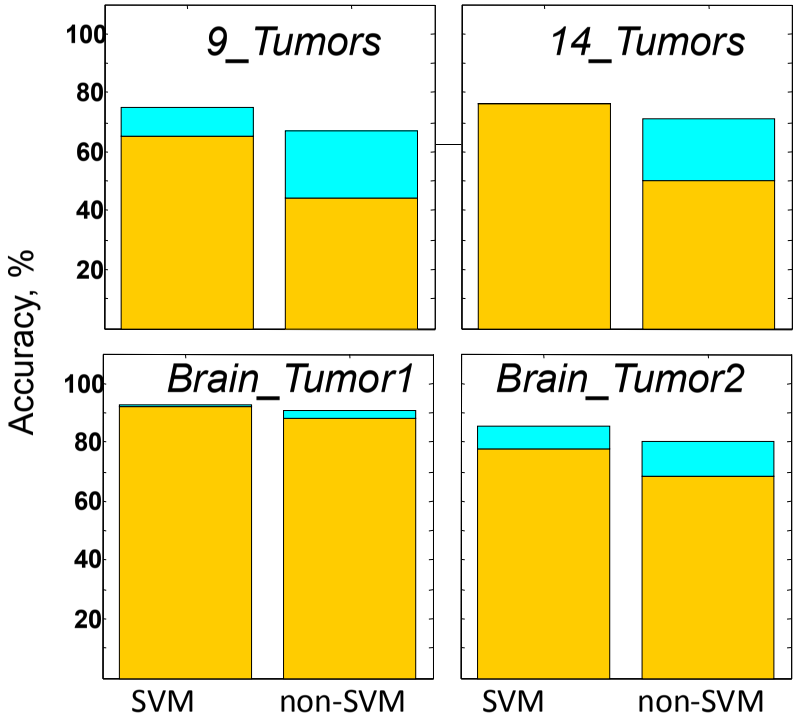


Results with gene selection

Improvement of diagnostic performance by gene selection (*averages for the four datasets*)

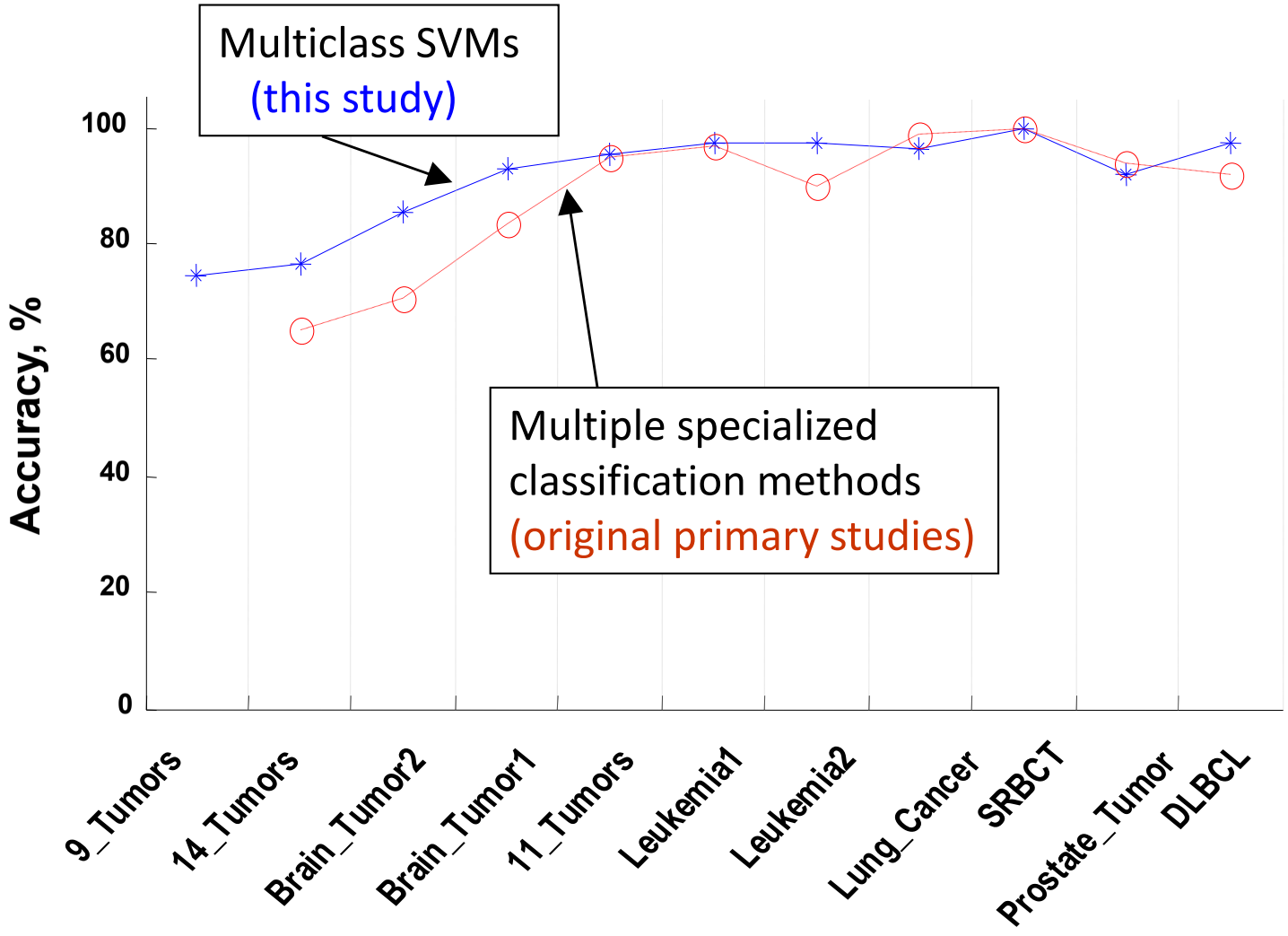


Diagnostic performance *before* and *after* gene selection



Average reduction of genes is 10-30 times

Comparison with previously published results



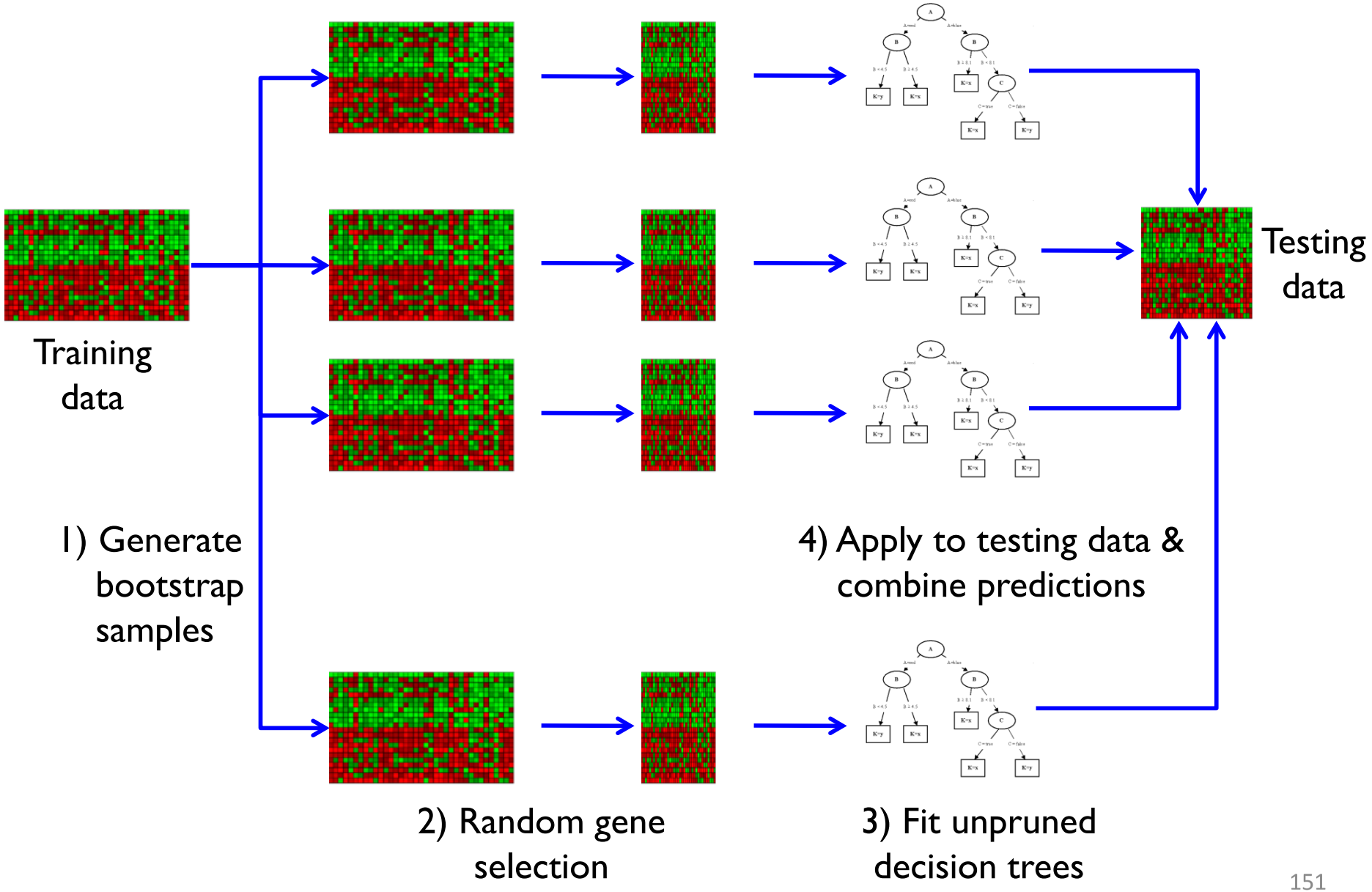
Summary of results

- Multi-class SVMs are the best family among the tested algorithms outperforming KNN, NN, PNN, DT, and WV.
- Gene selection in some cases improves classification performance of all classifiers, especially of non-SVM algorithms;
- Ensemble classification does not improve performance of SVM and other classifiers;
- Results obtained by SVMs favorably compare with the literature.

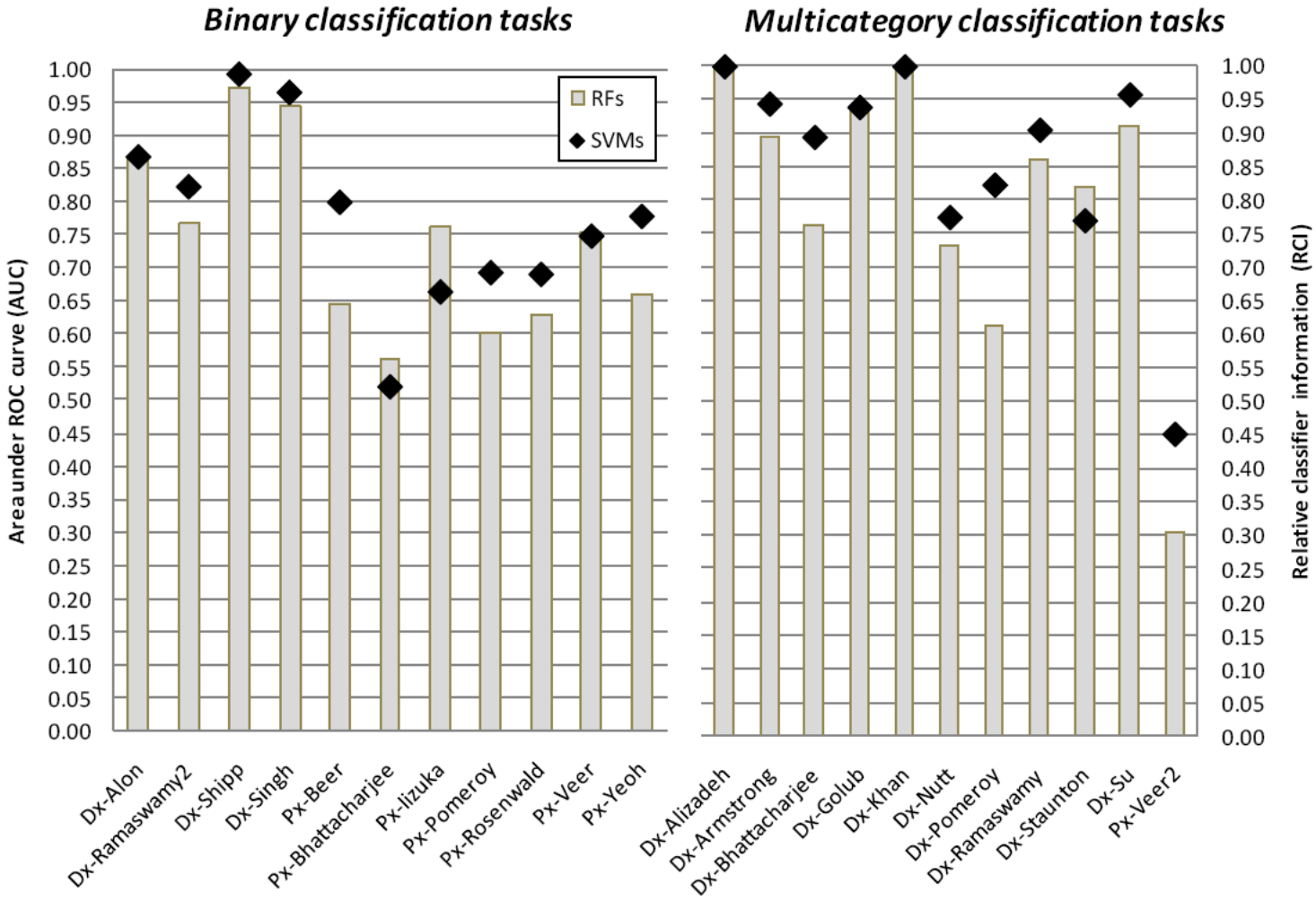
Random Forest (RF) classifiers

- Appealing properties
  - Work when # of predictors > # of samples
  - Embedded gene selection
  - Incorporate interactions
  - Based on theory of ensemble learning
  - Can work with binary & multiclass tasks
  - Does not require much fine-tuning of parameters
- Strong theoretical claims
- Empirical evidence: (Diaz-Uriarte and Alvarez de Andres, *BMC Bioinformatics*, 2006) reported superior classification performance of RFs compared to SVMs and other methods

Key principles of RF classifiers

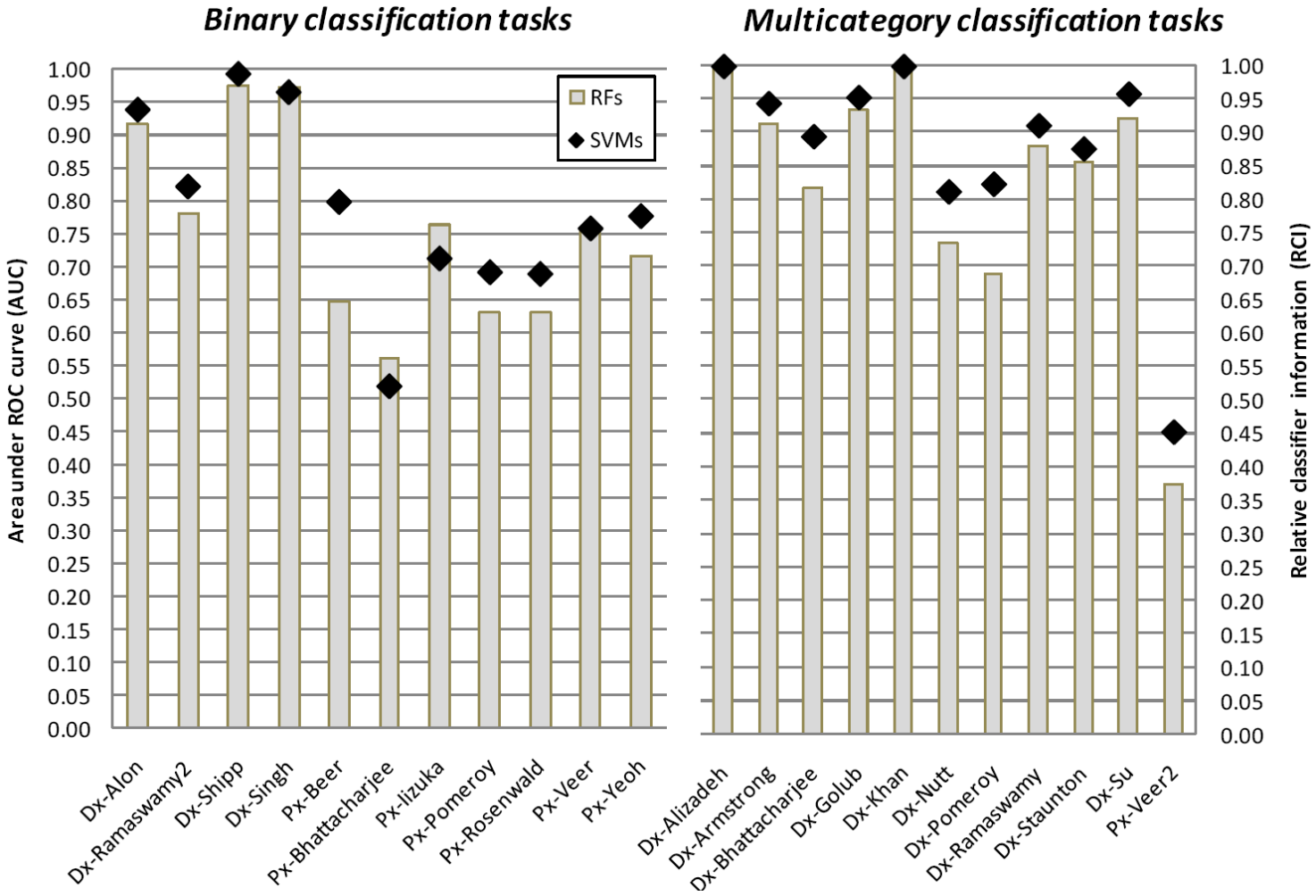


Results without gene selection



- SVMs *nominally* outperform RFs is 15 datasets, RFs outperform SVMs in 4 datasets, algorithms are exactly the same in 3 datasets.
- In 7 datasets SVMs outperform RFs *statistically significantly*.
- On average, the performance advantage of SVMs is 0.033 AUC and 0.057 RCI.

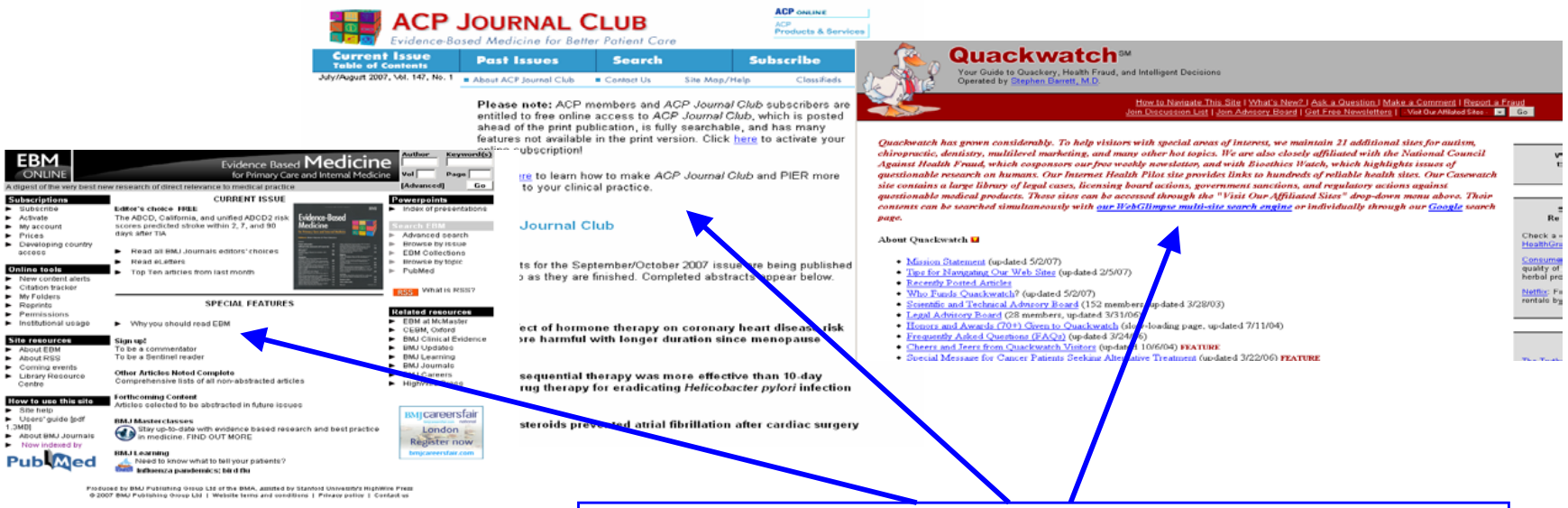
Results with gene selection



- SVMs *nominally* outperform RFs is 17 datasets, RFs outperform SVMs in 3 datasets, algorithms are exactly the same in 2 datasets.
- In 1 dataset SVMs outperform RFs *statistically significantly*.
- On average, the performance advantage of SVMs is 0.028 AUC and 0.047 RCI.

2. Text categorization in biomedicine

Models to categorize content and quality: Main idea



1. **Utilize existing (or easy to build)** training corpora

1: J Infect Dis. 2002 Mar 1;185(5):650-6. Epub 2002 Feb 14. Related Articles, Links

The University of Chicago Press

1 **The clinical significance of cerebrospinal fluid levels of kynurenine pathway metabolites and lactate in severe malaria.**

Medana IM, Hien TT, Day NP, Phu NH, Mai NT, Chu'ong LV, Chau TT, Taylor A, Salahifar H, Stocker R, Smythe G, Turner GD, Farrar J, White NJ, Hunt NH.

Nuffield Department of Clinical Laboratory Sciences, Oxford-Wellcome Centre for Tropical and Infectious Diseases

2 A retrospective study of 261 Vietnamese adults with severe malaria was conducted to determine the relationship between cerebrospinal fluid (CSF) levels of metabolites of the kynurenine pathway, the incidence of neurologic complications, and the disease outcome. Three metabolites were measured: the excitotoxin quinolinic acid (QA); the protective receptor antagonist kynurenic acid (KA); and the proinflammatory mediator picolinic acid (PA). These measurements were related prospectively to CSF lactate levels. QA and PA levels were elevated, compared with those of controls. There was no difference in the levels of KA between these groups. Although >40% of malaria patients had QA CSF concentrations in the micromolar range, there was no association with convulsions or depth of coma. Levels of QA and PA were associated significantly with death, but a multivariate analysis suggested that these elevations were a consequence of impaired renal function. CSF lactate remained an independent and significant predictor of poor outcome.

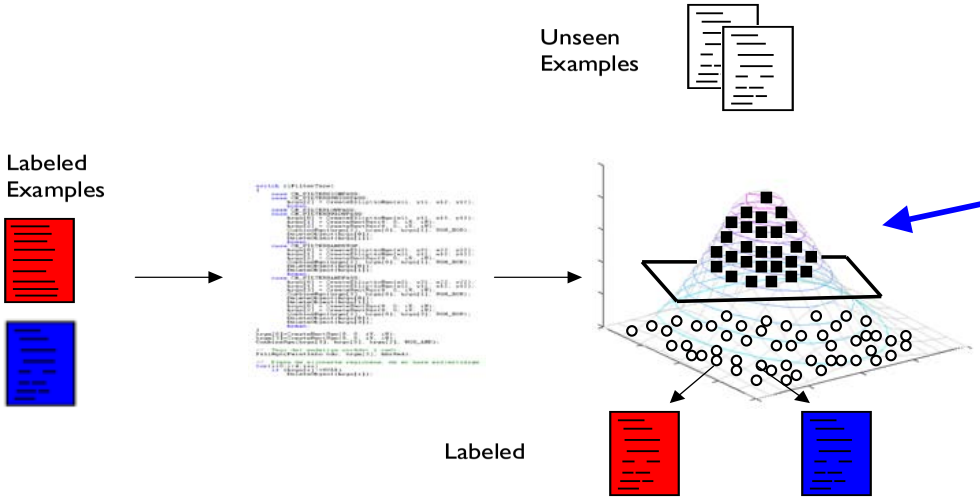
3 Publication Types:
- Clinical Trial
- Randomized Controlled Trial

4 MeSH Terms:
- Malaria, Cerebral/cerebrospinal fluid*
- Malaria, Cerebral/drug therapy
- Malaria, Cerebral/parasitology

PMID: 11865422 [PubMed - indexed for MEDLINE]

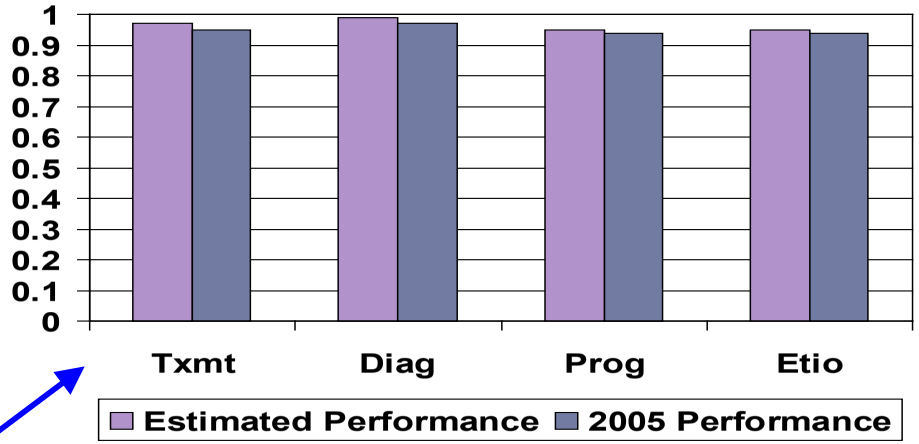
2. **Simple document** representations **(i.e.,** typically **→ stemmed and weighted words in title and abstract, Mesh terms if available;** occasionally **→ addition of Metamap CUIs, author info) as "**bag-of-words**"**

Models to categorize content and quality: Main idea



3. **Train SVM models that capture implicit categories of meaning or quality criteria**

4. **Evaluate models' performances**
- **with nested cross-validation or other appropriate error estimators**
- **use primarily AUC as well as other metrics (sensitivity, specificity, PPV, Precision/Recall curves, HIT curves, etc.)**



5. **Evaluate performance prospectively & compare to prior cross-validation estimates**

Models to categorize content and quality: Some notable results

Category	Average AUC	Range over n folds
Treatment	0.97*	0.96 - 0.98
Etiology	0.94*	0.89 – 0.95
Prognosis	0.95*	0.92 – 0.97
Diagnosis	0.95*	0.93 - 0.98

1. **SVM models have excellent ability to identify high-quality PubMed documents according to ACPJ gold standard**

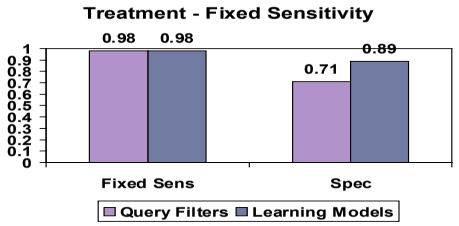
Method	Treatment - AUC	Etiology - AUC	Prognosis - AUC	Diagnosis - AUC
Google Pagerank	0.54	0.54	0.43	0.46
Yahoo Webranks	0.56	0.49	0.52	0.52
Impact Factor 2005	0.67	0.62	0.51	0.52
Web page hit count	0.63	0.63	0.58	0.57
Bibliometric Citation Count	0.76	0.69	0.67	0.60
Machine Learning Models	**0.96**	**0.95**	**0.95**	**0.95**

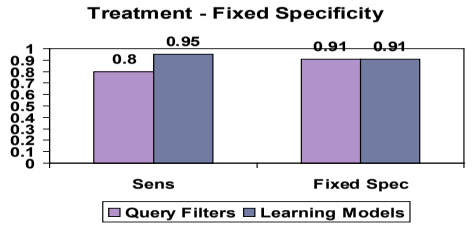
2. **SVM models have better classification performance than PageRank, Yahoo ranks, Impact Factor, Web Page hit counts, and bibliometric citation counts on the Web according to ACPJ gold standard**

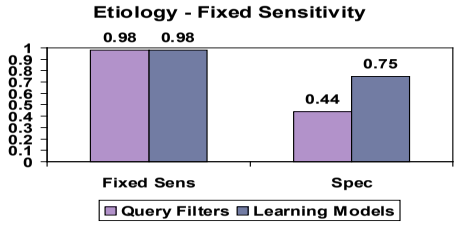
Models to categorize content and quality: Some notable results

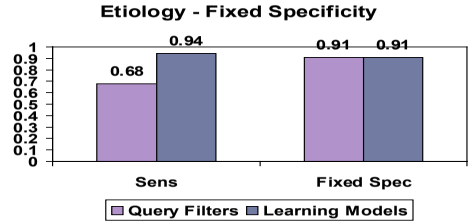
Gold standard: SSOAB	Area under the ROC curve*
SSOAB-specific filters	0.893
Citation Count	0.791
ACPJ Txmt-specific filters	0.548
Impact Factor (2001)	0.549
Impact Factor (2005)	0.558

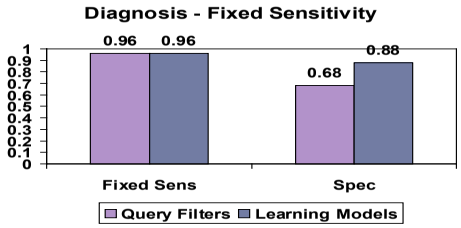
3. **SVM models have better classification performance than PageRank, Impact Factor and Citation count in Medline for SSOAB gold standard**

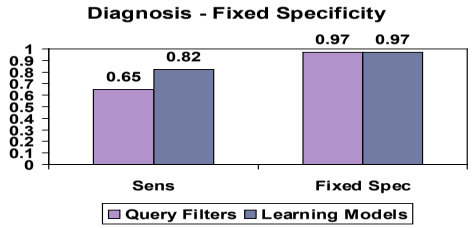


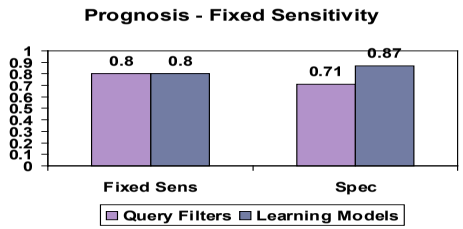


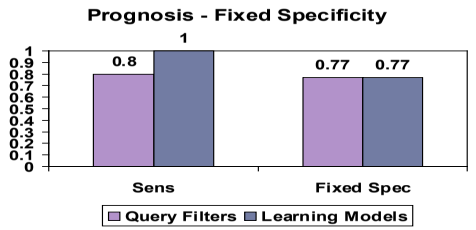








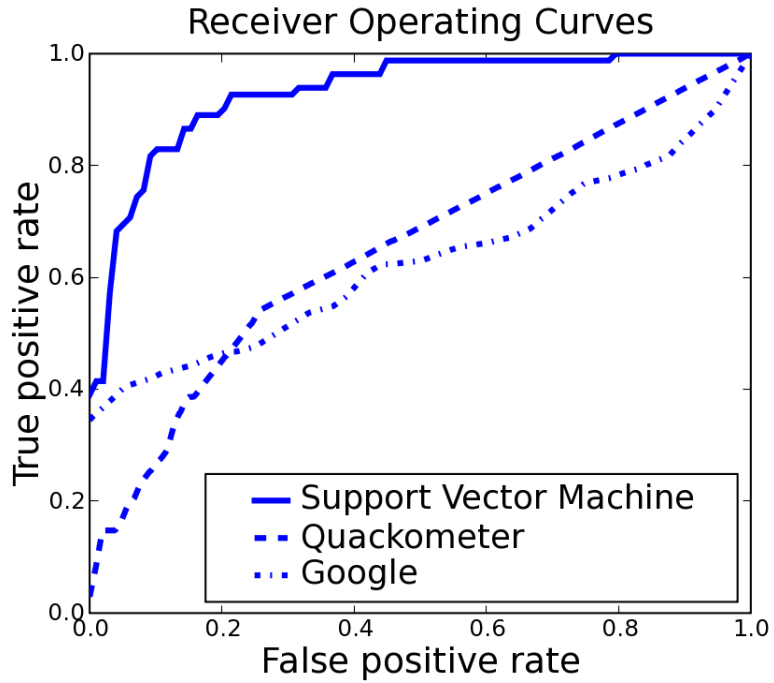




4. **SVM models have better sensitivity/specificity in PubMed than CQFs at comparable thresholds according to ACPJ gold standard**

Other applications of SVMs to text categorization

Model	Area Under the Curve
Machine Learning Models	0.93
Quackometer*	0.67
Google	0.63



1. **Identifying Web Pages with misleading treatment information according to special purpose gold standard (Quack Watch). SVM models outperform Quackometer and Google ranks in the tested domain of cancer treatment.**

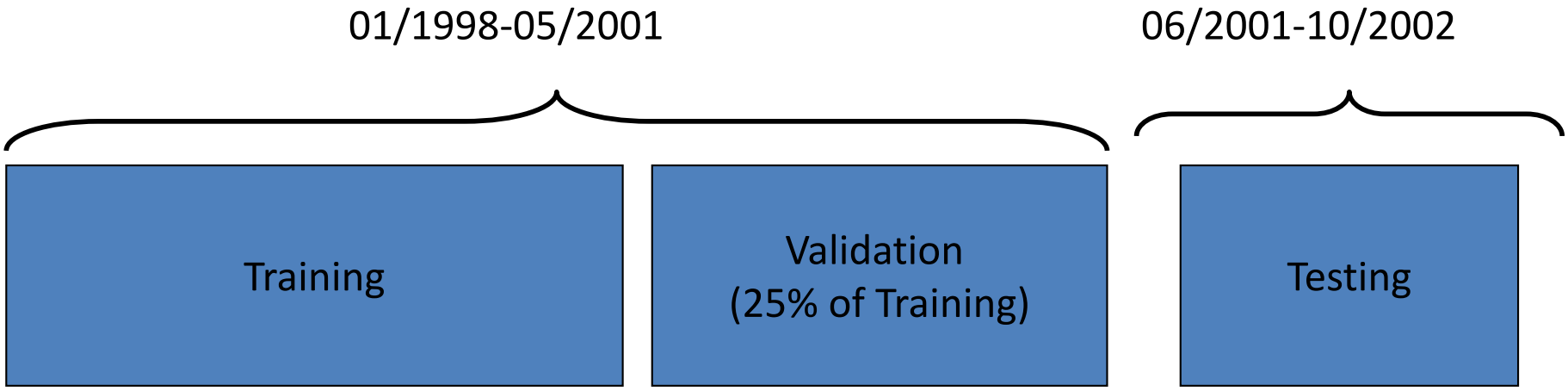
2. **Prediction of future paper citation counts (work of L. Fu and C.F. Aliferis, AMIA 2008)**

3. Prediction of clinical laboratory values

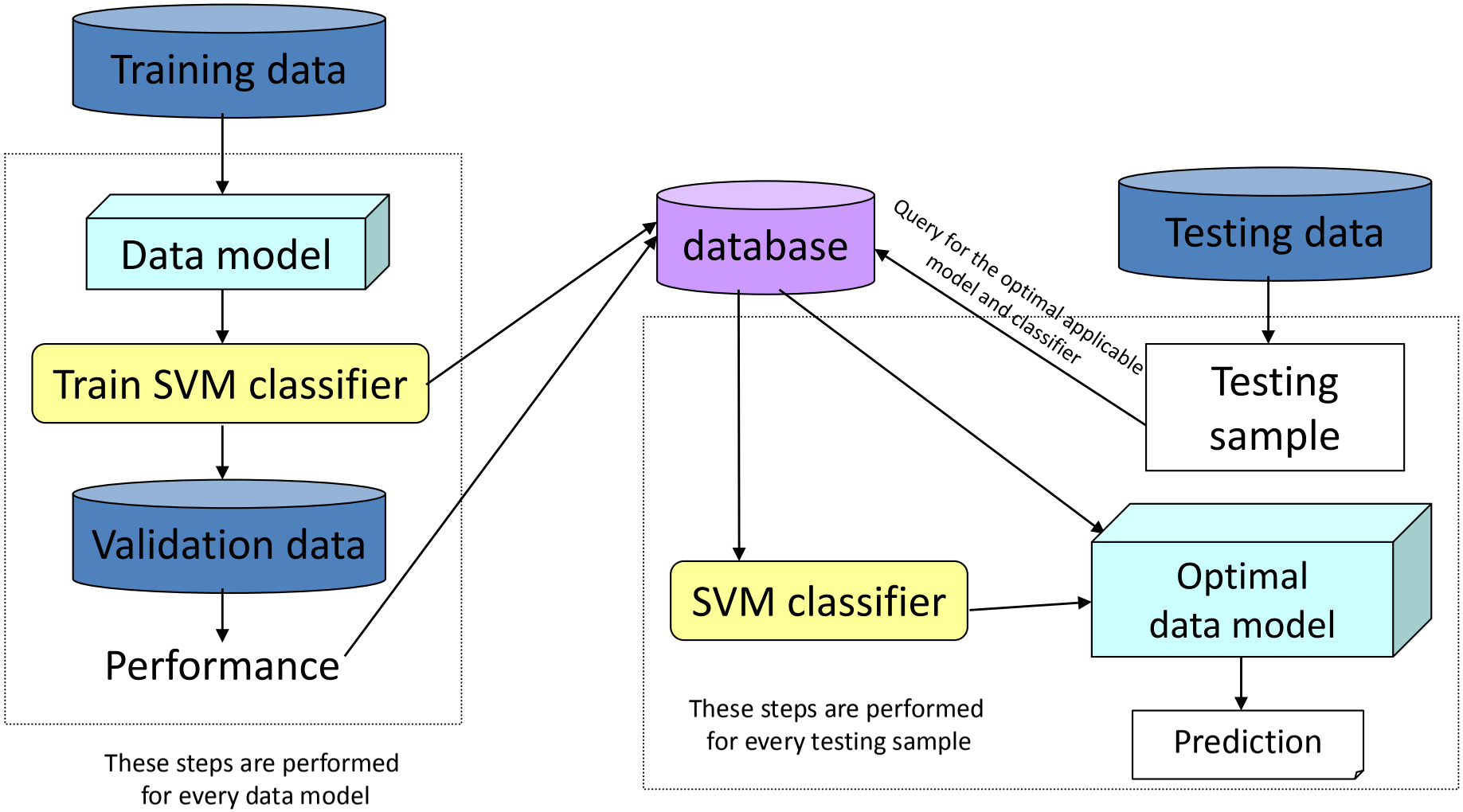
Dataset generation and experimental design

- StarPanel database contains ~$8 \cdot 10^6$ lab measurements of ~100,000 in-patients from Vanderbilt University Medical Center.
- Lab measurements were taken between 01/1998 and 10/2002.

For each combination of lab test and normal range, we generated the following datasets.



Query-based approach for prediction of clinical cab values



Classification results

Including cases with **K**=0 (i.e. samples with no prior lab measurements)

Area under ROC curve (without feature selection)

Laboratory test	*Range of normal values*					
	>1	**<99**	**[1, 99]**	**>2.5**	**<97.5**	**[2.5, 97.5]**
BUN	75.9%	**93.4%**	68.5%	81.8%	**92.2%**	66.9%
Ca	67.5%	80.4%	55.0%	77.4%	70.8%	60.0%
Calo	63.5%	52.9%	58.8%	46.4%	66.3%	58.7%
CO2	77.3%	**88.0%**	53.4%	77.5%	**90.5%**	58.1%
Creat	62.2%	**88.4%**	83.5%	**88.4%**	**94.9%**	83.8%
Mg	58.4%	71.8%	64.2%	67.0%	72.5%	62.1%
Osmol	77.9%	64.8%	65.2%	79.2%	82.4%	71.5%
PCV	62.3%	**91.6%**	69.7%	76.5%	**84.6%**	70.2%
Phos	70.8%	75.4%	60.4%	68.0%	81.8%	65.9%

Excluding cases with **K**=0 (i.e. samples with no prior lab measurements)

Area under ROC curve (without feature selection)

Laboratory test	*Range of normal values*					
	>1	**<99**	**[1, 99]**	**>2.5**	**<97.5**	**[2.5, 97.5]**
BUN	80.4%	**99.1%**	76.6%	**87.1%**	**98.2%**	70.7%
Ca	72.8%	**93.4%**	55.6%	81.4%	81.4%	63.4%
Calo	74.1%	60.0%	50.1%	64.7%	72.3%	57.7%
CO2	82.0%	**93.6%**	59.8%	84.4%	**94.5%**	56.3%
Creat	62.8%	**97.7%**	**89.1%**	**91.5%**	**98.1%**	**87.7%**
Mg	56.9%	70.0%	49.1%	58.6%	76.9%	59.1%
Osmol	50.9%	60.8%	60.8%	**91.0%**	**90.5%**	68.0%
PCV	74.9%	**99.2%**	66.3%	80.9%	80.6%	67.1%
Phos	74.5%	**93.6%**	64.4%	71.7%	**92.2%**	69.7%

A total of 84,240 SVM classifiers were built for 16,848 possible data models.

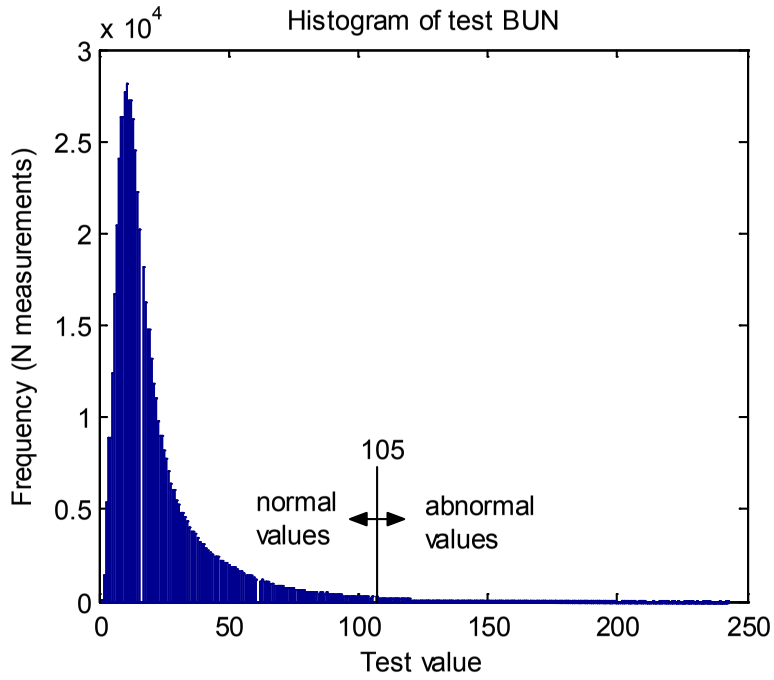
Improving predictive power and parsimony of a BUN model using feature selection

Model description

Test name	BUN
Range of normal values	< 99 perc.
Data modeling	SRT
Number of previous measurements	5
Use variables corresponding to hospitalization units?	Yes
Number of prior hospitalizations used	2

Dataset description

	N samples (total)	N abnormal samples	N variables
Training set	3749	78	3442
Validation set	1251	27	
Testing set	836	16	



Classification performance (area under ROC curve)

	All	RFE_Linear	RFE_Poly	HITON_PC	HITON_MB
Validation set	95.29%	98.78%	98.76%	99.12%	98.90%
Testing set	94.72%	99.66%	99.63%	99.16%	99.05%
Number of features	**3442**	**26**	**3**	**11**	**17**

Classification performance (area under ROC curve)

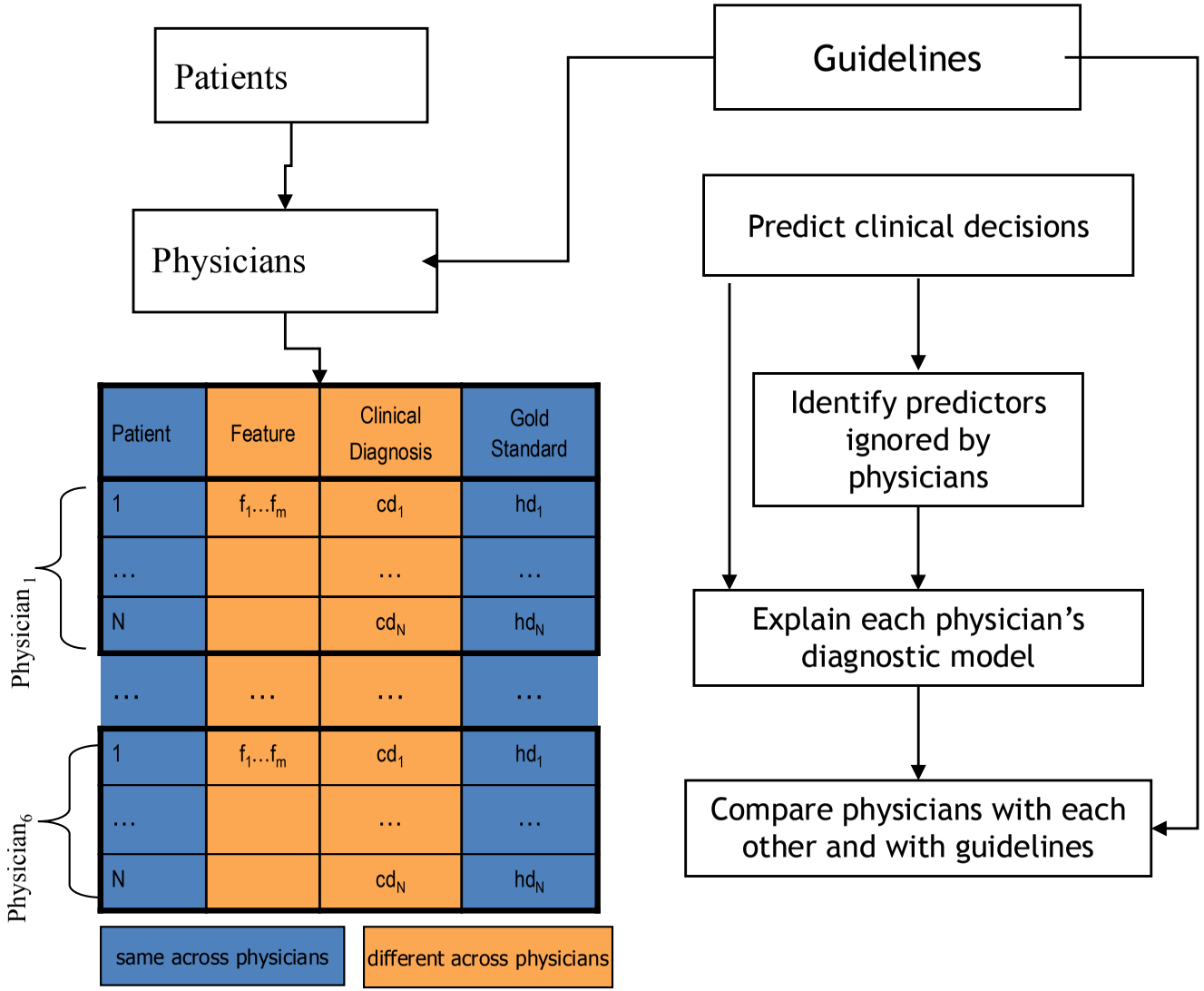
	All	RFE_Linear	RFE_Poly	HITON_PC	HITON_MB
Validation set	95.29%	98.78%	98.76%	99.12%	98.90%
Testing set	94.72%	99.66%	99.63%	99.16%	99.05%
Number of features	**3442**	**26**	**3**	**11**	**17**

Features

1	LAB: PM_1(BUN)	LAB: PM_1(BUN)	LAB: PM_1(BUN)	LAB: PM_1(BUN)
2	LAB: PM_2(Cl)	LAB: Indicator(PM_1(Mg))	LAB: PM_5(Creat)	LAB: PM_5(Creat)
3	LAB: DT(PM_3(K))	LAB: Test Unit NO_TEST_MEASUREMENT (Test Calo, PM 1)	LAB: PM_1(Phos)	LAB: PM_3(PCV)
4	LAB: DT(PM_3(Creat))		LAB: Indicator(PM_1(BUN))	LAB: PM_1(Mg)
5	LAB: Test Unit J018 (Test Ca, PM 3)		LAB: Indicator(PM_5(Creat))	LAB: PM_1(Phos)
6	LAB: DT(PM_4(Cl))		LAB: Indicator(PM_1(Mg))	LAB: Indicator(PM_4(Creat))
7	LAB: DT(PM_3(Mg))		LAB: DT(PM_4(Creat))	LAB: Indicator(PM_5(Creat))
8	LAB: PM_1(Cl)		LAB: Test Unit 7SCC (Test Ca, PM 1)	LAB: Indicator(PM_3(PCV))
9	LAB: PM_3(Gluc)		LAB: Test Unit RADR (Test Ca, PM 5)	LAB: Indicator(PM_1(Phos))
10	LAB: DT(PM_1(CO2))		LAB: Test Unit 7SMI (Test PCV, PM 4)	LAB: DT(PM_4(Creat))
11	LAB: DT(PM_4(Gluc))		DEMO: Gender	LAB: Test Unit 11NM (Test BUN, PM 2)
12	LAB: PM_3(Mg)			LAB: Test Unit 7SCC (Test Ca, PM 1)
13	LAB: DT(PM_5(Mg))			LAB: Test Unit RADR (Test Ca, PM 5)
14	LAB: PM_1(PCV)			LAB: Test Unit 7SMI (Test PCV, PM 4)
15	LAB: PM_2(BUN)			LAB: Test Unit CCL (Test Phos, PM 1)
16	LAB: Test Unit 11NM (Test PCV, PM 2)			DEMO: Gender
17	LAB: Test Unit 7SCC (Test Mg, PM 3)			DEMO: Age
18	LAB: DT(PM_2(Phos))			
19	LAB: DT(PM_3(CO2))			
20	LAB: DT(PM_2(Gluc))			
21	LAB: DT(PM_5(Calo))			
22	DEMO: Hospitalization Unit TVOS			
23	LAB: PM_1(Phos)			
24	LAB: PM_2(Phos)			
25	LAB: Test Unit 11NM (Test K, PM 5)			
26	LAB: Test Unit VHR (Test Calo, PM 1)			

4. Modeling clinical judgment

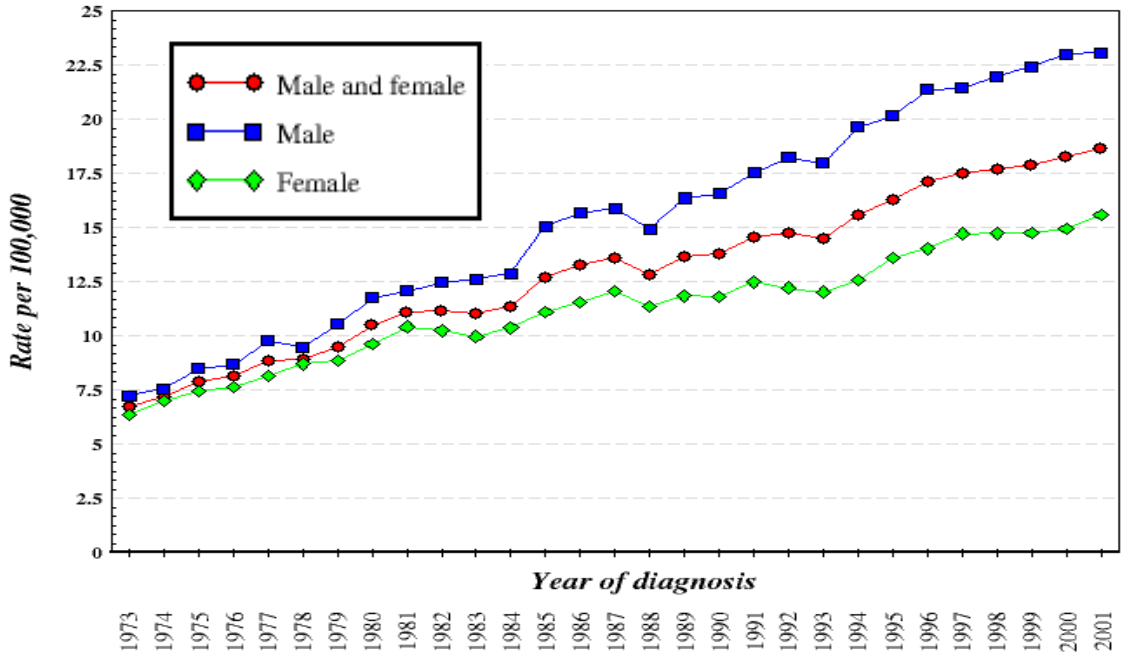
Methodological framework and study outline



Clinical context of experiment

Malignant melanoma is the most dangerous form of skin cancer

Incidence & mortality have been constantly increasing in the last decades.



Physicians and patients

Patients → N=177

76 melanomas - 101 nevi

Dermatologists → N = 6

3 experts - 3 non-experts

Data collection:

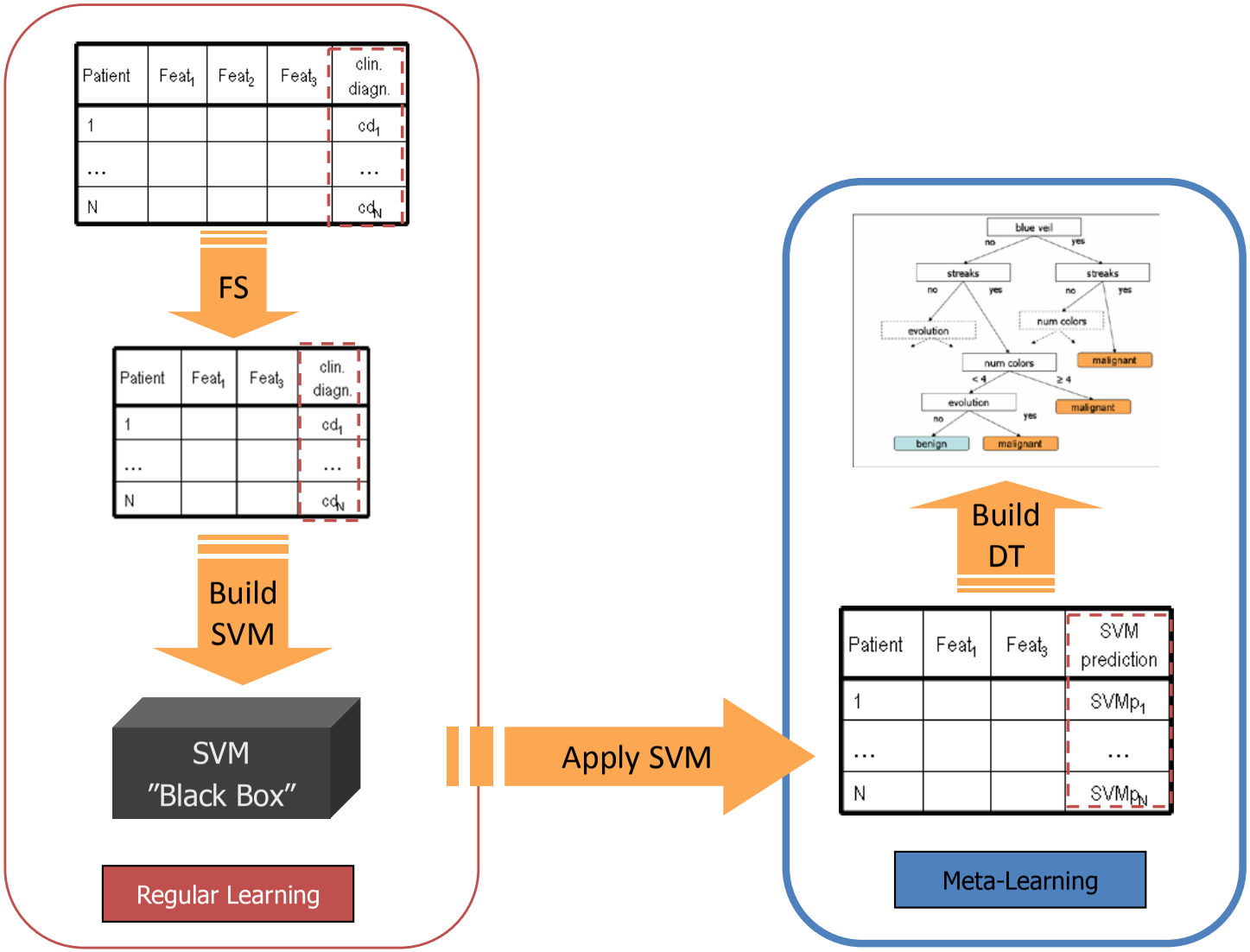
Patients seen prospectively, from 1999 to 2002 at Department of Dermatology, S.Chiara Hospital, Trento, Italy

inclusion criteria: histological diagnosis and >1 digital image available

Diagnoses made in 2004

Features			
Lesion location	Family history of melanoma	Irregular Border	Streaks (radial streaming, pseudopods)
Max-diameter	Fitzpatrick's Photo-type	Number of colors	Slate-blue veil
Min-diameter	Sunburn	Atypical pigmented network	Whitish veil
Evolution	Ephelis	Abrupt network cut-off	Globular elements
Age	Lentigos	Regression-Erythema	Comedo-like openings, milia-like cysts
Gender	Asymmetry	Hypo-pigmentation	Telangiectasia

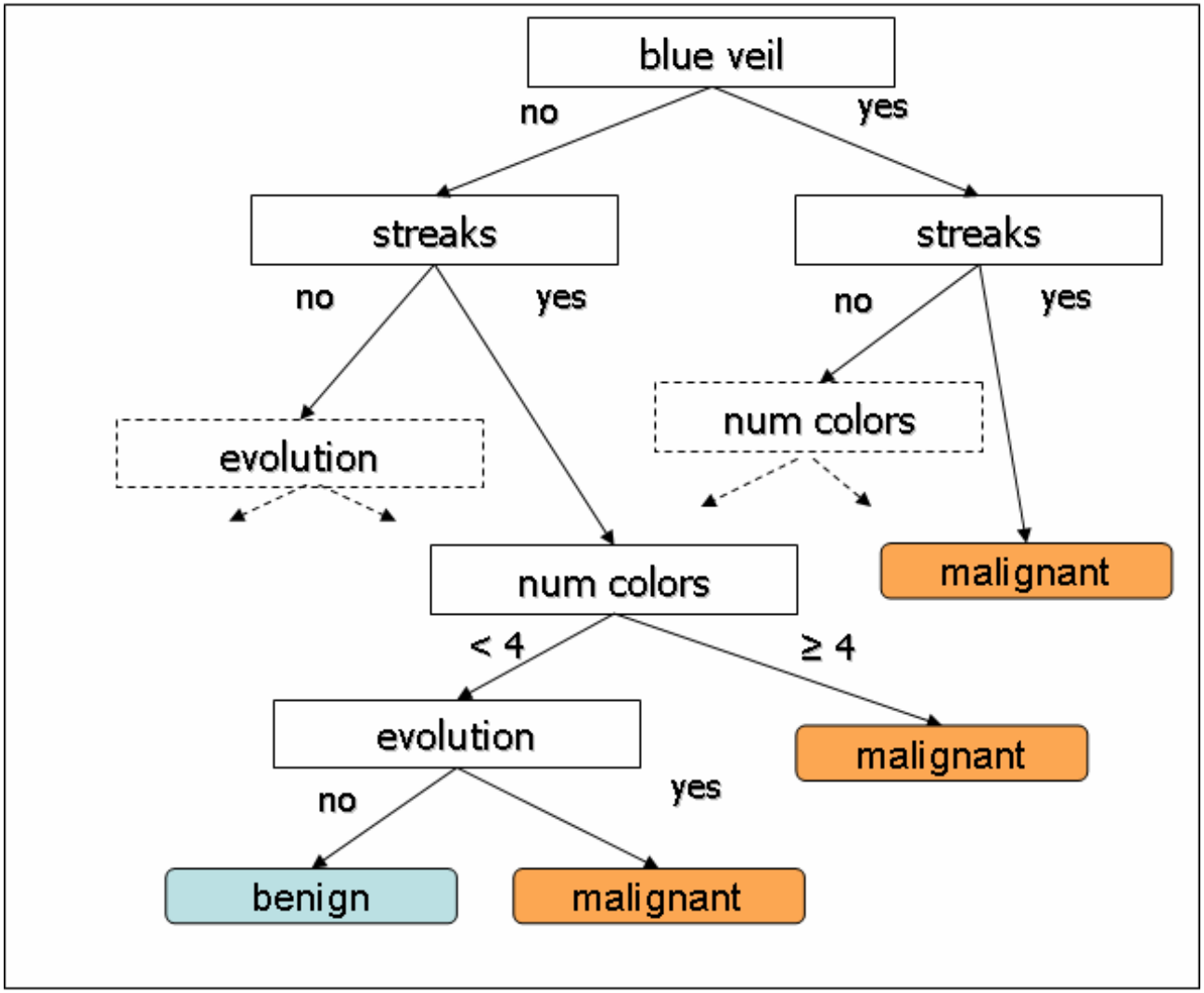
Method to explain physician-specific SVM models



Results: Predicting physicians' judgments

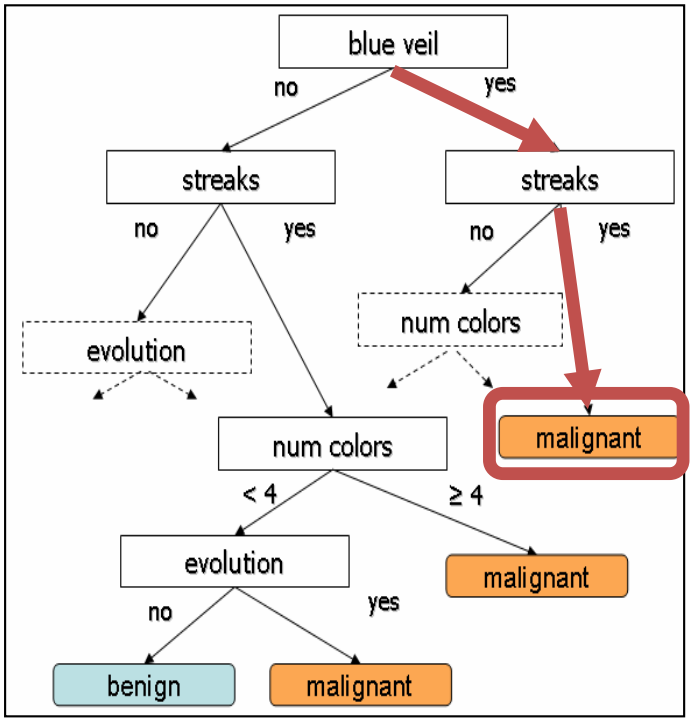
Physicians	All (features)	HITON_PC (features)	HITON_MB (features)	RFE (features)
Expert 1	**0.94 (24)**	0.92 (4)	0.92 (5)	0.95 (14)
Expert 2	**0.92 (24)**	0.89 (7)	0.90 (7)	0.90 (12)
Expert 3	**0.98 (24)**	0.95 (4)	0.95 (4)	0.97 (19)
NonExpert 1	**0.92 (24)**	0.89 (5)	0.89 (6)	0.90 (22)
NonExpert 2	**1.00 (24)**	0.99 (6)	0.99 (6)	0.98 (11)
NonExpert 3	**0.89 (24)**	0.89 (4)	0.89 (6)	0.87 (10)

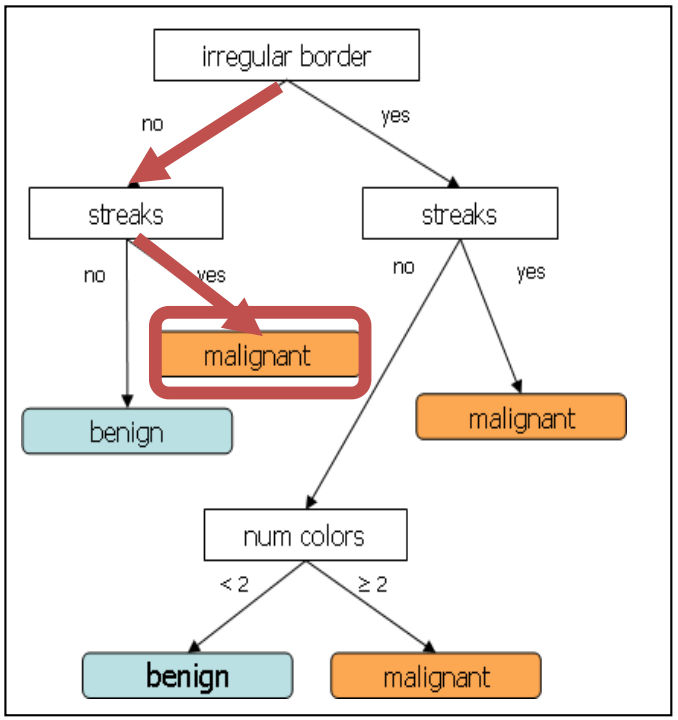
Results: Physician-specific models



Results: Explaining physician agreement

	Blue veil	irregular border	streaks
Patient *001*	yes	no	yes



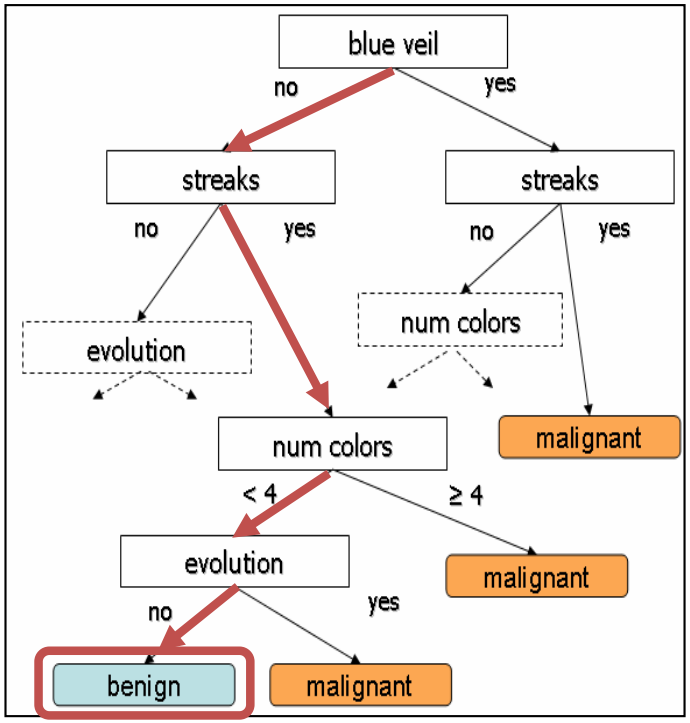


Expert 1
AUC=0.92
R^2=99%

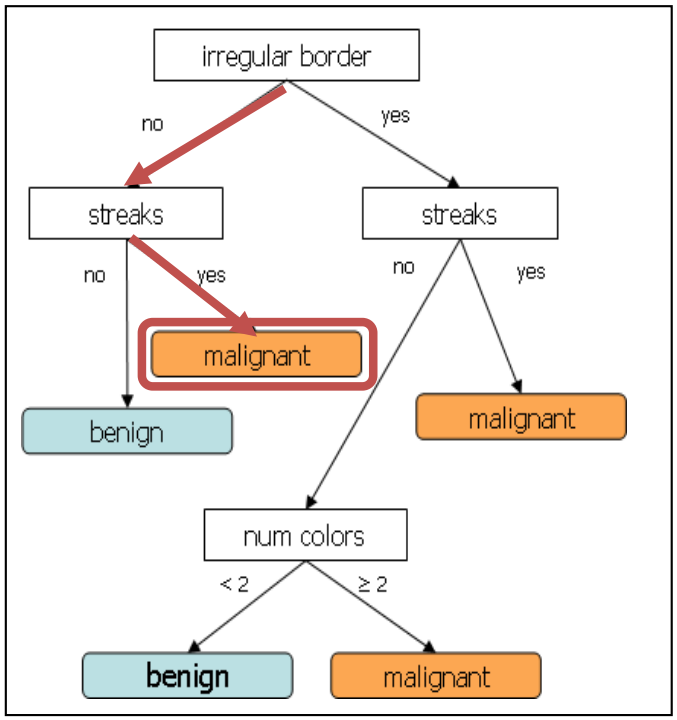
Expert 3
AUC=0.95
R^2=99%

Results: Explain physician disagreement

	Blue veil	irregular border	streaks	number of colors	evolution
Patient *002*	no	no	yes	3	no



Expert 1
AUC=0.92
R^2=99%



Expert 3
AUC=0.95
R^2=99%

Results: Guideline compliance

Physician	Reported guidelines	Compliance
Experts1,2,3, non-expert 1	Pattern analysis	**Non-compliant:** they ignore the majority of features (17 to 20) recommended by pattern analysis.
Non expert 2	ABCDE rule	**Non compliant:** asymmetry, irregular border and evolution are ignored.
Non expert 3	Non-standard. Reports using 7 features	**Non compliant:** 2 out of 7 reported features are ignored while some non-reported ones are not

On the contrary: In all guidelines, the more predictors present, the higher the likelihood of melanoma. All physicians were **compliant** with this principle.

5. Using SVMs for feature selection

Feature selection methods

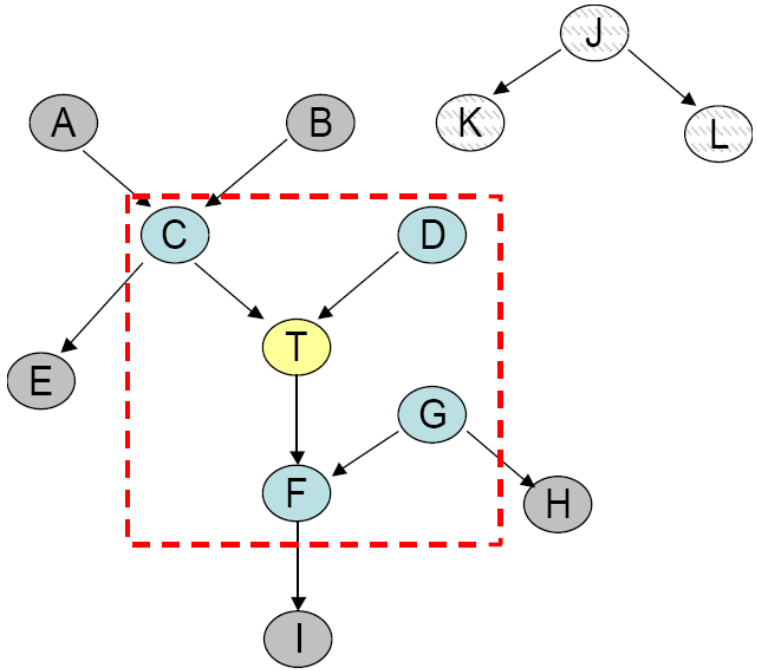
Feature selection methods (non-causal)

- **SVM-RFE** ← This is an SVM-based feature selection method
- Univariate + wrapper
- Random forest-based
- LARS-Elastic Net
- RELIEF + wrapper
- L0-norm
- Forward stepwise feature selection
- No feature selection

Causal feature selection methods

- **HITON-PC** ← This method outputs a Markov blanket of the response variable (under assumptions)
- HITON-MB
- IAMB
- BLCD
- K2MB

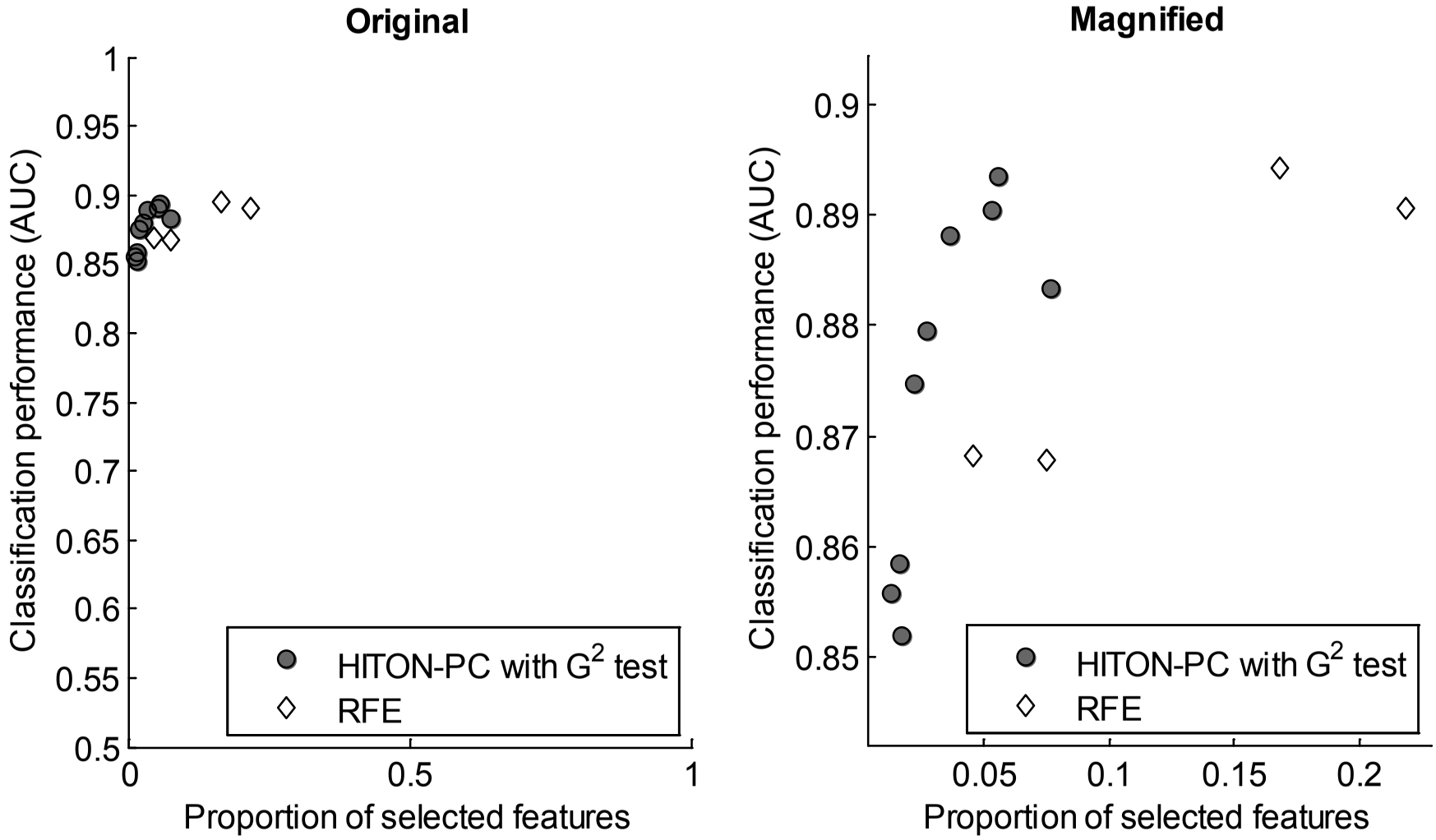
…



13 real datasets were used to evaluate feature selection methods

Dataset name	*Domain*	*Number of variables*	*Number of samples*	*Target*	*Data type*
Infant_Mortality	Clinical	86	5,337	Died within the first year	discrete
Ohsumed	Text	14,373	5,000	Relevant to nenonatal diseases	continuous
ACPJ_Etiology	Text	28,228	15,779	Relevant to eitology	continuous
Lymphoma	Gene expression	7,399	227	3-year survival: dead vs. alive	continuous
Gisette	Digit recognition	5,000	7,000	Separate 4 from 9	continuous
Dexter	Text	19,999	600	Relevant to corporate acquisitions	continuous
Sylva	Ecology	216	14,394	Ponderosa pine vs. everything else	continuous & discrete
Ovarian_Cancer	Proteomics	2,190	216	Cancer vs. normals	continuous
Thrombin	Drug discovery	139,351	2,543	Binding to thromin	discrete (binary)
Breast_Cancer	Gene expression	17,816	286	Estrogen-receptor positive (ER+) vs. ER-	continuous
Hiva	Drug discovery	1,617	4,229	Activity to AIDS HIV infection	discrete (binary)
Nova	Text	16,969	1,929	Separate politics from religion topics	discrete (binary)
Bankruptcy	Financial	147	7,063	Personal bankruptcy	continuous & discrete

Classification performance vs. proportion of selected features



Statistical comparison of predictivity and reduction of features

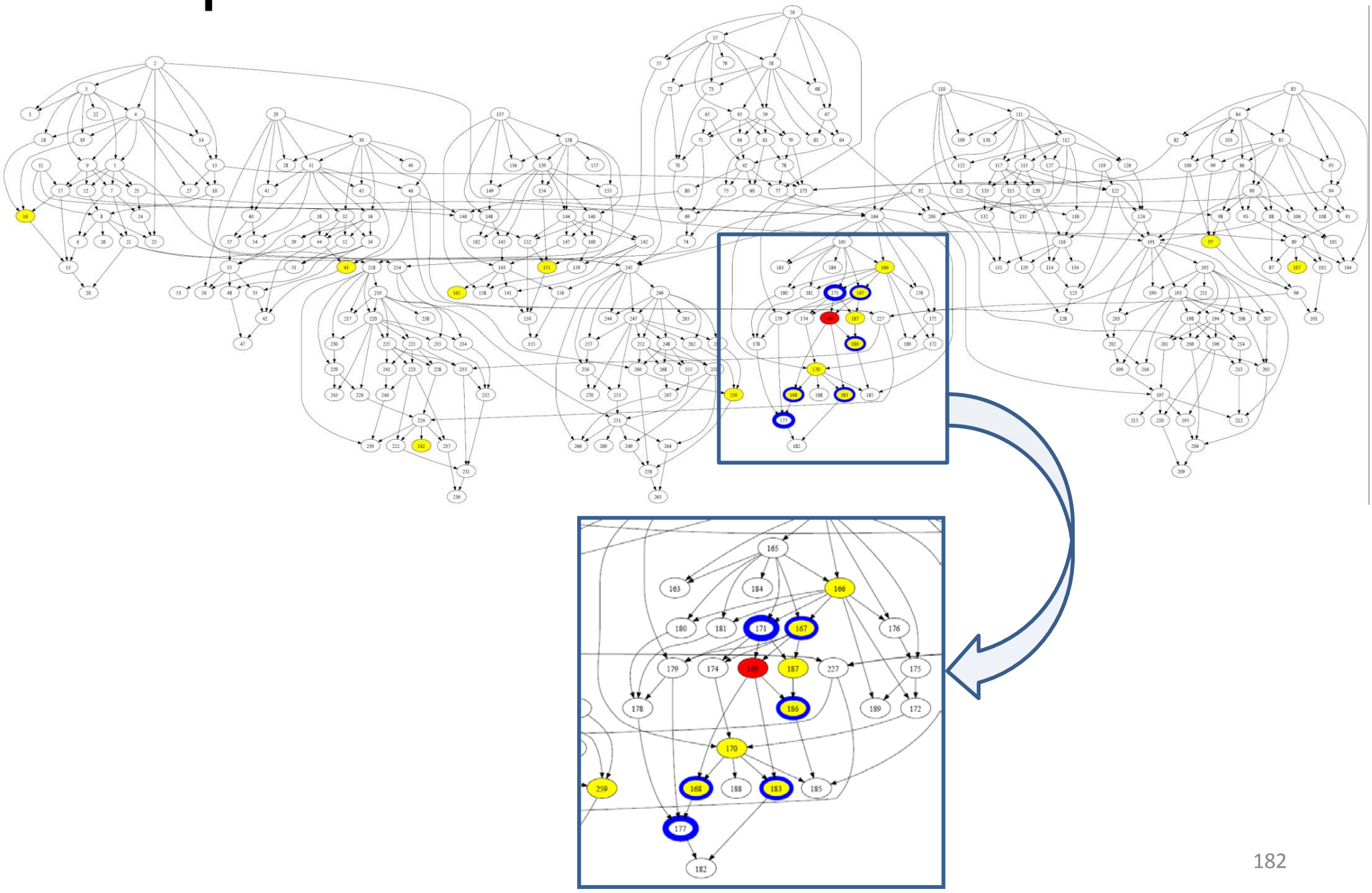
	Predicitivity		*Reduction*	
	P-value	Nominal winner	P-value	Nominal winner
SVM-RFE (4 variants)	0.9754	SVM-RFE	**0.0046**	HITON-PC
	0.8030	SVM-RFE	**0.0042**	HITON-PC
	0.1312	HITON-PC	0.3634	HITON-PC
	0.1008	HITON-PC	0.6816	SVM-RFE

- Null hypothesis: SVM-RFE and HITON-PC perform the same;
- Use permutation-based statistical test with alpha = 0.05.

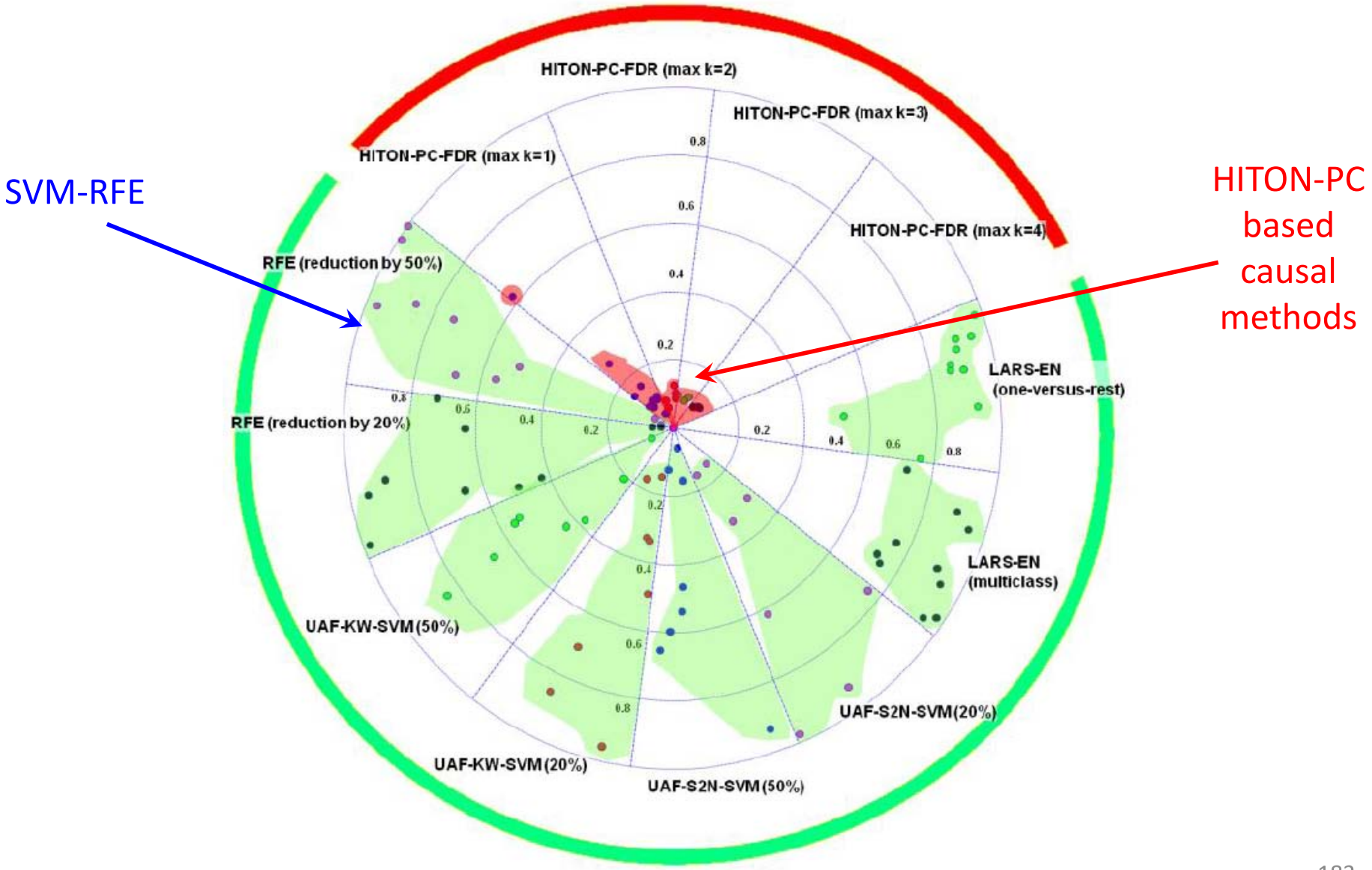
Simulated datasets with known causal structure used to compare algorithms

Bayesian network	Number of variables	Training samples	Number of selected targets
Child10	200	5 x 200, 5 x 500, 1 x 5000	10
Insurance10	270	5 x 200, 5 x 500, 1 x 5000	10
Alarm10	370	5 x 200, 5 x 500, 1 x 5000	10
Hailfinder10	560	5 x 200, 5 x 500, 1 x 5000	10
Munin	189	5 x 500, 1 x 5000	6
Pigs	441	5 x 200, 5 x 500, 1 x 5000	10
Link	724	5 x 200, 5 x 500, 1 x 5000	10
Lung_Cancer	800	5 x 200, 5 x 500, 1 x 5000	11
Gene	801	5 x 200, 5 x 500, 1 x 5000	11

Comparison of SVM-RFE and HITON-PC



Comparison of all methods in terms of causal graph distance

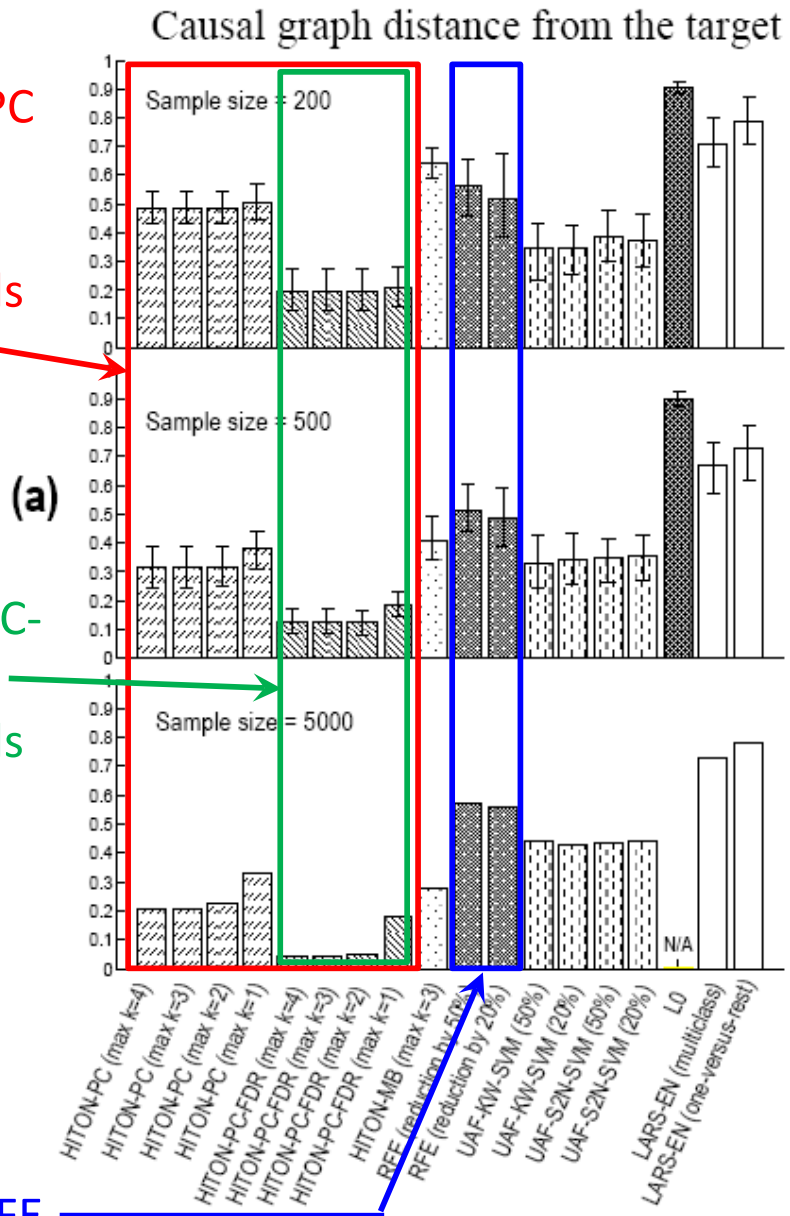


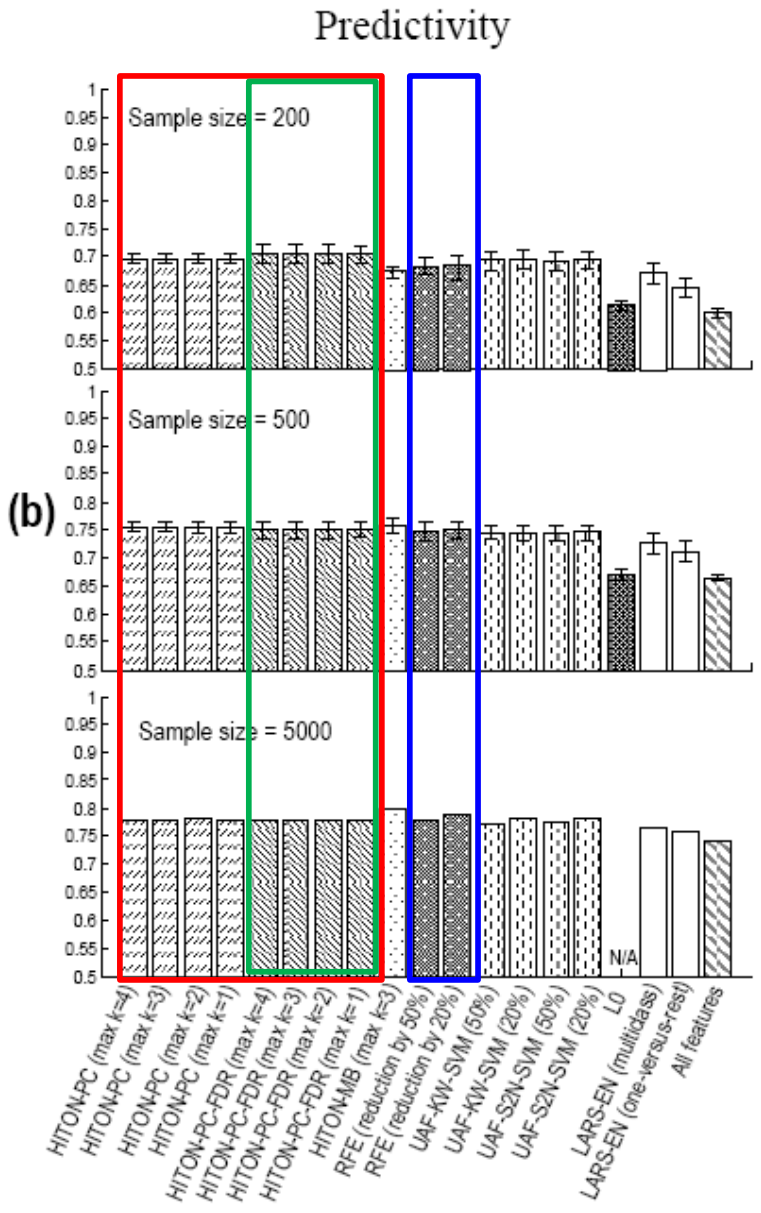
Summary results

HITON-PC based causal methods

HITON-PC-FDR methods

SVM-RFE





Statistical comparison of graph distance

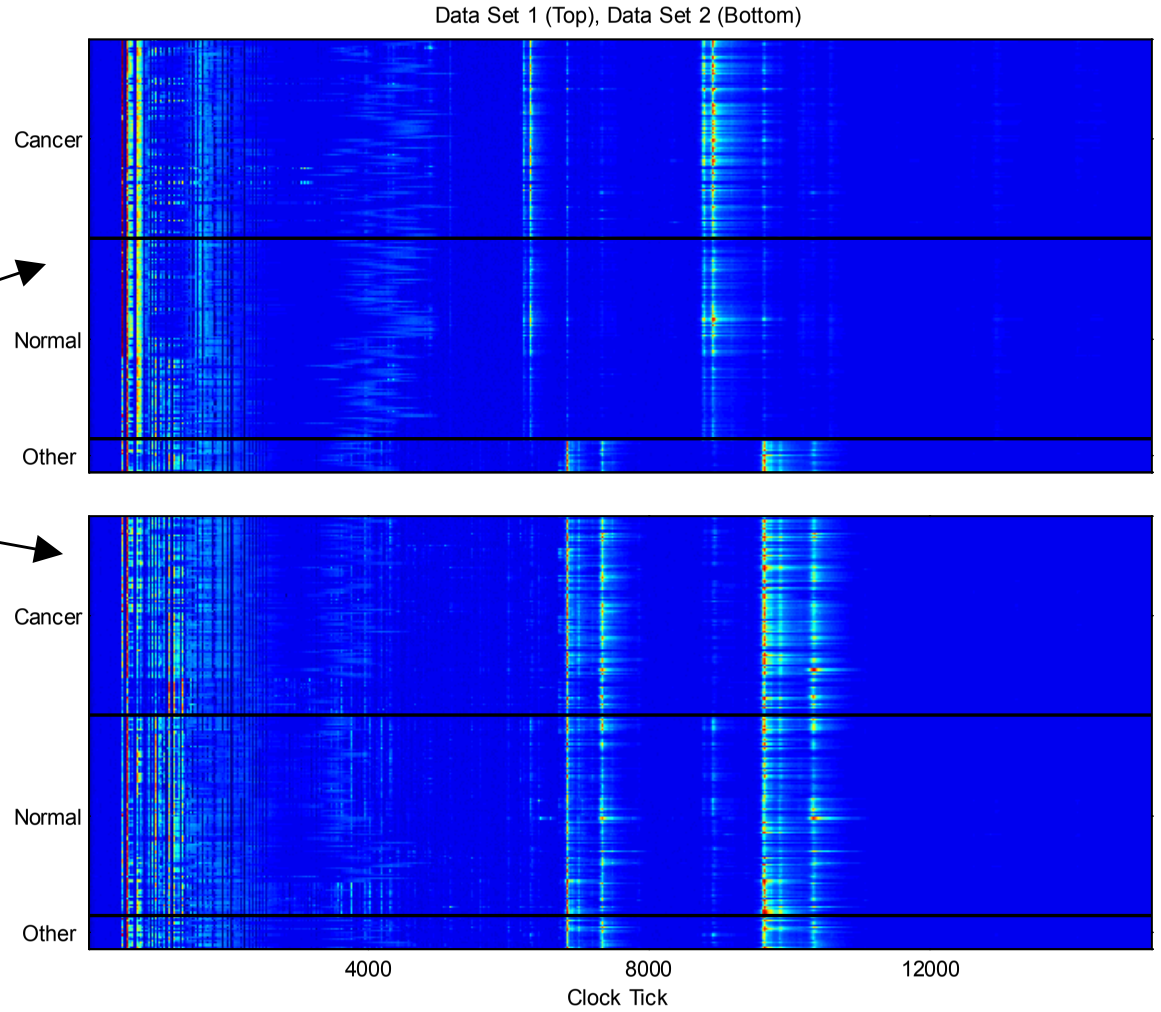
	Sample size = 200		Sample size = 500		Sample size = 5000	
Comparison	P-value	Nominal winner	P-value	Nominal winner	P-value	Nominal winner
average **HITON-PC-FDR** with G^2 test vs. average **SVM-RFE**	**<0.0001**	HITON-PC-FDR	**0.0028**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR

- Null hypothesis: SVM-RFE and HITON-PC-FDR perform the same;
- Use permutation-based statistical test with alpha = 0.05.

6. Outlier detection in ovarian cancer proteomics data

Ovarian cancer data

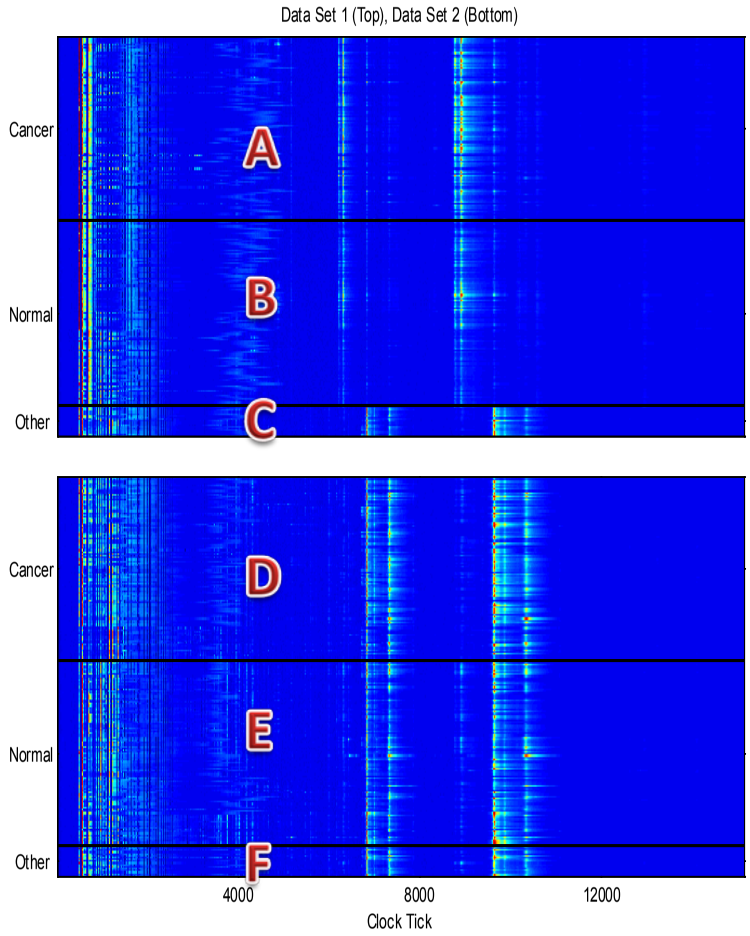
Same set of 216 patients, obtained using the Ciphergen H4 ProteinChip array (dataset 1) and using the Ciphergen WCX2 ProteinChip array (dataset 2).



The gross break at the “benign disease” juncture in dataset 1 and the similarity of the profiles to those in dataset 2 suggest change of protocol in the middle of the first experiment.

Experiments with one-class SVM

Assume that sets {A, B} are normal and {C, D, E, F} are outliers. Also, assume that we do not know what are normal and outlier samples.



- Experiment 1: Train one-class SVM on {A, B, C} and test on {A, B, C}: Area under ROC curve = **0.98**
- Experiment 2: Train one-class SVM on {A, C} and test on {B, D, E, F}: Area under ROC curve = **0.98**

Software

Interactive media and animations

SVM Applets

- http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml
- http://www.smartlab.dibe.unige.it/Files/sw/Applet%20SVM/svmapplet.html
- http://www.eee.metu.edu.tr/~alatan/Courses/Demo/AppletSVM.html
- http://www.dsl-lab.org/svm_tutorial/demo.html (requires Java 3D)

Animations

- Support Vector Machines:
  http://www.cs.ust.hk/irproj/Regularization%20Path/svmKernelpath/2moons.avi
  http://www.cs.ust.hk/irproj/Regularization%20Path/svmKernelpath/2Gauss.avi
  http://www.youtube.com/watch?v=3liCbRZPrZA
- Support Vector Regression:
  http://www.cs.ust.hk/irproj/Regularization%20Path/movie/ga0.5lam1.avi

Several SVM implementations for beginners

- GEMS: http://www.gems-system.org
- Weka: http://www.cs.waikato.ac.nz/ml/weka/
- Spider (for Matlab): http://www.kyb.mpg.de/bs/people/spider/
- CLOP (for Matlab): http://clopinet.com/CLOP/

Several SVM implementations for intermediate users

- LibSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - General purpose
  - Implements binary SVM, multiclass SVM, SVR, one-class SVM
  - Command-line interface
  - Code/interface for C/C++/C#, Java, Matlab, R, Python, Pearl
- SVMLight: http://svmlight.joachims.org/
  - General purpose (designed for text categorization)
  - Implements binary SVM, multiclass SVM, SVR
  - Command-line interface
  - Code/interface for C/C++, Java, Matlab, Python, Pearl

More software links at http://www.support-vector-machines.org/SVM_soft.html and http://www.kernel-machines.org/software

Conclusions

Strong points of SVM-based learning methods

- Empirically achieve excellent results in high-dimensional data with very few samples
- Internal capacity control to avoid overfitting
- Can learn both simple linear and very complex nonlinear functions by using “kernel trick”
- Robust to outliers and noise (use “slack variables”)
- Convex QP optimization problem (thus, it has global minimum and can be solved efficiently)
- Solution is defined only by a small subset of training points (“support vectors”)
- Number of free parameters is bounded by the number of support vectors and not by the number of variables
- Do not require direct access to data, work only with dot-products of data-points.

Weak points of SVM-based learning methods

- Measures of uncertainty of parameters are not currently well-developed
- Interpretation is less straightforward than classical statistics
- Lack of parametric statistical significance tests
- Power size analysis and research design considerations are less developed than for classical statistics

Bibliography

Part 1: Support vector machines for binary classification: classical formulation

- Boser BE, Guyon IM, Vapnik VN: **A training algorithm for optimal margin classifiers**. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT)* 1992**:**144-152.
- Burges CJC: **A tutorial on support vector machines for pattern recognition**. *Data Mining and Knowledge Discovery* 1998, **2:**121-167.
- Cristianini N, Shawe-Taylor J: *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press; 2000.
- Hastie T, Tibshirani R, Friedman JH: *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer; 2001.
- Herbrich R: *Learning kernel classifiers: theory and algorithms*. Cambridge, Mass: MIT Press; 2002.
- Schölkopf B, Burges CJC, Smola AJ: *Advances in kernel methods: support vector learning*. Cambridge, Mass: MIT Press; 1999.
- Shawe-Taylor J, Cristianini N: *Kernel methods for pattern analysis*. Cambridge, UK: Cambridge University Press; 2004.
- Vapnik VN: *Statistical learning theory*. New York: Wiley; 1998.

Part 1: *Basic principles of statistical machine learning*

- Aliferis CF, Statnikov A, Tsamardinos I: **Challenges in the analysis of mass-throughput data: a technical commentary from the statistical machine learning perspective.** *Cancer Informatics* 2006, **2:**133-162.
- Duda RO, Hart PE, Stork DG: *Pattern classification*. 2nd edition. New York: Wiley; 2001.
- Hastie T, Tibshirani R, Friedman JH: *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer; 2001.
- Mitchell T: *Machine learning*. New York, NY, USA: McGraw-Hill; 1997.
- Vapnik VN: *Statistical learning theory*. New York: Wiley; 1998.

Part 2: Model selection for SVMs

- Hastie T, Tibshirani R, Friedman JH: *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer; 2001.
- Kohavi R: **A study of cross-validation and bootstrap for accuracy estimation and model selection**. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)* 1995, **2:**1137-1145.
- Scheffer T: **Error estimation and model selection**. Ph.D.Thesis, Technischen Universität Berlin, School of Computer Science; 1999.
- Statnikov A, Tsamardinos I, Dosbayev Y, Aliferis CF: **GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data**. *Int J Med Inform* 2005, **74:**491-503.

Part 2: *SVMs for multicategory data*

- Crammer K, Singer Y: **On the learnability and design of output codes for multiclass problems**. *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT)* 2000.
- Platt JC, Cristianini N, Shawe-Taylor J: **Large margin DAGs for multiclass classification**. *Advances in Neural Information Processing Systems (NIPS)* 2000, **12:**547-553.
- Schölkopf B, Burges CJC, Smola AJ: *Advances in kernel methods: support vector learning*. Cambridge, Mass: MIT Press; 1999.
- Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy S: **A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis**. *Bioinformatics* 2005, **21:**631-643.
- Weston J, Watkins C: **Support vector machines for multi-class pattern recognition**. *Proceedings of the Seventh European Symposium On Artificial Neural Networks* 1999, **4:**6.

Part 2: Support vector regression

- Hastie T, Tibshirani R, Friedman JH: *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer; 2001.
- Smola AJ, Schölkopf B: **A tutorial on support vector regression**. *Statistics and Computing* 2004, **14:**199-222.

Part 2: Novelty detection with SVM-based methods and Support Vector Clustering

- Scholkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC: **Estimating the Support of a High-Dimensional Distribution**. *Neural Computation* 2001, **13:**1443-1471.
- Tax DMJ, Duin RPW: **Support vector domain description**. *Pattern Recognition Letters* 1999, **20:**1191-1199.
- Hur BA, Horn D, Siegelmann HT, Vapnik V: **Support vector clustering.** *Journal of Machine Learning Research* 2001, **2**:125–137.

Part 2: *SVM-based variable selection*

- Guyon I, Elisseeff A: **An introduction to variable and feature selection**. *Journal of Machine Learning Research* 2003, **3:**1157-1182.
- Guyon I, Weston J, Barnhill S, Vapnik V: **Gene selection for cancer classification using support vector machines**. *Machine Learning* 2002, **46:**389-422.
- Hardin D, Tsamardinos I, Aliferis CF: **A theoretical characterization of linear SVM-based feature selection**. *Proceedings of the Twenty First International Conference on Machine Learning (ICML)* 2004.
- Statnikov A, Hardin D, Aliferis CF: **Using SVM weight-based methods to identify causally relevant and non-causally relevant variables**. *Proceedings of the NIPS 2006 Workshop on Causality and Feature Selection* 2006.
- Tsamardinos I, Brown LE: **Markov Blanket-Based Variable Selection in Feature Space**. *Technical report DSL-08-01* 2008.
- Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V: **Feature selection for SVMs**. *Advances in Neural Information Processing Systems (NIPS)* 2000, **13:**668-674.
- Weston J, Elisseeff A, Scholkopf B, Tipping M: **Use of the zero-norm with linear models and kernel methods**. *Journal of Machine Learning Research* 2003, **3:**1439-1461.
- Zhu J, Rosset S, Hastie T, Tibshirani R: **1-norm support vector machines**. *Advances in Neural Information Processing Systems (NIPS)* 2004, **16**.

Part 2: *Computing posterior class probabilities for SVM classifiers*

- Platt JC: **Probabilistic outputs for support vector machines and comparison to regularized likelihood methods.** In *Advances in Large Margin Classifiers.* Edited by Smola A, Bartlett B, Scholkopf B, Schuurmans D. Cambridge, MA: MIT press; 2000.

Part 3: *Classification of cancer gene expression microarray data (Case Study I)*

- Diaz-Uriarte R, Alvarez de Andres S: **Gene selection and classification of microarray data using random forest**. *BMC Bioinformatics* 2006, **7:**3.
- Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy S: **A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis**. *Bioinformatics* 2005, **21:**631-643.
- Statnikov A, Tsamardinos I, Dosbayev Y, Aliferis CF: **GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data**. *Int J Med Inform* 2005, **74:**491-503.
- Statnikov A, Wang L, Aliferis CF: **A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification**. *BMC Bioinformatics* 2008, **9:**319.

Part 3: *Text Categorization in Biomedicine (Case Study 2)*

- Aphinyanaphongs Y, Aliferis CF: **Learning Boolean queries for article quality filtering**. *Medinfo 2004* 2004, **11:**263-267.
- Aphinyanaphongs Y, Tsamardinos I, Statnikov A, Hardin D, Aliferis CF: **Text categorization models for high-quality article retrieval in internal medicine**. *J Am Med Inform Assoc* 2005, **12:**207-216.
- Aphinyanaphongs Y, Statnikov A, Aliferis CF: **A comparison of citation metrics to machine learning filters for the identification of high quality MEDLINE documents**. *J Am Med Inform Assoc* 2006, **13:**446-455.
- Aphinyanaphongs Y, Aliferis CF: **Prospective validation of text categorization models for indentifying high-quality content-specific articles in PubMed**. *AMIA 2006 Annual Symposium Proceedings* 2006.
- Aphinyanaphongs Y, Aliferis C: **Categorization Models for Identifying Unproven Cancer Treatments on the Web**. *MEDINFO* 2007.
- Fu L, Aliferis C: **Models for Predicting and Explaining Citation Count of Biomedical Articles**. *AMIA 2008 Annual Symposium Proceedings* 2008.

Part 3: *Modeling clinical judgment*

(Case Study 4)

- Sboner A, Aliferis CF: **Modeling clinical judgment and implicit guideline compliance in the diagnosis of melanomas using machine learning**. *AMIA 2005 Annual Symposium Proceedings* 2005:664-668.

Part 3: *Using SVMs for feature selection*

(Case Study 5)

- Aliferis CF, Statnikov A, Tsamardinos I, Mani S, Koutsoukos XD: **Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification. Part II: Analysis and Extensions**. *Journal of Machine Learning Research* 2008.
- Aliferis CF, Statnikov A, Tsamardinos I, Mani S, Koutsoukos XD: **Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification. Part I: Algorithms and Empirical Evaluation**. *Journal of Machine Learning Research* 2008.

Part 3: *Outlier detection in ovarian cancer proteomics data (Case Study 6)*

- Baggerly KA, Morris JS, Coombes KR: **Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments**. *Bioinformatics* 2004, **20:**777-785.
- Petricoin EF, Ardekani AM, Hitt BA, Levine PJ, Fusaro VA, Steinberg SM, Mills GB, Simone C, Fishman DA, Kohn EC, Liotta LA: **Use of proteomic patterns in serum to identify ovarian cancer**. *Lancet* 2002, **359:**572-577.

Thank you for your attention!
Questions/Comments?

Email: Alexander.Statnikov@med.nyu.edu

URL: http://ww.nyuinformatics.org