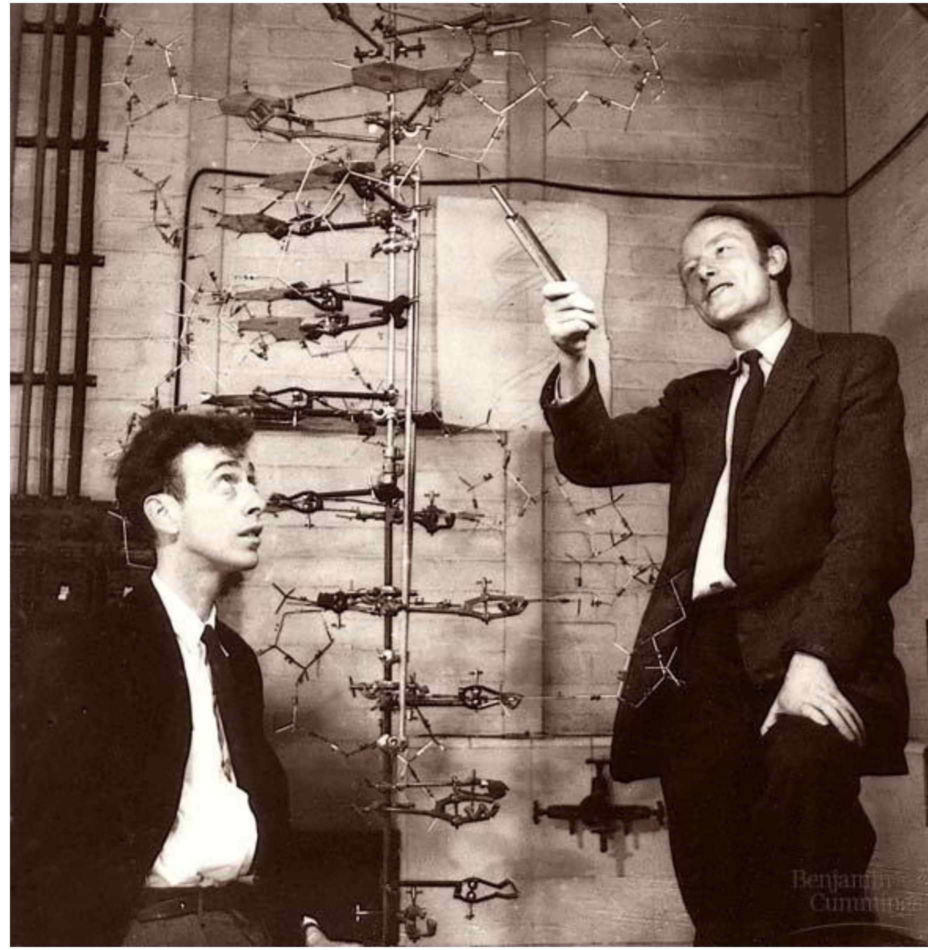
Machine Learning
&
SVM

Shannon

"Information is any difference that makes a difference." **Bateman**



" It has not escaped our notice that the specific pairing we have postulated immediately suggests a possible copying mechanism for the genetic material. " **Watson&Crick**

Convergence:
Machine learning → Brain ← Learning machine

- What is machine learning and why it has become increasingly important in biology
- Recent advances on machine learning theory and its applications in biology
- Biological systems are learning machines
- Towards the building of ultimate learning machine: brain (artificial?)

Feynman Lectures on computation

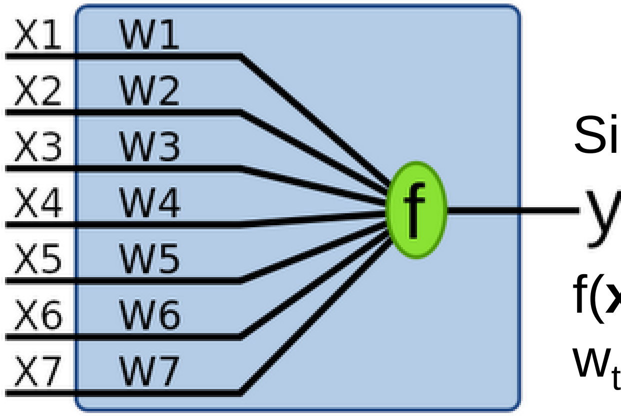
Learning through statistical inference

- Input **X**, output Y, learning means to estimate conditional probability P(Y|**X**) from training samples $(y_1,\mathbf{x_1})$, $(y_2,\mathbf{x_2})$,…,$(y_n,\mathbf{x_n})$, *dim*(**x**)=*p*. e.g. Y=phenotype, **X**=genotype.
- Statistical Models: $Y = f(\mathbf{X},\boldsymbol{\alpha}) + \varepsilon$ with $E(\varepsilon)=0$, **X** and ε independent. i.e. $E(Y|\mathbf{X}=\mathbf{x}) = f(\mathbf{x})$, P(Y|**X**) depends on **X** only through f(**X**).

- MLE (R. Fisher)
- Learning = estimate f(•), when Y={0,1} (or k-class labels), **Classification**; when Y=continuous, **Regression**. Many statistical models

Perceptrons

Alan Turing: Universal computer-Turing machine



Simplest ANN (**Frank Rosenblatt**, 1957)

f(**x**) = 1 if **w•x+b** > 0; 0 else.

$w_{t+1}(j) = w_t(j) + \alpha(y - f(x))x(j)$ (j=1,…,n)

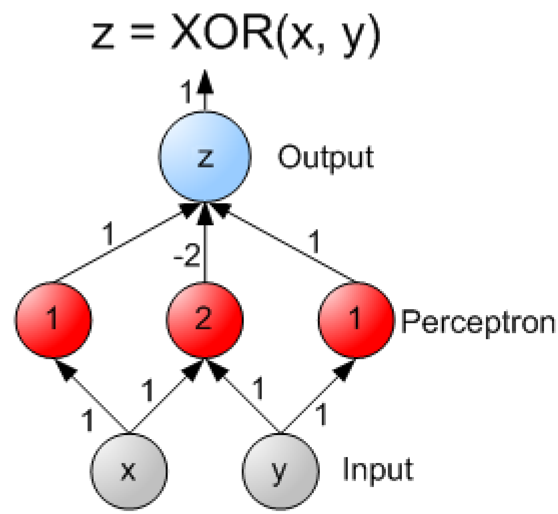
It can learn many functions: e.g. AND, OR

But Marvin Minsky & Seymour Papert showed it cannot learn XOR (1969)

A three layer Perceptron net capable of calculating XOR.

Multi-layer NN can learn continuous functions, using e.g. sigmoid function (equivalent to Logistic regression)

Problems with ANNs: choose structure, Local minima, computational intensive, black box, ..



Data explosion:
Web, biology,etc.

Statistician's catchup

One view says that our field should concentrate on that small part of information science that we do best, namely probabilistic inference based on mathematics. If this view is adopted, we should become resigned to the fact that the role of Statistics as a player in the "information revolution" will steadily diminish over time.

Another point of view holds that statistics ought to be concerned with **data analysis**. *The field should be defined in terms of a set of problems — rather than a set of tools — that pertain to data*. Should this point of view ever become the dominant one, a big change would be required in our practice and academic programs.

First and foremost, we would have to *make peace with computing*. It's here to stay; that's where the data is. This has been one of the most glaring omissions in the set of tools that have so far defined Statistics. Had we incorporated computing methodology from its inception as a fundamental statistical tool (as opposed to simply a convenient way to apply our existing tools) *many of the other data related fields would not have needed to exist*. They would have been part of our field.

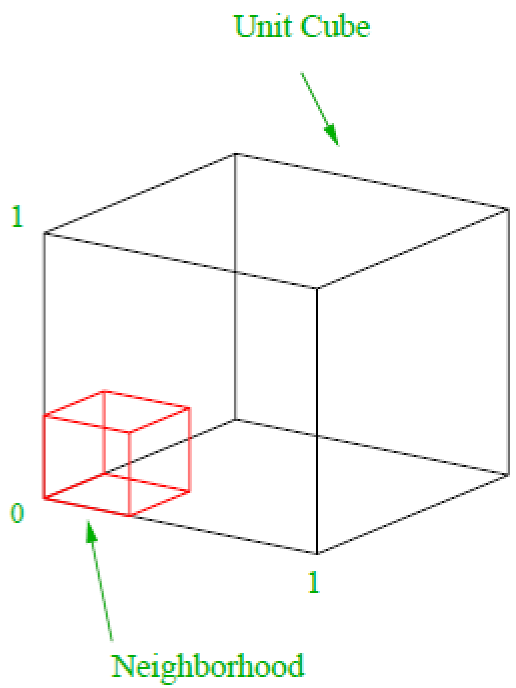
Friedman (lightly edited by O'Connor)

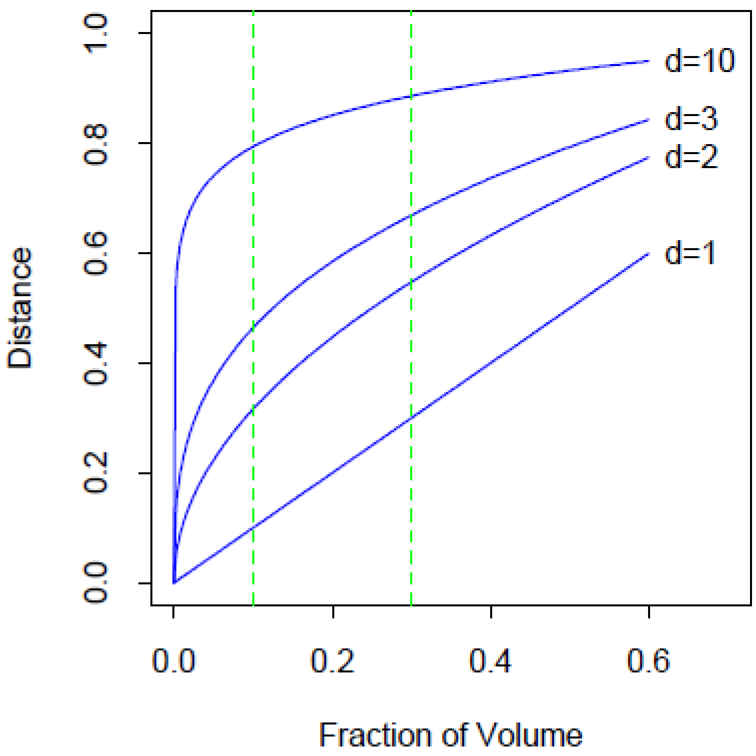
ML vs. Stat

Glossary	
Machine learning(Data mining)	Statistics (Statistical learning)
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = $1,000,000	large grant = $50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

Robert Tibshiriani

Curse of dimensionality (Bellman, 1961)





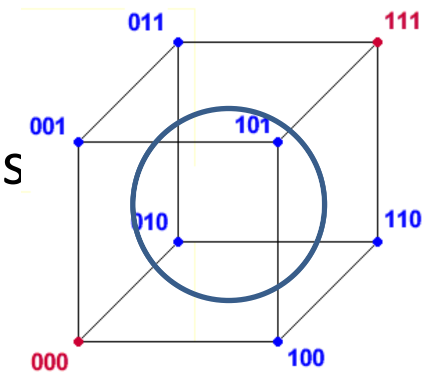
The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. *In 10 dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data*. *No neighborhood is "small"!*

Concentration of measure in n-dim

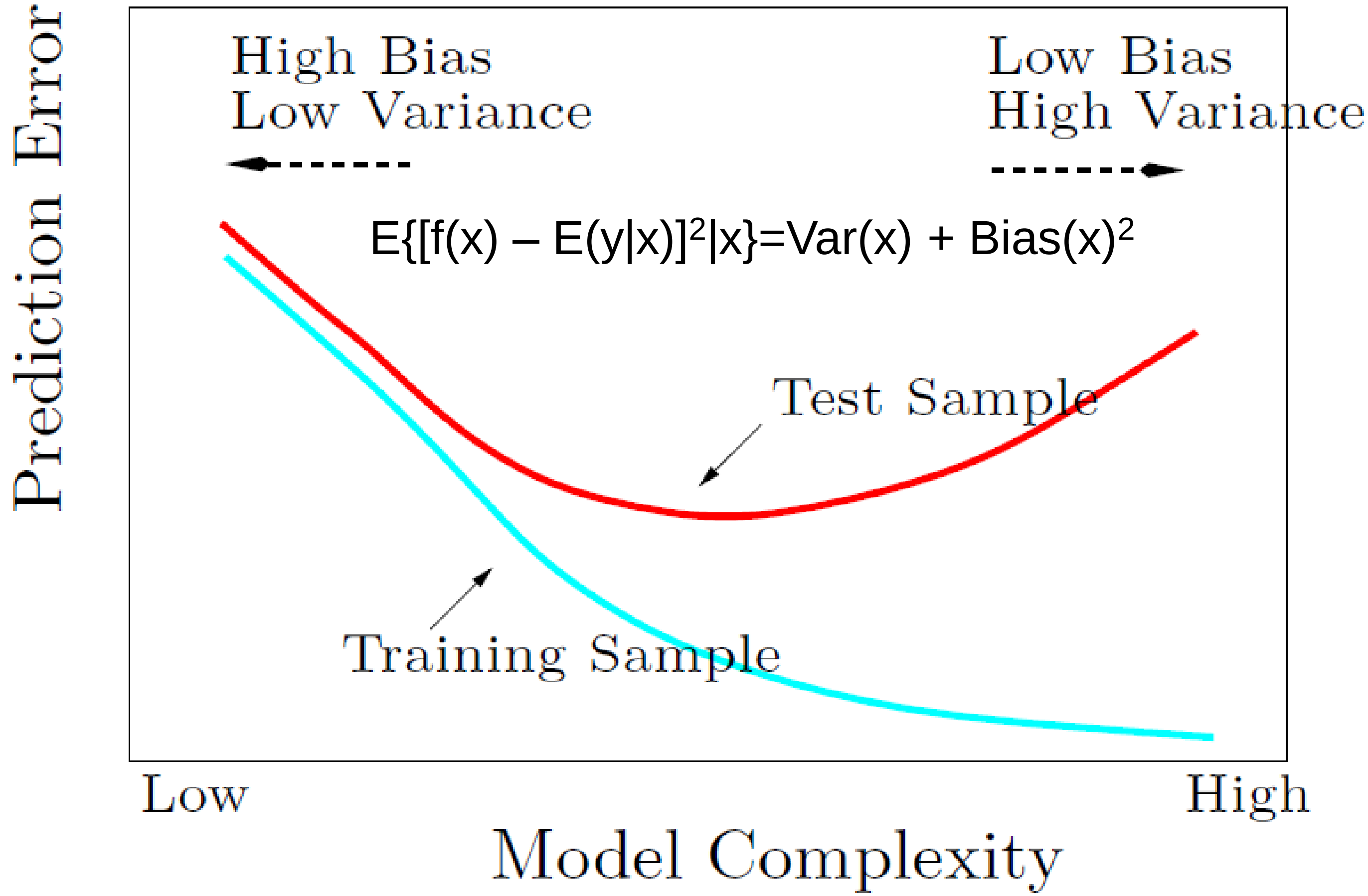
(V. Milman, M. Gromov, M. Talagrand early 1970s)

As n $\rightarrow \infty$:

- There is no such thing as "a thin shell of small volume" in hamming-ball (hypercube of volume 1), such shell contains almost all the volume.
- A cube is not contained in any ball of finite radius and the unit ball (as all balls of finite radius) has volume 0!
- The volume of a ball of radius R•Sqrt(n/2πe) = 0, if R ≤ 1; ∞ else.
- Surface area concentration near the equator. Similarly Hamming ball volume also concentrates near the hypersurface containing the diagonal. → *geometric meaning of "The weak law of large numbers"*



Bias-variance tradeoff (Like drivng a car)



"I remember my friend **Johnny von Neumann** used to say, *'with four parameters I can fit an elephant* and with five I can make him wiggle his trunk.'[Enrico Fermi](),

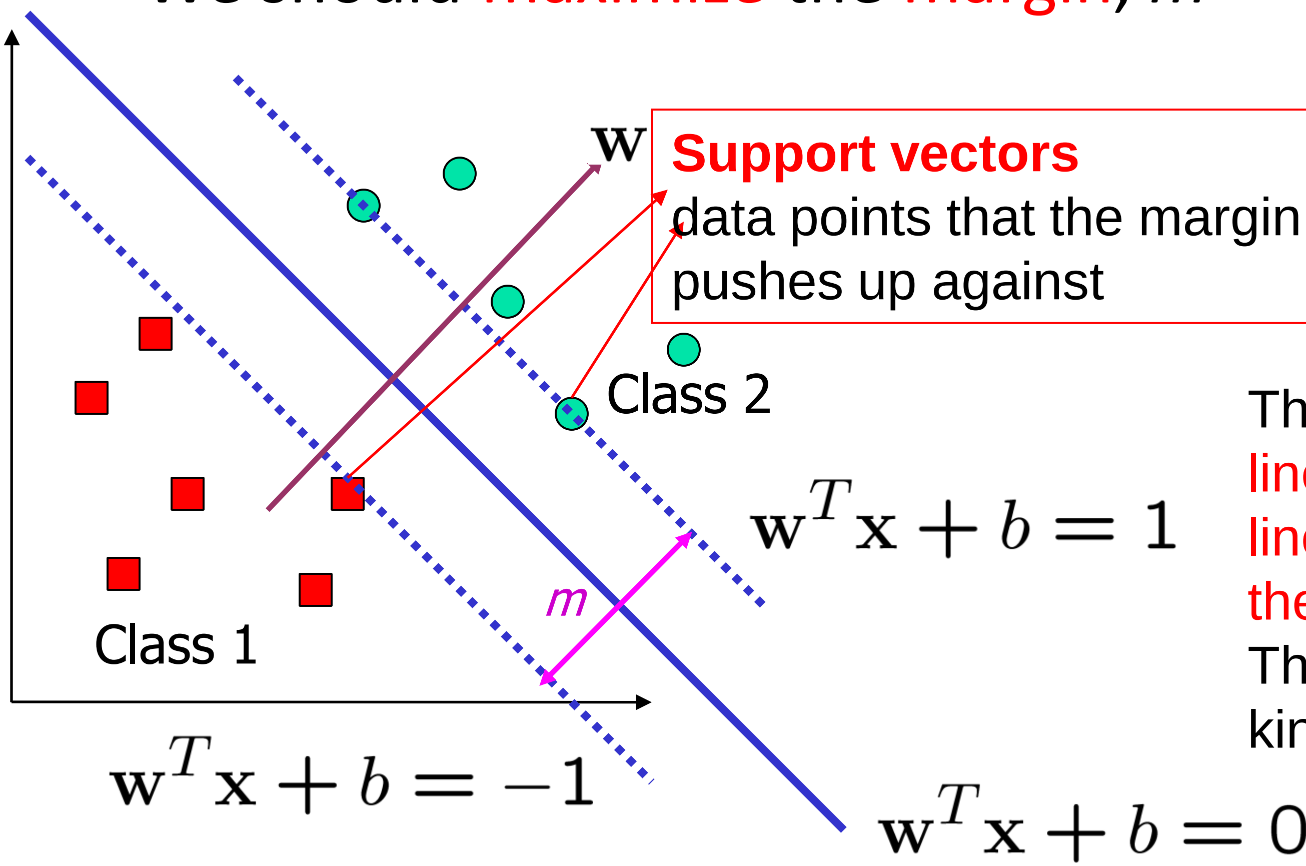
Cellular Atomata: von Neumann machine

Support Vector Machines (*SVM*)

Vapnik (1995)

The decision boundary should be as far away from the data of both classes as possible

–We should maximize the margin, *m*



$$m = \frac{2}{||\mathbf{w}||}$$

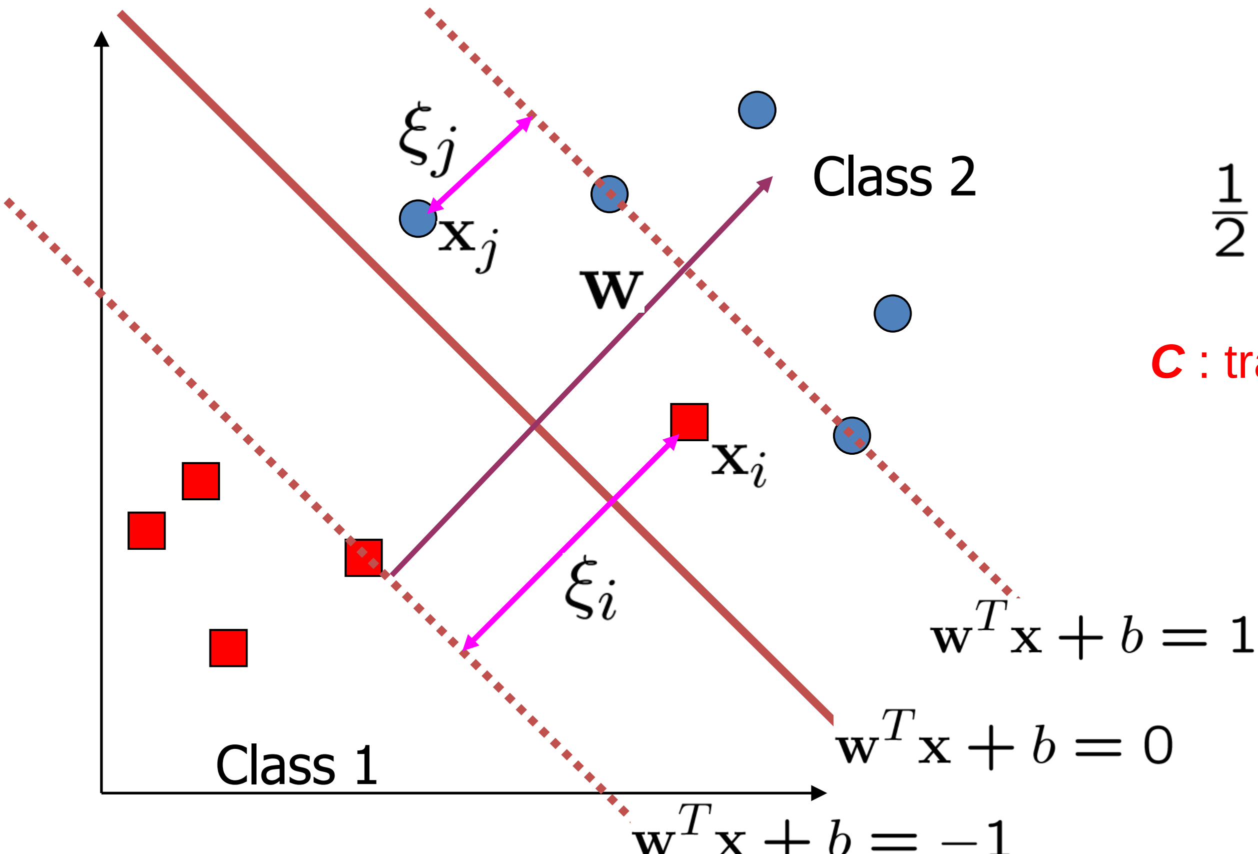
$$\|\mathbf{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}.$$

The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of *LSVM*

Non-separable case: "soft" margin

We allow "error" ξ_i in classification; it is based on the output of the discriminant function $\mathbf{w}^T\mathbf{x}$+b

ξ_i approximates the number of misclassified samples



New objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

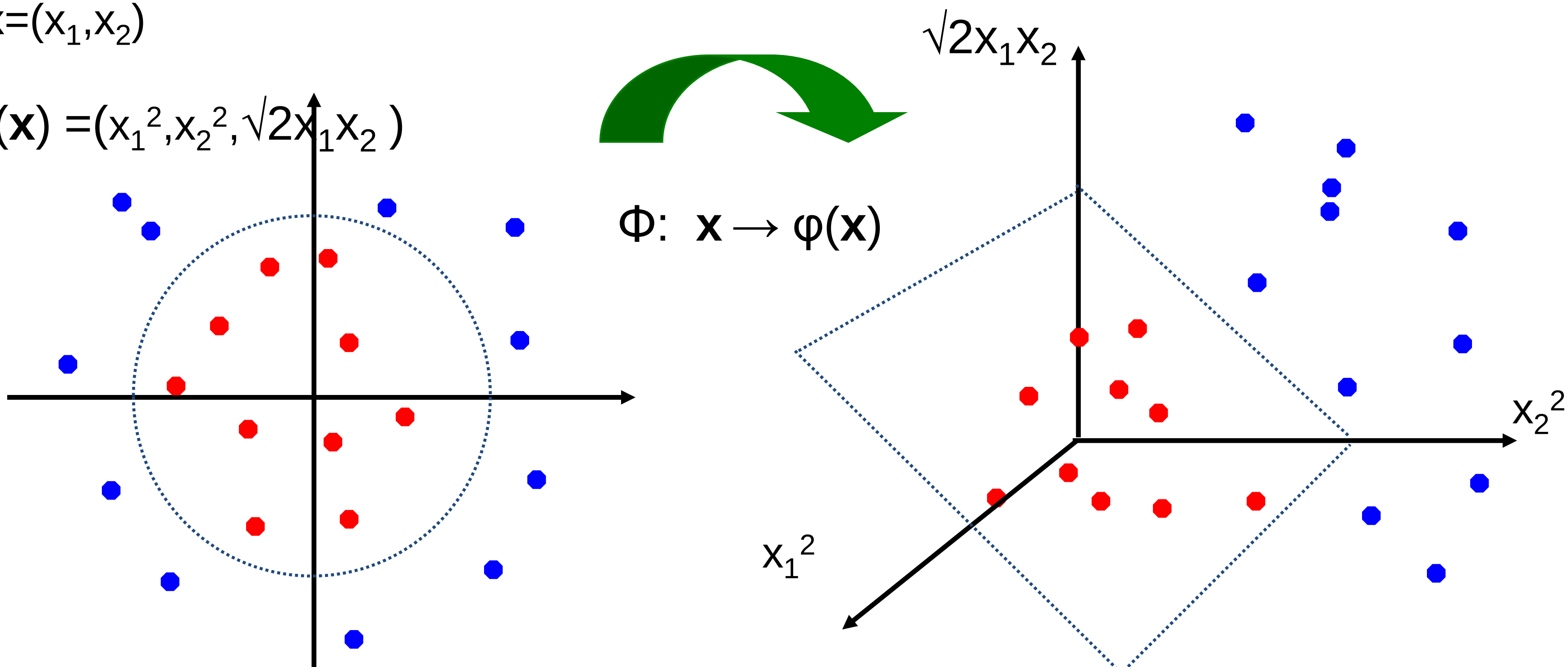
C : tradeoff parameter between error and margin; chosen by the user; large C means a higher penalty to errors

Non-linear SVM: Feature Space

General idea: the original input space (x) can be mapped to some higher-dimensional feature space ($\varphi(\mathbf{x})$)where the training set is separable:

$x=(x_1,x_2)$

$\varphi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2\ )$



If data are mapped into higher a space of sufficiently high dimension, then they will in general be linearly separable;

N data points are in general separable in a space of N-1 dimensions or more!!!

The *kernel* trick

Change all inner products to kernel functions

For training,

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Convex geometry, Duality, Langrangian multiplier α (virtual force)

Original

$$\text{max.} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

With kernel function

$$\text{max.} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{K(\mathbf{x}_i, \mathbf{x}_j)}$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

QP solver can be used to find α_i efficiently!!!

Examples of Kernel Functions

- Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^T\mathbf{x_j}$
- Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j}) = (1 + \mathbf{x_i}^T\mathbf{x_j})^p$
- Gaussian (radial-basis function network):

$$K(\mathbf{x_i},\mathbf{x_j}) = \exp\left(-\frac{\left\|\mathbf{x_i} - \mathbf{x_j}\right\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0 \mathbf{x_i}^T\mathbf{x_j} + \beta_1)$
- String Kernels


Predict mCpG

General Steps

- Prepare data matrix $\{(x_i,y_i)\}$
- Select a Kernel function
- Select the error parameter C
- "Train" the system (to find all α_i)
- New data can be classified using α_i and Support Vectors

shRNA target prediction;GWAS