CSC2545 – Topics in Machine Learning:
Kernel Methods and Support Vector Machines

- A comprehensive introduction to SVMs and other kernel methods, including theory, algorithms and applications.
- Instructor: Anthony Bonner
- Prerequisites: linear algebra, vector calculus, basic probability, mathematical maturity, programming experience.
- Expected work: 3-4 assignments.
- Lectures: TF 2-3:30pm (sometimes 2-3pm)
- For more information see www.cs.toronto.edu/~bonner

Course Content

- Some things kernel methods and SVMs can do:
  - non-linear classification
  - non-linear regression
  - non-linear dimensionality reduction
- How and why they work:
  - Hilbert space
  - statistical learning theory
  - regularization
- Possible additional topics:
  - convex optimization
  - implementation issues
  - Bayesian kernel methods

Data

Vectors

- Collections of features
  e.g. height, weight, blood pressure, age, ...
- Can map categorical variables into vectors

Matrices

- Images, Movies
- Remote sensing and satellite data (multispectral)

Strings

- Documents
- Gene sequences

Structured Objects

- XML documents
- Graphs

Optical Character Recognition

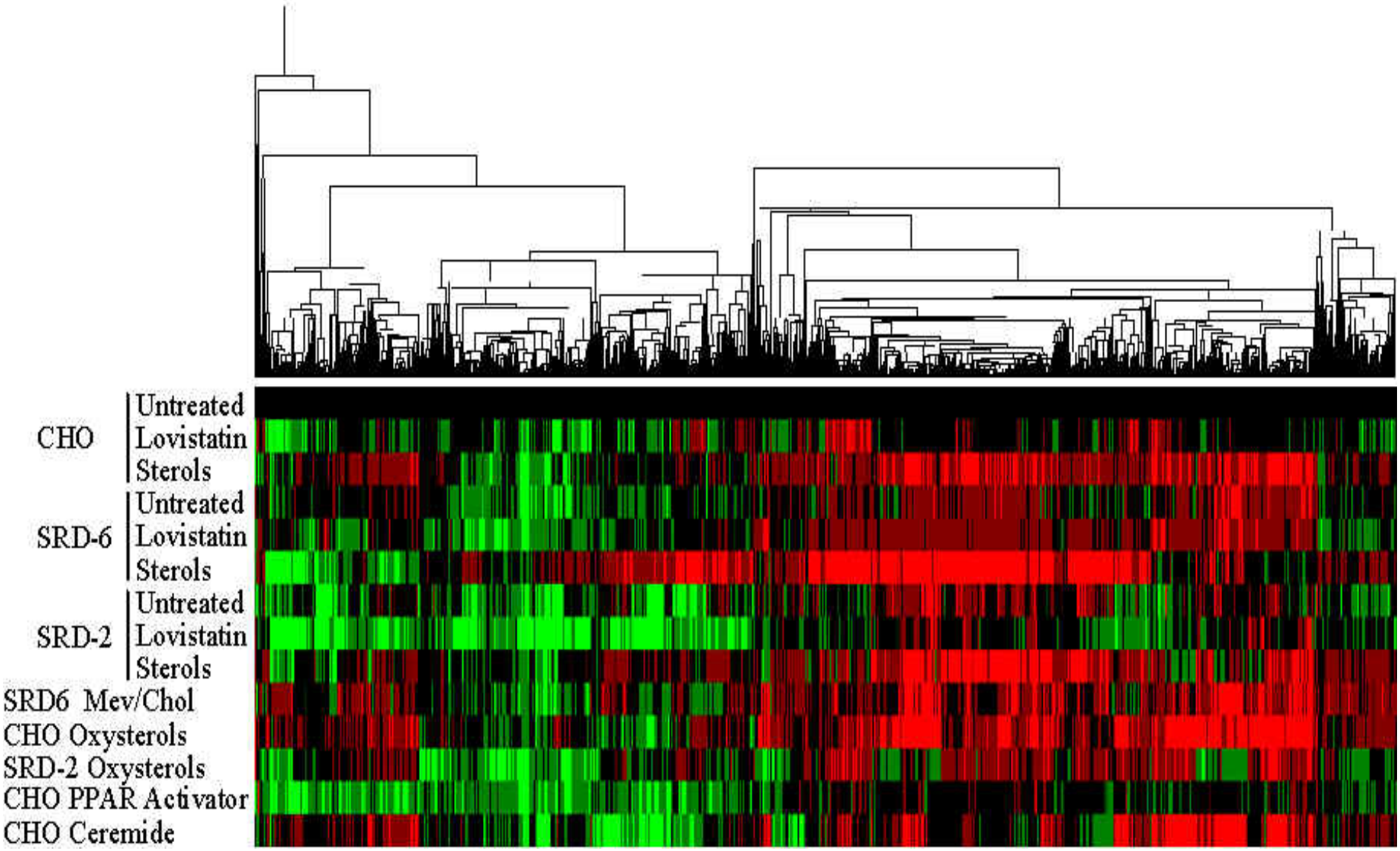
Reuters Database

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="13522" NEWID="8001">
<DATE>20-MAR-1987 16:54:10.55</DATE>
<TOPICS><D>earn</D></TOPICS>
<PLACES><D>usa</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
&#5;&#5;&#5;F
&#22;&#22;&#1;f2479&#31;reute
r f BC-GANTOS-INC-&lt;GTOS>-4TH   03-20 0056</UNKNOWN>
<TEXT>&#2;
<TITLE>GANTOS INC &lt;GTOS> 4TH QTR JAN 31 NET</TITLE>
<DATELINE>    GRAND RAPIDS, MICH., March 20 -
    </DATELINE><BODY>Shr 43 cts vs 37 cts
    Net 2,276,000 vs 1,674,000
    Revs 32.6 mln vs 24.4 mln
    Year
    Shr 90 cts vs 69 cts
    Net 4,508,000 vs 3,096,000
    Revs 101.0 mln vs 76.9 mln
    Avg shrs 5,029,000 vs 4,464,000
    NOTE: 1986 fiscal year ended Feb 1, 1986
 Reuter
&#3;</BODY></TEXT>
</REUTERS>
```

Faces

More Faces

Microarray Data



Biological Sequences

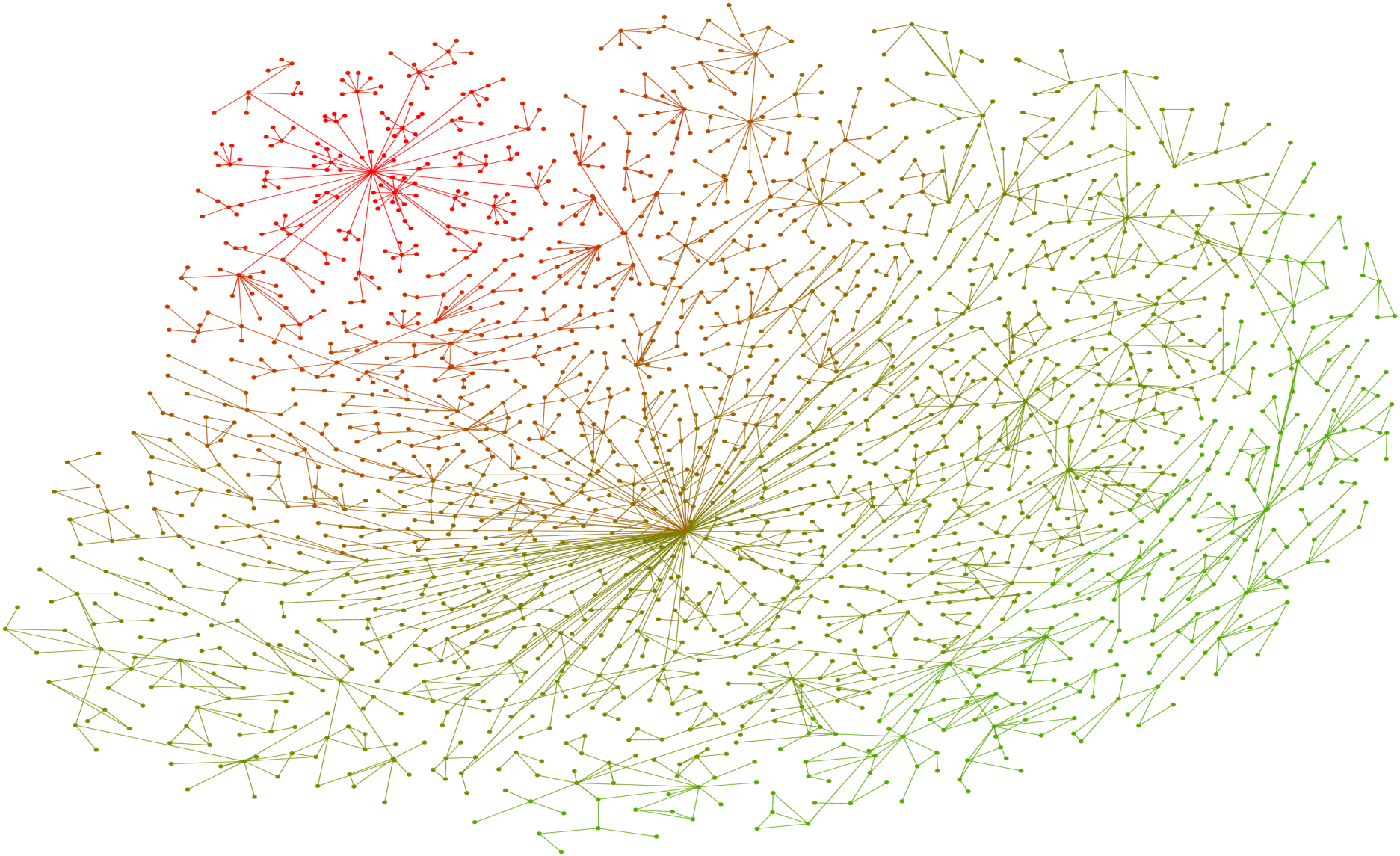
Goal

Estimate function of protein based on sequence information.

Example Data

>0_d1vcaa2 2.1.1.4.1 (1-90) N-terminal domain of vascular cell adhesion molecule-1 (VCAM-1) [human (Homo sapiens)]
FKIETTPESRYLAQIGDSVSLTCSTTGCESPFFSWRTQIDSPLNGKVTNEGTTSTLTMNPVSFGNEHSYL
CTATCESRKLEKGIQVEIYS
>0_d1zxq_2 2.1.1.4.2 (1-86) N-terminal domain of intracellular adhesion molecule-2, ICAM-2 [human (Homo sapiens)]
KVFEVHVRPKKLAVEPKGSLEVNCSTTCNQPEVGGLETSLNKILLDEQAQWKHYLVSNISHDTVLQCHFT
CSGKQESMNSNVSVYQ
>0_d1tlk__ 2.1.1.4.3 Telokin [turkey (Meleagris gallopavo)]
VAEEKPHVKPYFTKTILDMDVVEGSAARFDCKVEGYPDPEVMWFKDDNPVKESRHFQIDYDEEGNCSLTI
SEVCGDDDAKYTCKAVNSLGEATCTAELLVETM
>0_d2ncm__ 2.1.1.4.4 N-terminal domain of neural cell adhesion molecule (NCAM) [human (Homo sapiens)]
RVLQVDIVPSQGEISVGESKFFLCQVAGDAKDKDISWFSPNGEKLSPNQQRISVVWNDDDSSTLTIYNAN
IDDAGIYKCVVTAEDGTQSEATVNVKIFQ
>0_d1tnm__ 2.1.1.4.5 Titin [Human (Homo sapiens), module M5]
RILTKPRSMTVYEGESARFSCDTDGEPVPTVTWLRKGQVLSTSARHQVTTTKYKSTFEISSVQASDEGNY
SVVVENSEGKQEAEFTLTIQK
>0_d1wiu__ 2.1.1.4.6 Twitchin [Nematode (Caenorhabditis elegans)]
LKPKILTASRKIKIKAGFTHNLEVDFIGAPDPTATWTVGDSGAALAPELLVDAKSSTTSIFFPSAKRADS
GNYKLKVKNELGEDEAIFEVIVQ
>0_d1koa_1 2.1.1.4.6 (351-447) Twitchin [Nematode (Caenorhabditis elegans)]
QPRFIVKPYGTEVGEGQSANFYCRVIASSPPVVTWHKDDRELKQSVKYMKRYNGNDYGLTINRVKGDDKG
EYTVRAKNSYGTKEEIVFLNVTRHSEP

Graphs



What to do with data

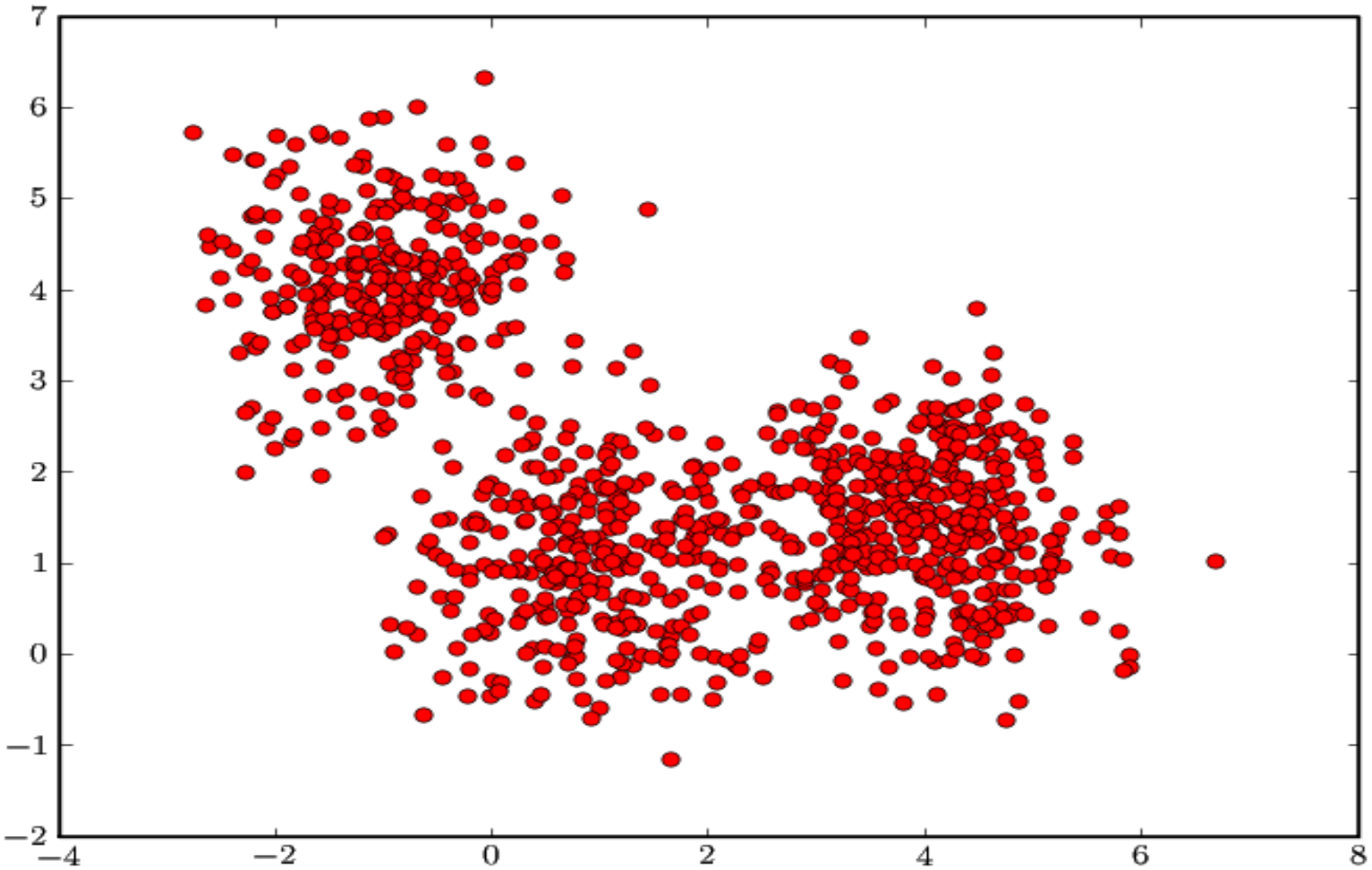
Unsupervised Learning

- Find clusters of the data
- Find low-dimensional representation of the data (e.g. unroll a swiss roll, find structure)
- Find interesting directions in data
- Interesting coordinates and correlations
- Find novel observations / database cleaning

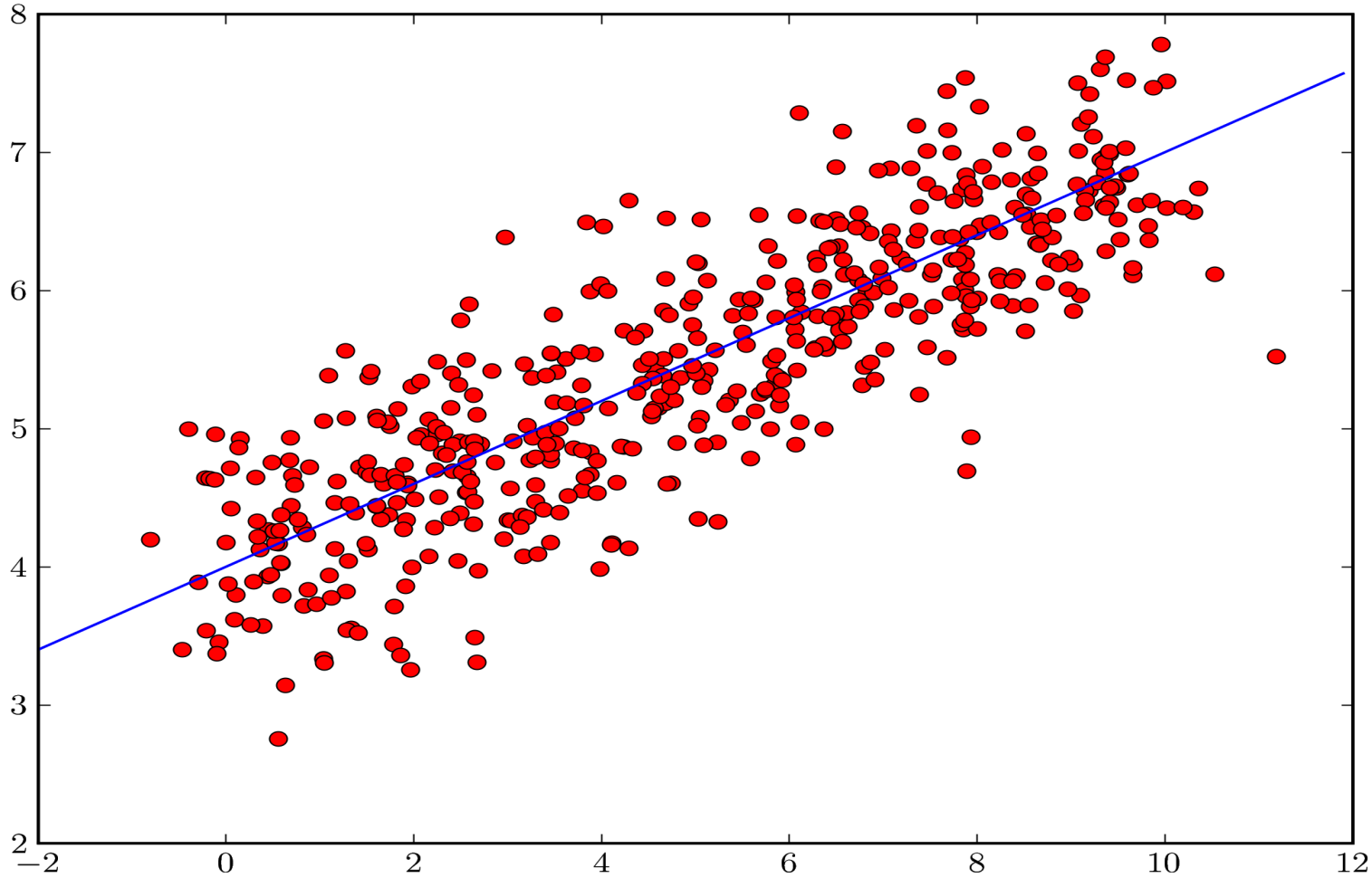
Supervised Learning

- Classification (distinguish apples from oranges)
- Speech recognition
- Regression (tomorrow's stock value)
- Predict time series
- Annotate strings

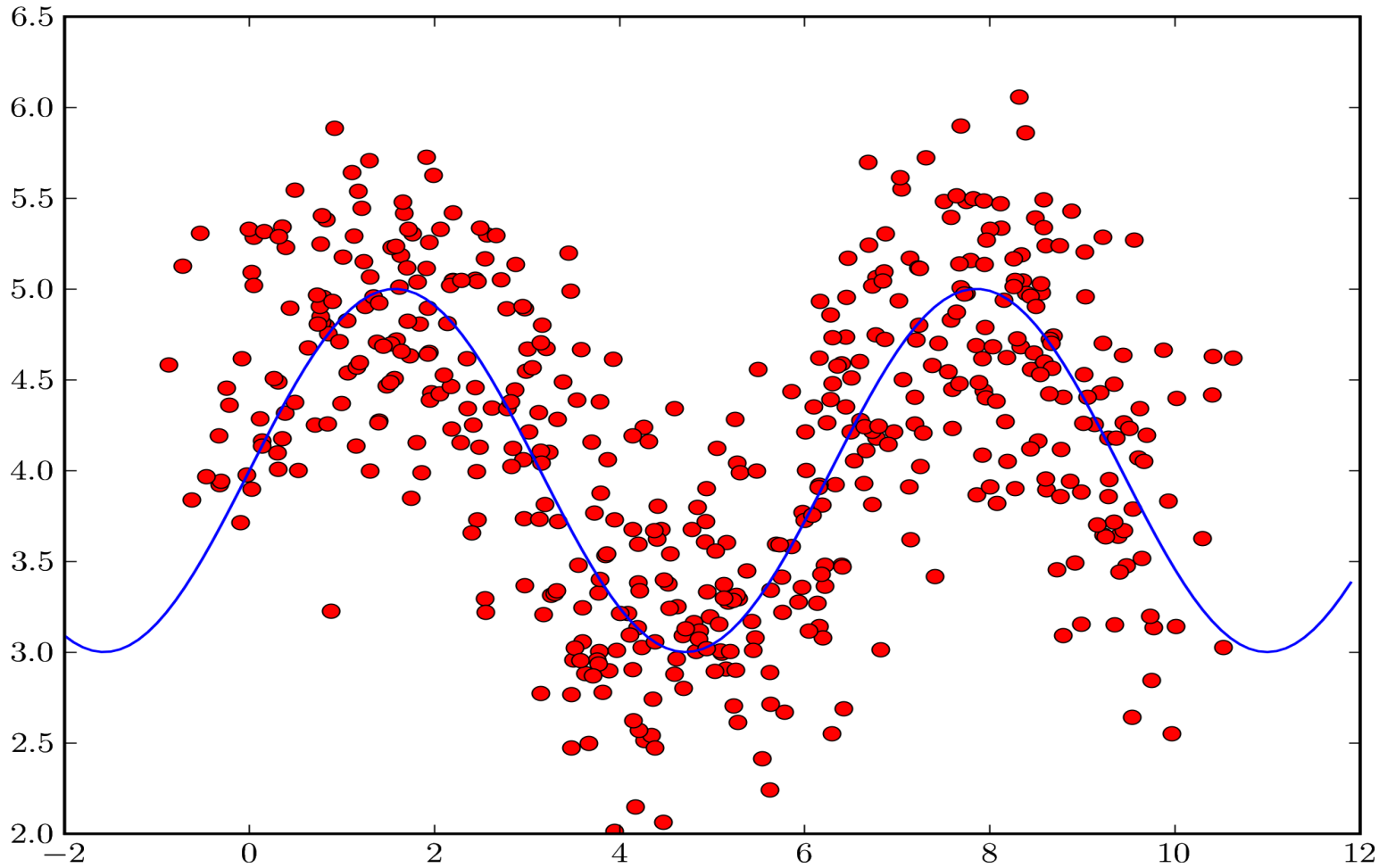
Clustering



Principal Components



Linear Subspace



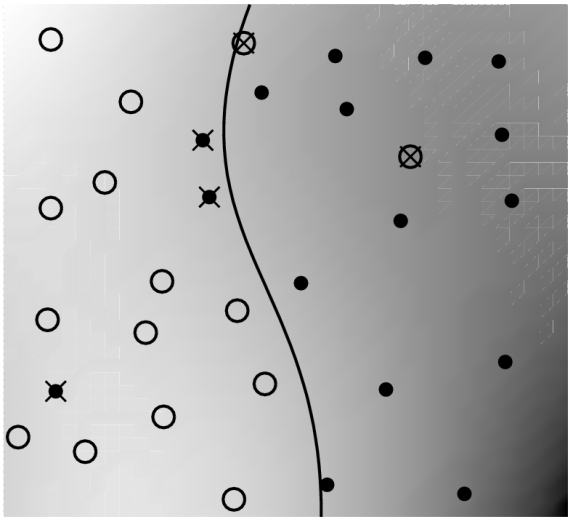
Classification

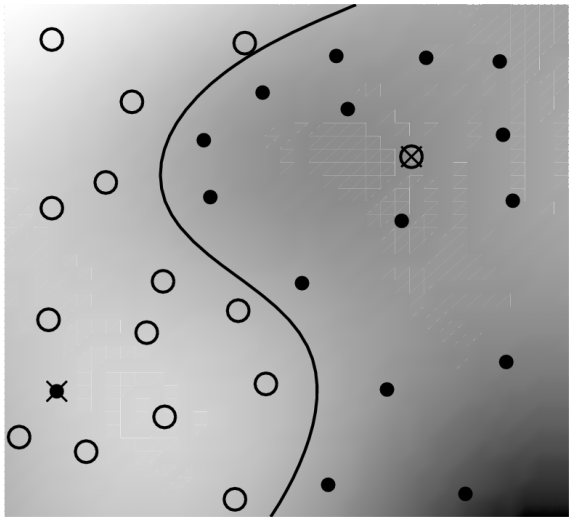
Data

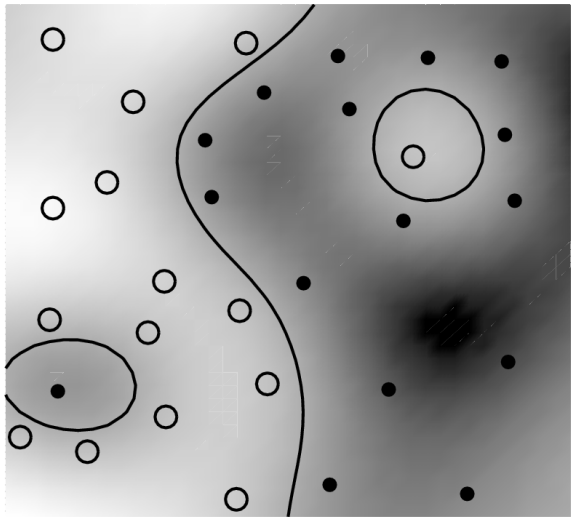
Pairs of observations (x_i, y_i) drawn from distribution
e.g., (blood status, cancer), (credit transactions, fraud), (sound profile of jet engine, defect)

Goal

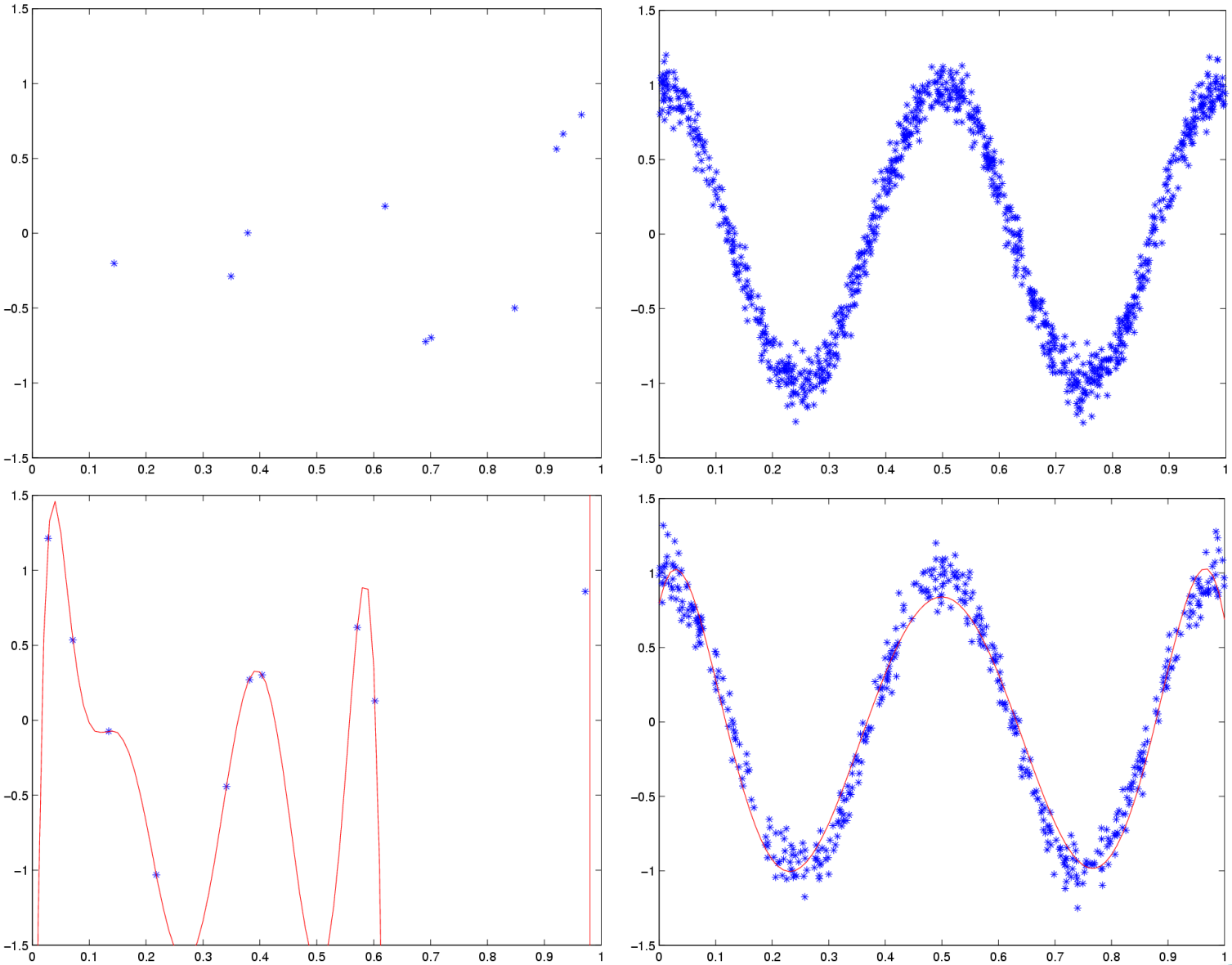
Estimate $y \in \{\pm 1\}$ **given** x at a new location. Or find a function $f(x)$ that does the trick.







Regression



Regression

Data

Pairs of observations (x_i, y_i) generated from some joint distribution $\Pr(x, y)$, e.g.,

- market index, SP100
- fab parfameters, yield
- user profile, price

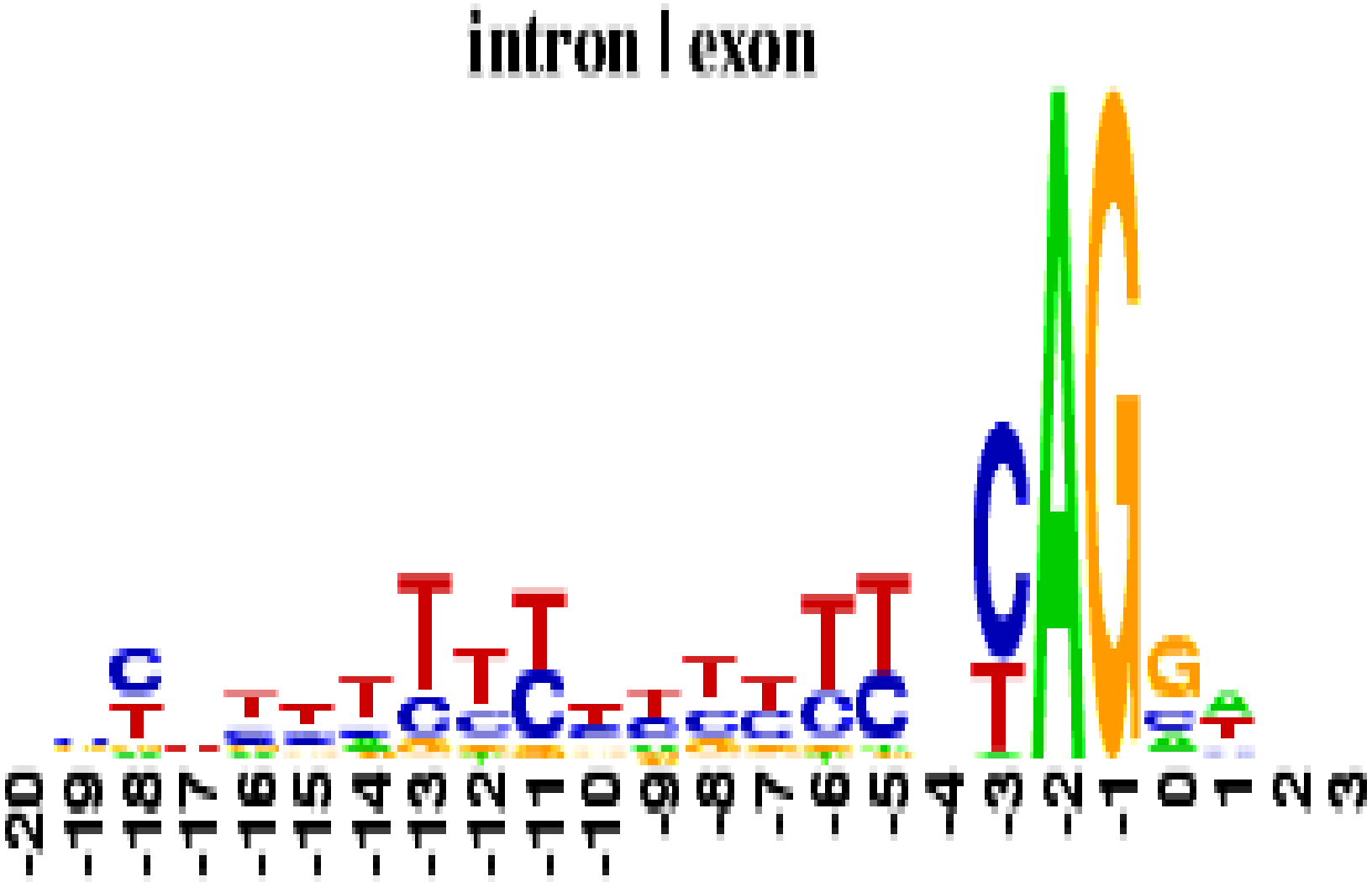
Task

Estimate y, given x, such that some loss $c(x, y, f(x))$ is minimized.

Examples

- Quadratic error between y and $f(x)$, i.e. $c(x, y, f(x)) = \frac{1}{2}(y - f(x))^2$.
- Absolute value, i.e., $c(x, y, f(x)) = |y - f(x))|$.

Annotating Strings



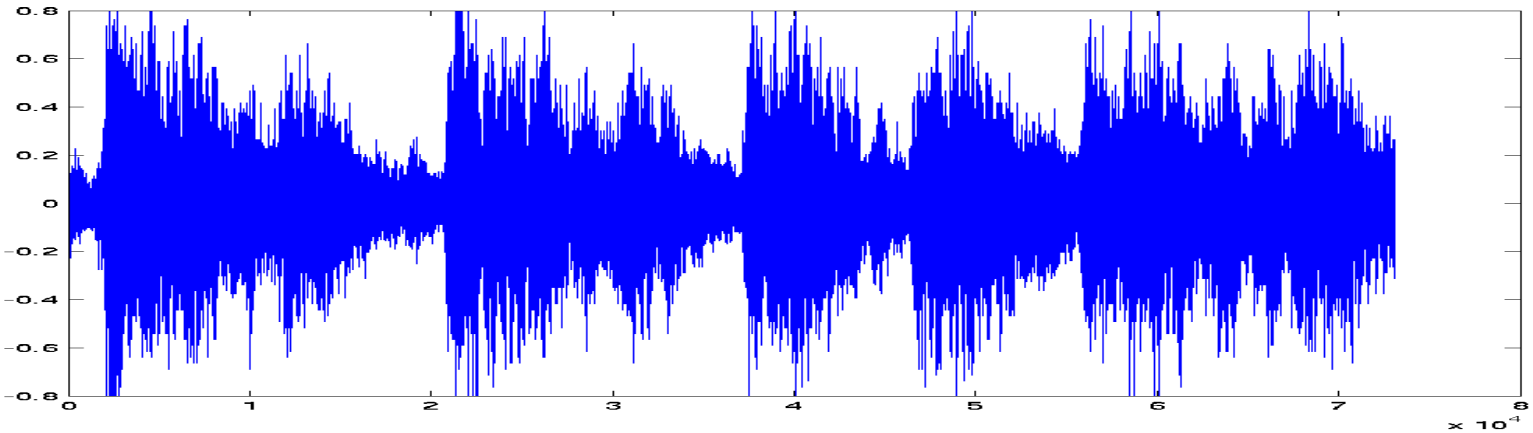
Annotating Audio

Goal

- Possible meaning of an audio sequence
- Give confidence measure

Example (from Australian Prime Minister's speech)

- a stray alien
- Australian



Novelty Detection

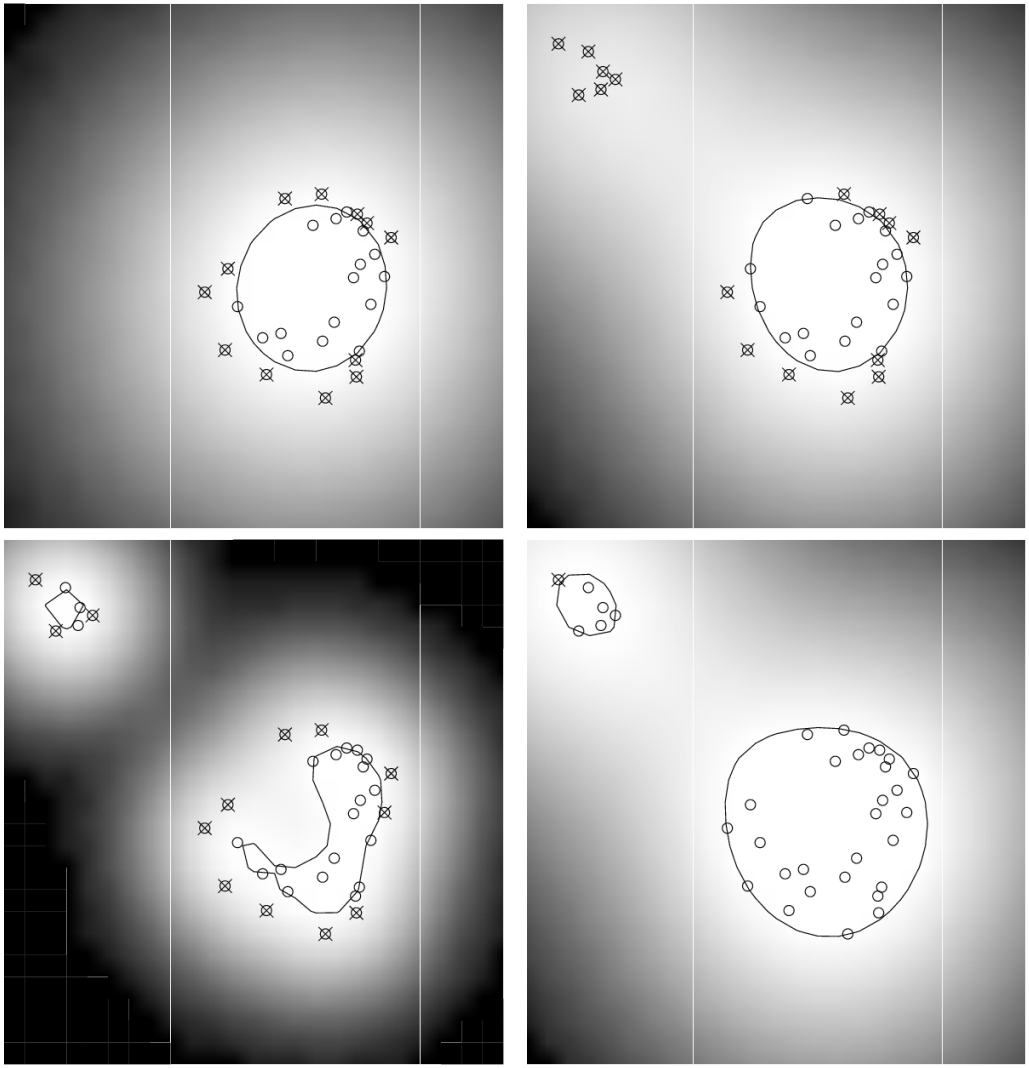
Data

Observations x_i from some $\mathrm{P}(x)$, e.g.,

- network usage patterns
- handwritten digits
- alarm sensors
- factory status

Task

Find unusual events, clean database, distinguish typical examples.



Learning and Similarity: some Informal Thoughts

- input/output sets $\mathcal{X}, \mathcal{Y}$
- training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathcal{Y}$
- "generalization": given a previously unseen $x \in \mathcal{X}$, find a suitable $y \in \mathcal{Y}$
- (x, y) should be "similar" to $(x_1, y_1), \ldots, (x_m, y_m)$
- how to measure similarity?
  - for outputs: *loss function* (e.g., for $\mathcal{Y} = \{\pm 1\}$, zero-one loss)
  - for inputs: *kernel*

Similarity of Inputs

- symmetric function

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$(x, x') \mapsto k(x, x')$$

- for example, if $\mathcal{X} = \mathbb{R}^N$: canonical dot product

$$k(x, x') = \sum_{i=1}^{N} [x]_i [x']_i$$

- if $\mathcal{X}$ is not a dot product space: assume that k has a representation as a dot product in a linear space $\mathcal{H}$, i.e., there exists a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that

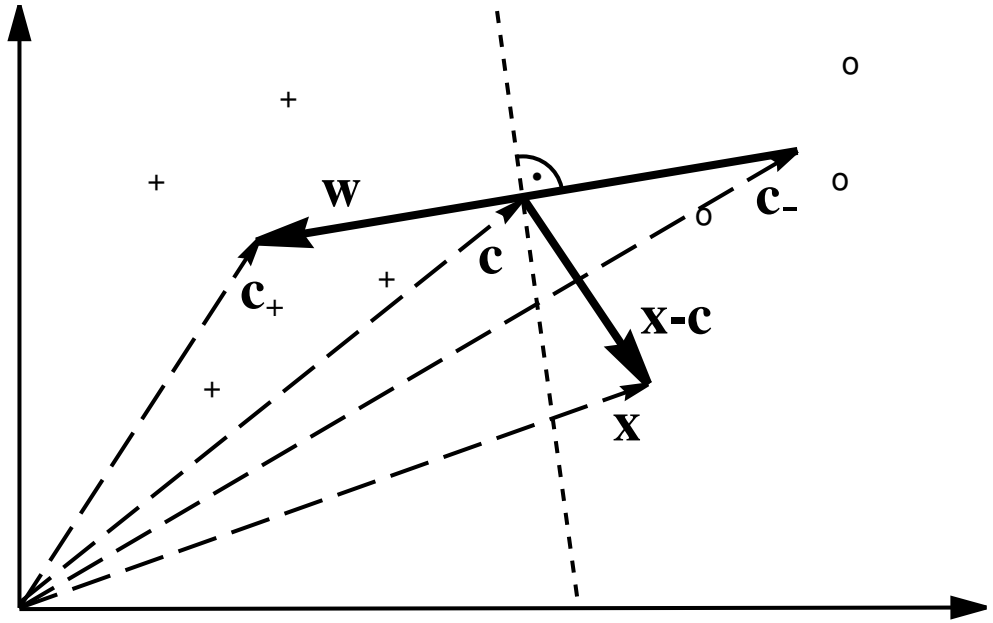
$$k(x, x') = \left\langle \Phi(x), \Phi(x') \right\rangle .$$

- in that case, we can think of the patterns as $\Phi(x), \Phi(x')$, and carry out geometric algorithms in the dot product space ("feature space") $\mathcal{H}$.

An Example of a Kernel Algorithm

Idea: classify points $\mathbf{x} := \Phi(x)$ in feature space according to which of the two class means is closer.

$$\mathbf{c}_+ := \frac{1}{m_+} \sum_{y_i=1} \Phi(x_i), \quad \mathbf{c}_- := \frac{1}{m_-} \sum_{y_i=-1} \Phi(x_i)$$



Compute the sign of the dot product between $\mathbf{w} := \mathbf{c}_+ - \mathbf{c}_-$ and $\mathbf{x} - \mathbf{c}$.

An Example of a Kernel Algorithm, ctd. [25]

$$f(x) = \operatorname{sgn}\left(\frac{1}{m_+}\sum_{\{i:y_i=+1\}}\langle\Phi(x),\Phi(x_i)\rangle - \frac{1}{m_-}\sum_{\{i:y_i=-1\}}\langle\Phi(x),\Phi(x_i)\rangle + b\right)$$

$$= \operatorname{sgn}\left(\frac{1}{m_+}\sum_{\{i:y_i=+1\}}k(x,x_i) - \frac{1}{m_-}\sum_{\{i:y_i=-1\}}k(x,x_i) + b\right)$$

where

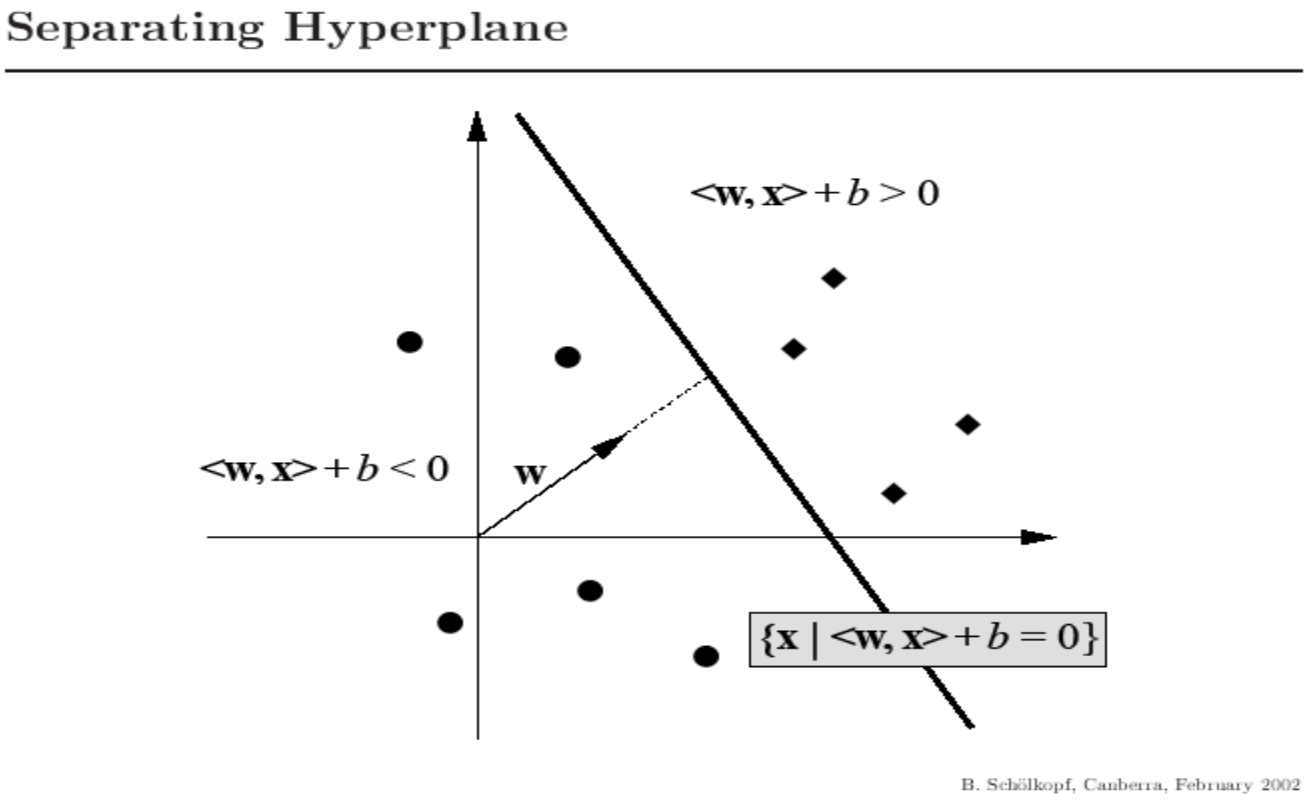
$$b = \frac{1}{2}\left(\frac{1}{m_-^2}\sum_{\{(i,j):y_i=y_j=-1\}}k(x_i,x_j) - \frac{1}{m_+^2}\sum_{\{(i,j):y_i=y_j=+1\}}k(x_i,x_j)\right).$$

- provides a geometric interpretation of Parzen windows

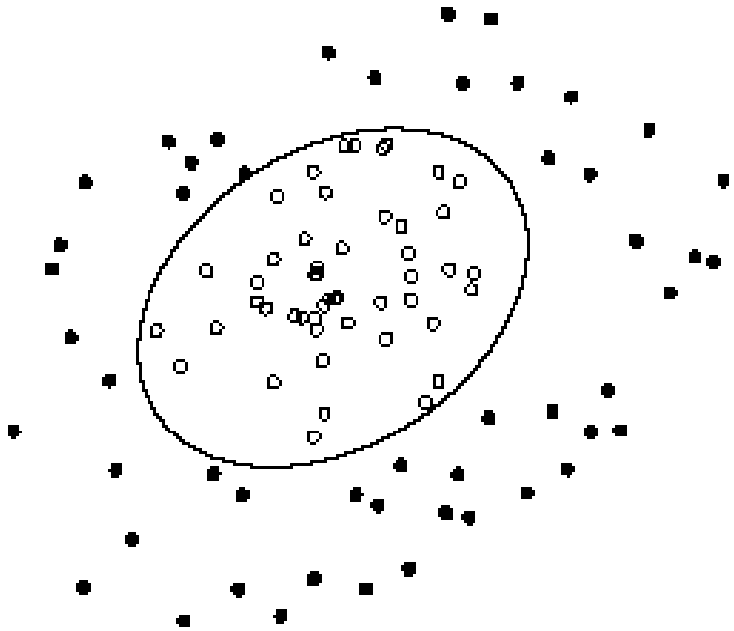
Non-linear Classification via Support Vector Machines

- Based on several ideas:
  - linear classification
  - *increasing* the dimensionality
  - the kernel trick
  - maximizing the margin
  - duality

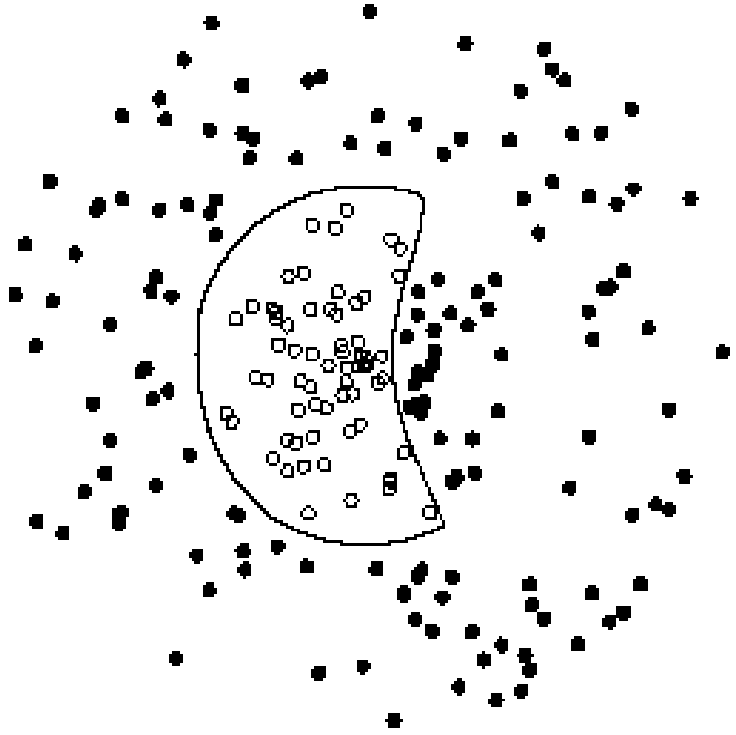
Linear Classification



Non-linear Classification



separation by ellipsoid



separation by 4th degree polynomial

Kernel trick

kernel trick: dot-products in feature space can be computed as a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$

idea: work directly on $\mathbf{x}$, avoid having to compute $\Phi(\mathbf{x})$ at all

example:

$$\begin{aligned} K(\mathbf{a}, \mathbf{b}) &= (\mathbf{a} \cdot \mathbf{b})^3 = ((a_1, a_2) \cdot (b_1, b_2))^3 \\ &= (a_1 b_1 + a_2 b_2)^3 \\ &= a_1^3 b_1^3 + 3 a_1^2 b_1^2 a_2 b_2 + 3 a_1 b_1 a_2^2 b_2^2 + a_2^3 b_2^3 \\ &= ((a_1^3, \sqrt{3} a_1^2 a_2, \sqrt{3} a_1 a_2^2, a_2^3) \cdot (b_1^3, \sqrt{3} b_1^2 b_2, \sqrt{3} b_1 b_2^2, b_2^3)) \\ &= \Phi(\mathbf{a}) \cdot \Phi(\mathbf{b}) \end{aligned}$$

Kernels

examples:

1. polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^z$
2. Gaussian kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2)$
3. sigmoid kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa(\mathbf{x}_i \cdot \mathbf{x}_j) + a)$

each kernel computation corresponds to dot product calculation for particular mapping $\Phi(\mathbf{x})$ – implicitly maps to high-dim space

why useful?

- rewrite training examples using more complex features
- dataset not linearly separable in original space may be linearly separable in higher-dimensional space