Plan of Class 4

- Radial Basis Functions with moving centers;
- Multilayer Perceptrons;
- Projection Pursuit Regression and ridge functions approximation;
- Principal Component Analysis: basic ideas;
- Radial Basis Functions and dimensionality reduction;
- From Additive Splines to ridge functions approximation;

Regularization approach

A smooth function that approximates the data set $D = \{(\mathbf{x}_i, y_i) \in R^d \times R\}_{i=1}^N$ can be found minimizing the functional:

$$H[f] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 + \lambda \int_{R^d} d\mathbf{s} \, \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}$$

where $\tilde{G}$ is a positive, even, function, decreasing to zero at infinity, and λ a small, positive number.

Main result

(Duchon, 1977; Meinguet, 1979; Wahba, 1977; Madych and Nelson, 1990; Poggio and Girosi, 1989; Girosi, 1992)

The function that minimizes the functional

$$H[f] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 + \lambda \int_{R^d} d\mathbf{s} \, \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}$$

has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) + \sum_{\alpha=1}^{k} d_\alpha \gamma_\alpha(\mathbf{x})$$

where G is conditionally positive definite of order m, $\tilde{G}$ is the Fourier transform of G and $\{\gamma_\alpha\}_{\alpha=1}^{k}$ is a basis in the space of polynomials of degree $m-1$.

Computation of the coefficients

The coefficients are found by solving the linear system:

$$(G + \lambda I)\mathbf{c} + \Gamma^T \mathbf{d} = \mathbf{y}$$

$$\Gamma \mathbf{c} = 0$$

where I is the identity matrix, and we have defined

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_i = c_i \ , \quad (\mathbf{d})_i = d_i$$

$$(G)_{ij} = G(\mathbf{x}_i - \mathbf{x}_j) \ , \quad (\Gamma)_{\alpha i} = \gamma_\alpha(\mathbf{x}_i)$$

Radial Basis Functions

If the function $\tilde{G}$ is radial ($\tilde{G} = \tilde{G}(\|\mathbf{s}\|)$ the smoothness functional ϕ is rotationally invariant, that is

$$\phi[f(\mathbf{x})] = \phi[f(R\mathbf{x})] \quad \forall R \in O(d)$$

where $O(d)$ is the group of rotations in d dimensions.

The regularization solution becomes the so called Radial Basis Functions technique:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x})$$

Radial Basis Functions

Define $r = \|\mathbf{x}\|$

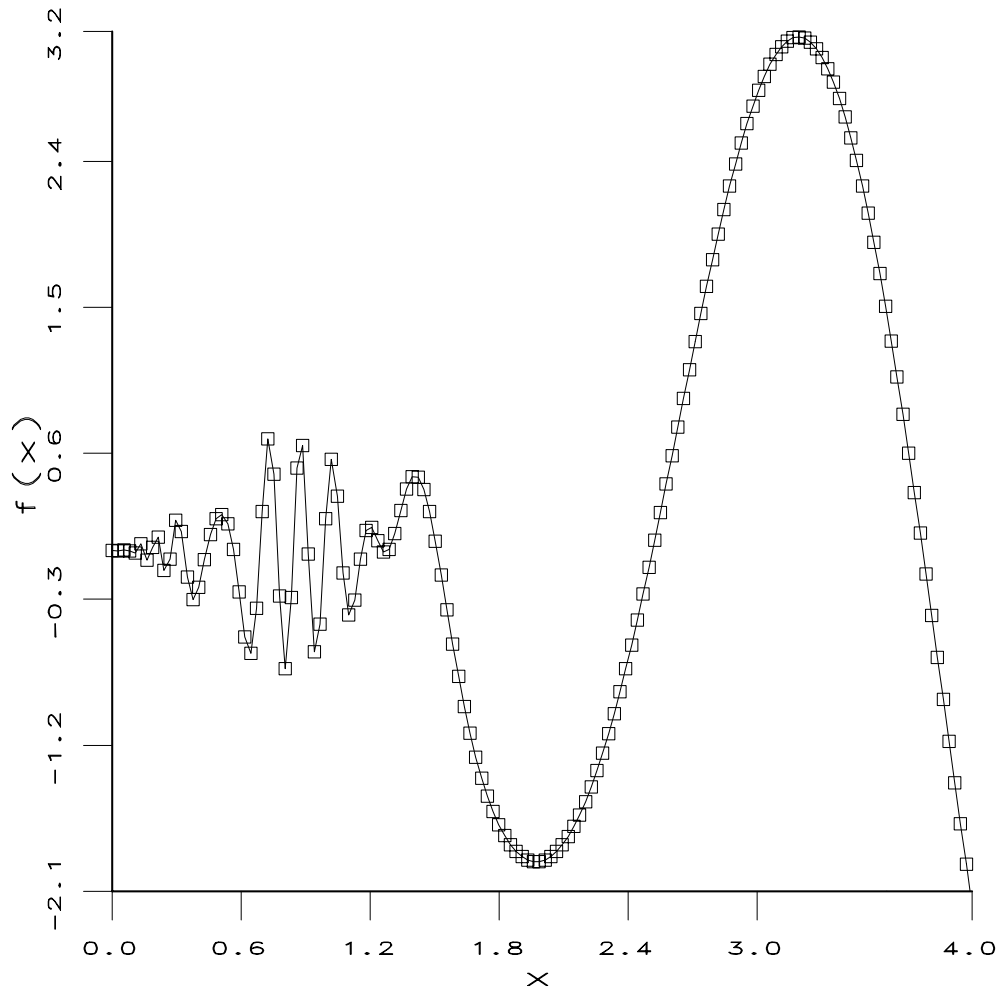
$G(\mathbf{x}) = e^{-r^2}$	gaussian
$G(\mathbf{x}) = \sqrt{r^2 + c^2}$	multiquadric
$G(\mathbf{x}) = \frac{1}{\sqrt{c^2+r^2}}$	inverse multiquadric
$G(\mathbf{x}) = r^{2n+1}$	multivariate splines
$G(\mathbf{x}) = r^{2n} \ln r$	multivariate splines

Some good properties of RBF

- Very well motivated in the framework of regularization theory;
- The solution is unique and equivalent to solve a linear system;
- Amount of smoothness is tunable (with λ);
- Radial Basis Functions are *universal approximators*;
- Huge amount of literature on this subject;
- (Interpretation in terms of "neural networks");
- Biologically plausible;
- Simple interpretation in terms of "smooth look-up table";
- Similar to other non-parametric techniques, such as nearest neighbor and kernel regression;

Some not-so-good properties of RBF

- Computationally expensive ($O(N^3)$ where N =number of data);
- Linear system often badly ill-conditioned;
- The same amount of smoothness is imposed on different regions of the domain;



This function has different smoothness properties in different regions of its domain

Least Squares Regularization Networks

We look for an *approximation* to the regularization solution:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i)$$

$$\Downarrow$$

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{x} - \mathbf{t}_\alpha)$$

where $n << N$ and the vectors $\mathbf{t}_\alpha$ are called *centers*.

(Broomhead and Lowe, 1988; Moody and Darken, 1989; Poggio and Girosi, 1989)

Least Squares Regularization Networks

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{x} - \mathbf{t}_\alpha)$$

Suppose the centers $\mathbf{t}_\alpha$ have been fixed.

How do we find the coefficients c_α?

$\Downarrow$

Least Squares

Least Squares Regularization Networks

Define

$$E(c_1, \ldots, c_n) = \sum_{i=1}^{N} (y_i - f^*(\mathbf{x}_i))^2$$

The least squares criterion is

$$\min_{c_\alpha} \; E(c_1, \ldots, c_n)$$

The problem is convex and quadratic in the c_α, and the solution satisfies:

$$\frac{\partial E}{\partial c_\alpha} = 0$$

Least Squares Regularization Networks

$$\frac{\partial E}{\partial c_\alpha} = 0$$

$$\Downarrow$$

$$G^T G \ \mathbf{c} = G^T \ \mathbf{y}$$

where we have defined

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_\alpha = c_\alpha \ , (G)_{i\alpha} = G(\mathbf{x}_i - \mathbf{t}_\alpha)$$

Therefore $\mathbf{c} = G^+ \mathbf{y}$, where

$$G^+ = (G^T G)^{-1} G^T$$

is the *pseudoinverse* of G.

Least Squares Regularization Networks

Given the centers $\mathbf{t}_\alpha$ we know how to find the c_α.

How do we choose the $\mathbf{t}_\alpha$?

1. a subset of the examples;

2. by a clustering algorithm (k-means, for examples);

3. by least squares ("moving centers");

Centers as a subset of the examples

Fair technique. The subset is a random subset, which should reflect the distribution of the data.

Not many theoretical results availabl.

Main problem: how many centers?

Main answer: we don't know. Cross validation techniques seem a reasonable choice.

Clustering

Clustering a set of data points $\{\mathbf{x}_i\}_{i=1}^N$ in R^d means finding a set of m *representative points* $\{\mathbf{m}_k\}_{k=1}^m$.

Let S_k be the Voronoi polytopes of center $\mathbf{m}_k$:

$$S_k = \{\mathbf{x} : \|\mathbf{x} - \mathbf{m}_k\| < \|\mathbf{x} - \mathbf{m}_j\|, j \neq k\}$$

A set of representative points $\mathcal{M} = \{\mathbf{m}_k\}_{k=1}^m$ is optimal if solves:

$$\min_{\mathcal{M}} \sum_{k=1}^{m} \sum_{\mathbf{x}_i \in S_k} \|\mathbf{x}_i - \mathbf{m}_k\|$$

K-means algorithm

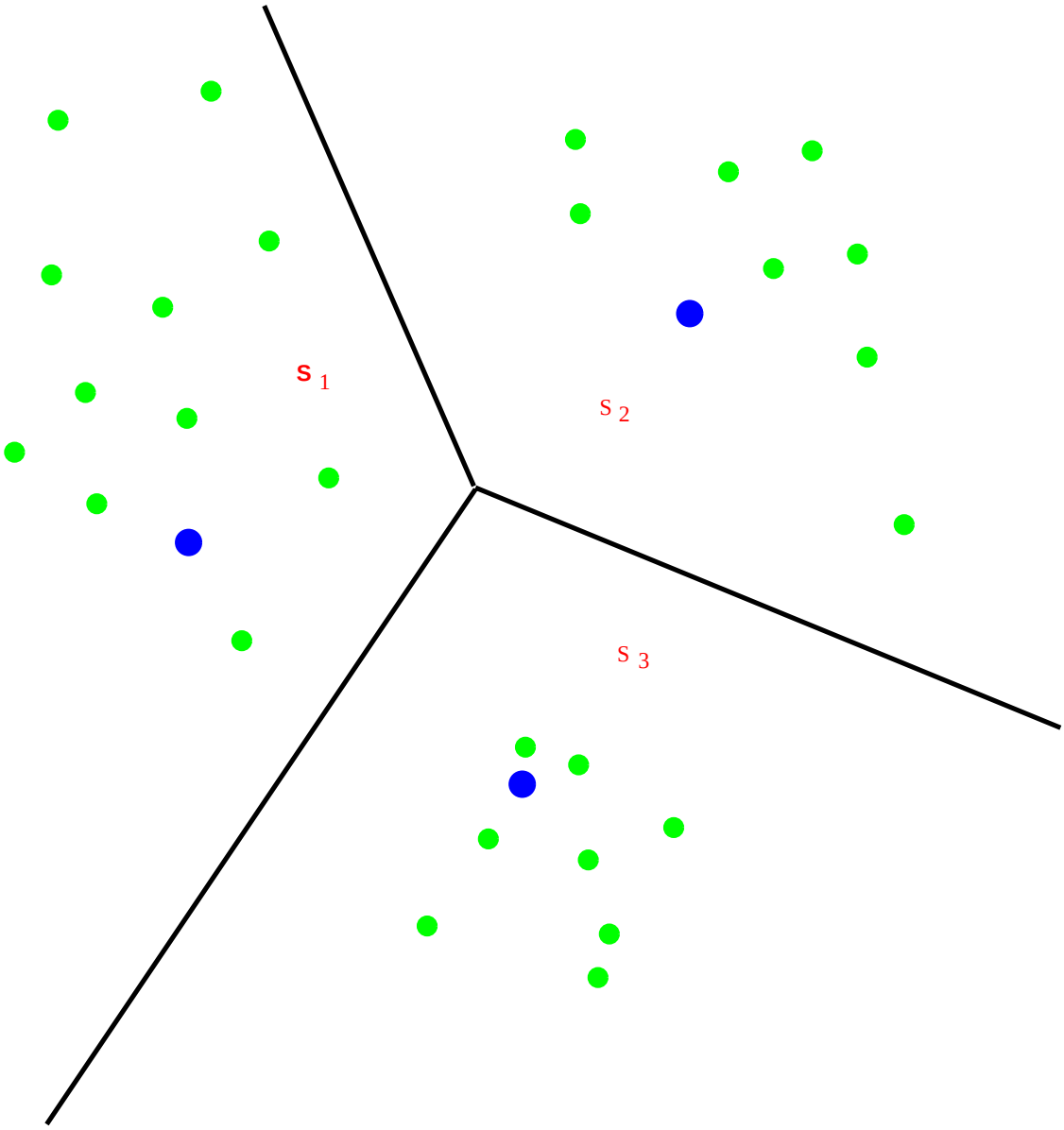
The *k-means algorithm* (MacQueen, 1967) is an iterative technique for finding optimal representative points (cluster centers):

$$\mathbf{m}_k^{(t+1)} = \frac{1}{\#S_k^{(t)}} \sum_{\mathbf{x}_i \in S_k^{(t)}} \mathbf{x}_i$$

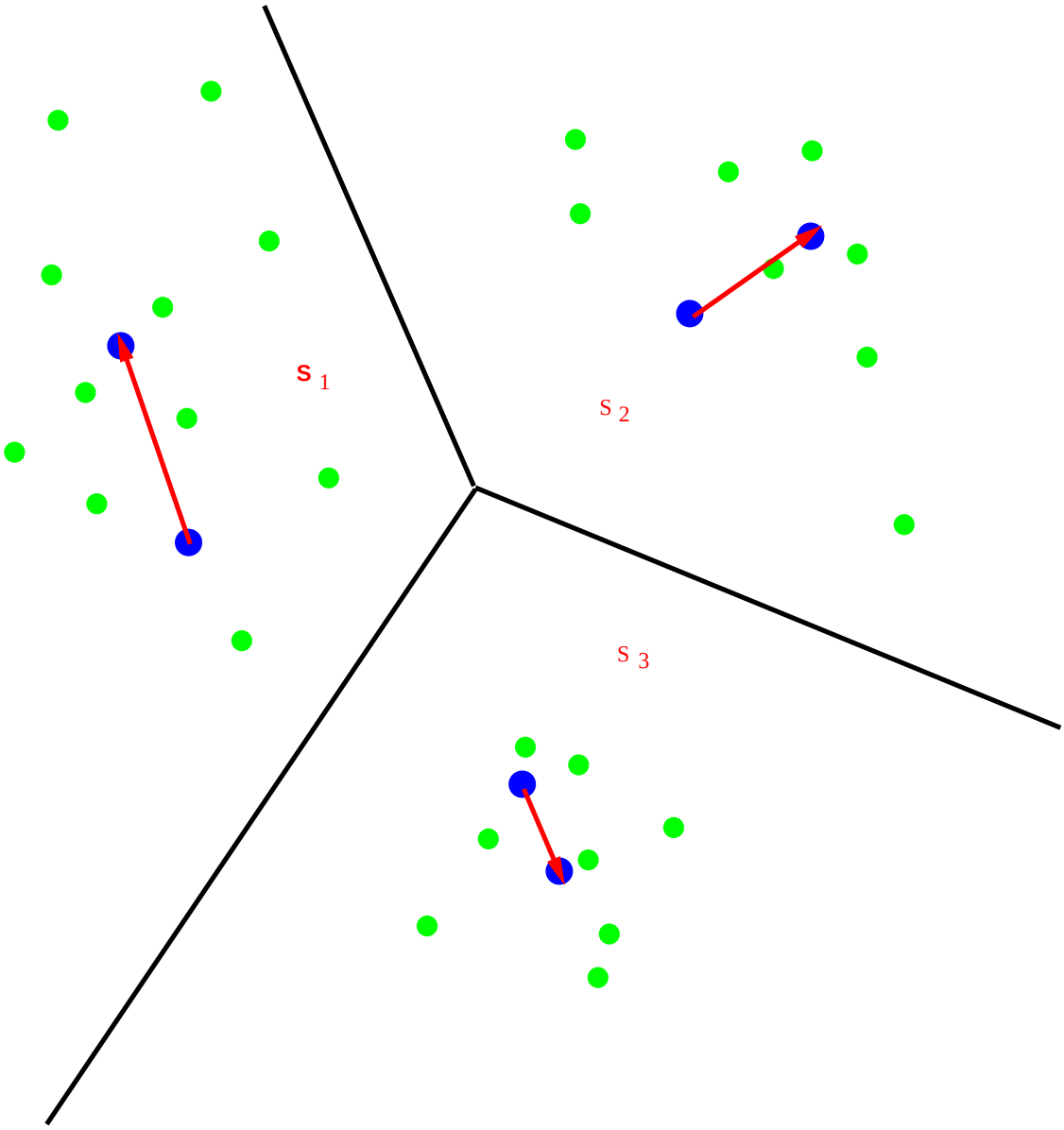
This algorithm is guaranteed to find a local mininum.

Many variations of this algorithm have been proposed.

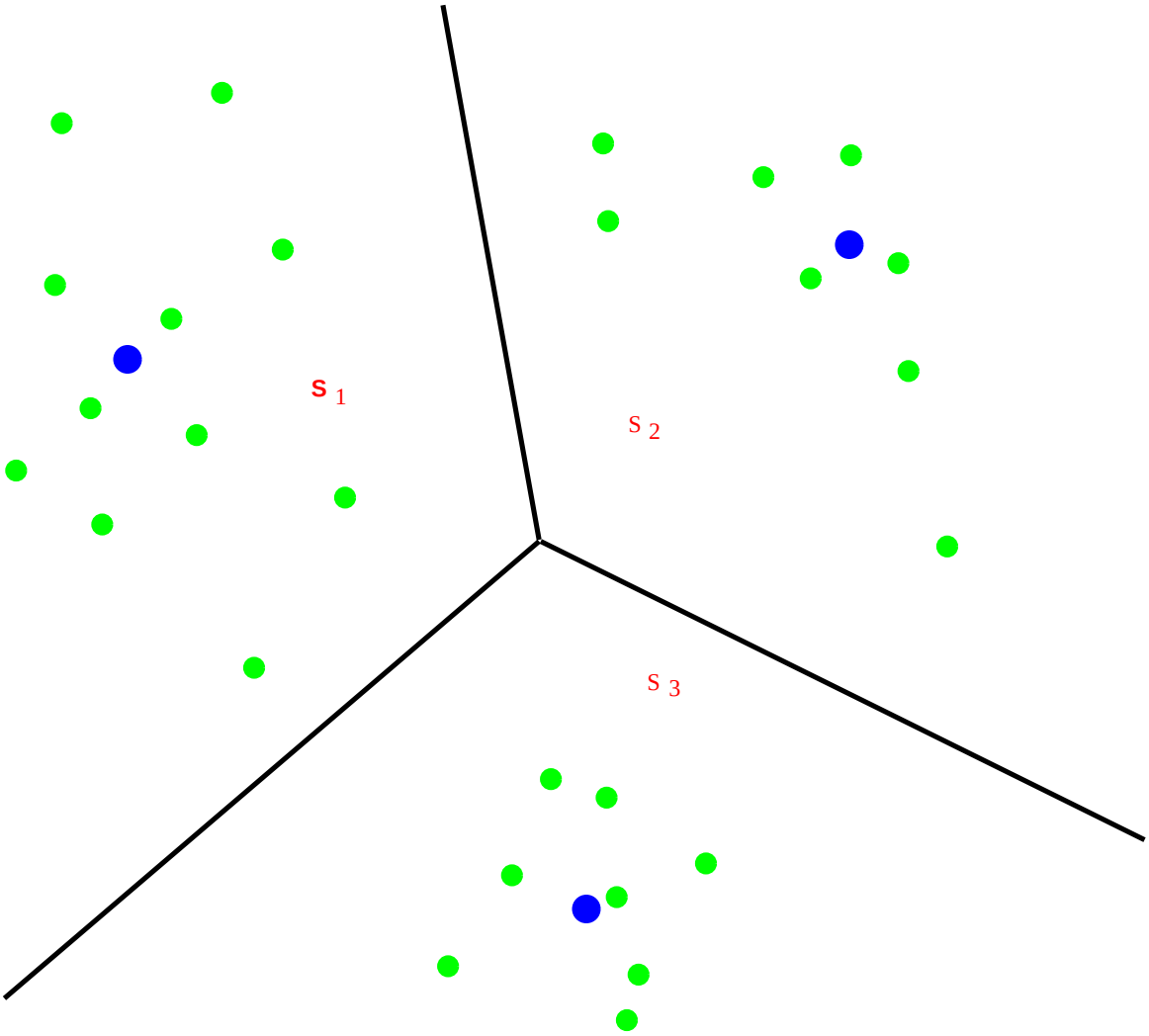
K-means algorithm



K-means algorithm



K-means algorithm



Finding the centers by clustering

Very common. However it makes sense only if the input data points are clustered.

No theoretical results.

Not clear that it is a good idea, especially for pattern classification cases.

Moving centers

Define

$$E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n) = \sum_{i=1}^{N} (y_i - f^*(\mathbf{x}_i))^2$$

The least squares criterion is

$$\min_{c_\alpha, \mathbf{t}_\alpha} E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n)$$

The problem is not convex and quadratic anymore: expect multiple local minima.

Moving centers

:-) Very flexible, in principle very powerful;

:-) Some theoretical understanding;

:-(Very expensive computationally due to the local minima problem;

:-(Centers sometimes move in "weird" ways;

Some new approximation schemes

Radial Basis Functions with moving centers is a particular case of a function approximation technique of the form:

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha H(\mathbf{x}, \mathbf{p}_\alpha)$$

where $\{\mathbf{p}_\alpha\}_{\alpha=1}^{n}$ is a set of parameters, which can be estimated by least squares techniques.

Radial Basis Functions corresponds to the choice

$$H(\mathbf{x}, \mathbf{p}_\alpha) = G(\|\mathbf{x} - \mathbf{p}_\alpha\|)$$

Neural Networks

Neural networks correspond to a different choice. We set $\mathbf{p}_\alpha \equiv (\mathbf{w}_\alpha, \theta_\alpha)$ and choose:

$$H(\mathbf{x}, \mathbf{p}_\alpha) = \sigma(\mathbf{x} \cdot \mathbf{w}_\alpha + \theta_\alpha)$$

where σ is a *sigmoidal* function. The resulting approximation scheme is

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha \sigma(\mathbf{x} \cdot \mathbf{w}_\alpha + \theta_\alpha)$$

and it is called *Multilayer Perceptron with one layer of hidden units*

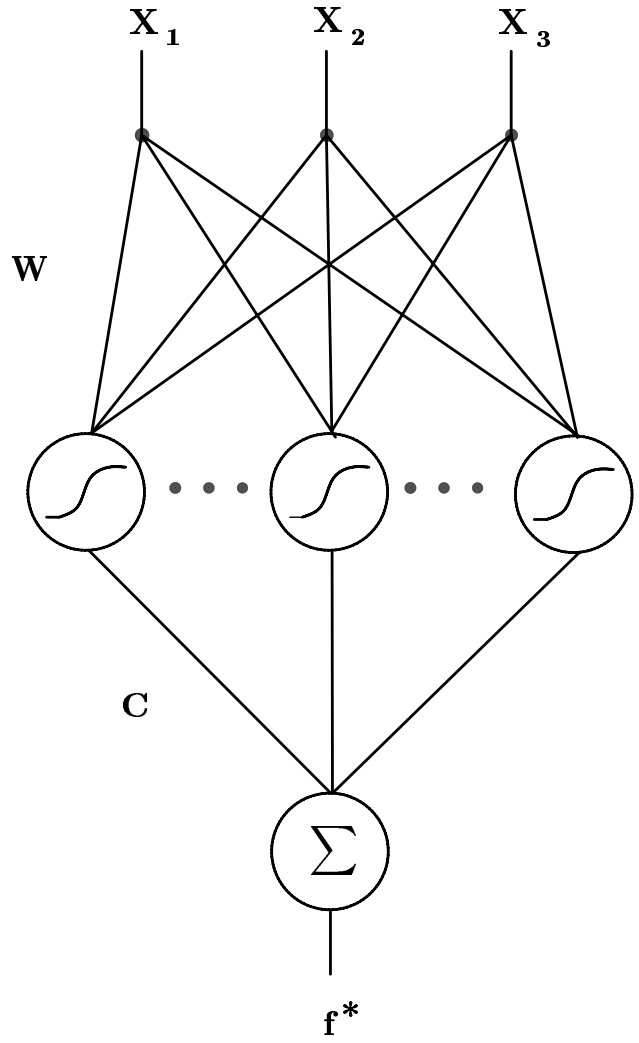
Examples of sigmoidal functions

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$

$$f(x) = \tanh(\beta x)$$

Notice that when β is very large the sigmoid approaches a step function.

One hidden layer Perceptron



Multilayer Perceptrons

:-) Approximation schemes of this type have been successful in a number of cases;

:-(Interpretation of the approximation technique is not immediate;

:-) Multilayer Perceptrons are "universal approximator";

:-) Some theoretical results available;

:-(Computationally expensive, local minima;

:-— Motivation of this technique?

Multilayer perceptrons are particular cases of the more general **ridge function approximation** techniques:

$$f(\mathbf{x}) = \sum_{\mu=1}^{n} h_\mu(\mathbf{x} \cdot \mathbf{w}_\mu)$$

in which we have chosen

$$h_\mu(x) = c_\mu h(x + \theta_\mu)$$

for some fixed function h (sigmoid).

Projection ursuit Regression (Friedman amd Stuetzle, 1981; Huber, 1986)

Look for an approximating function of the form

$$f(\mathbf{x}) = \sum_{\mu=1}^{n} h_\mu(\mathbf{x} \cdot \mathbf{w}_\mu)$$

where $\mathbf{w}_\mu$ are unit vectors and h_μ are functions to be fitted to the data (often cubic splines, trigonometric polynomials or and Radial Basis Functions).

The underlying idea is that all the information lies in few, one dimensional projections of the data.

The unit vectors $\mathbf{w}_\mu$ are the "interesting projections" that have to be pursued.

PPR algorithm

- Assume that the first $\nu - 1$ terms of the expansion have been determined, and define the residuals

$$r_i^{(\nu-1)} = y_i - \sum_{\mu=1}^{\nu-1} h_\mu(\mathbf{x}_i \cdot \mathbf{w}_\mu)$$

- Find the ν-th vector $\mathbf{w}_\nu$ and the ν-th function h_ν that solve

$$\min_{\mathbf{w}_\nu, h_\nu} \sum_{i=1}^{N} (r_i^{(\nu-1)} - h_\nu(\mathbf{x}_i \cdot \mathbf{w}_\nu))^2$$

- Go back to the first step and iterate

Motivation of PPR

Cluster analysis of high-dimensional set of data points D: Diaconis and Freedman (1984) show that *for most high-dimensional clouds, most one-dimensional projections are approximately normal*.

In cluster analysis normal means "uninteresting".

Therefore there are *few interesting projections*, that can be found maximizing a "projection index" that measures deviation from normality.

Principal Component Analysis

Let $\{\mathbf{x}_\alpha\}_{\alpha=1}^{N}$ be a set of points in R^d, with d possibly very large.

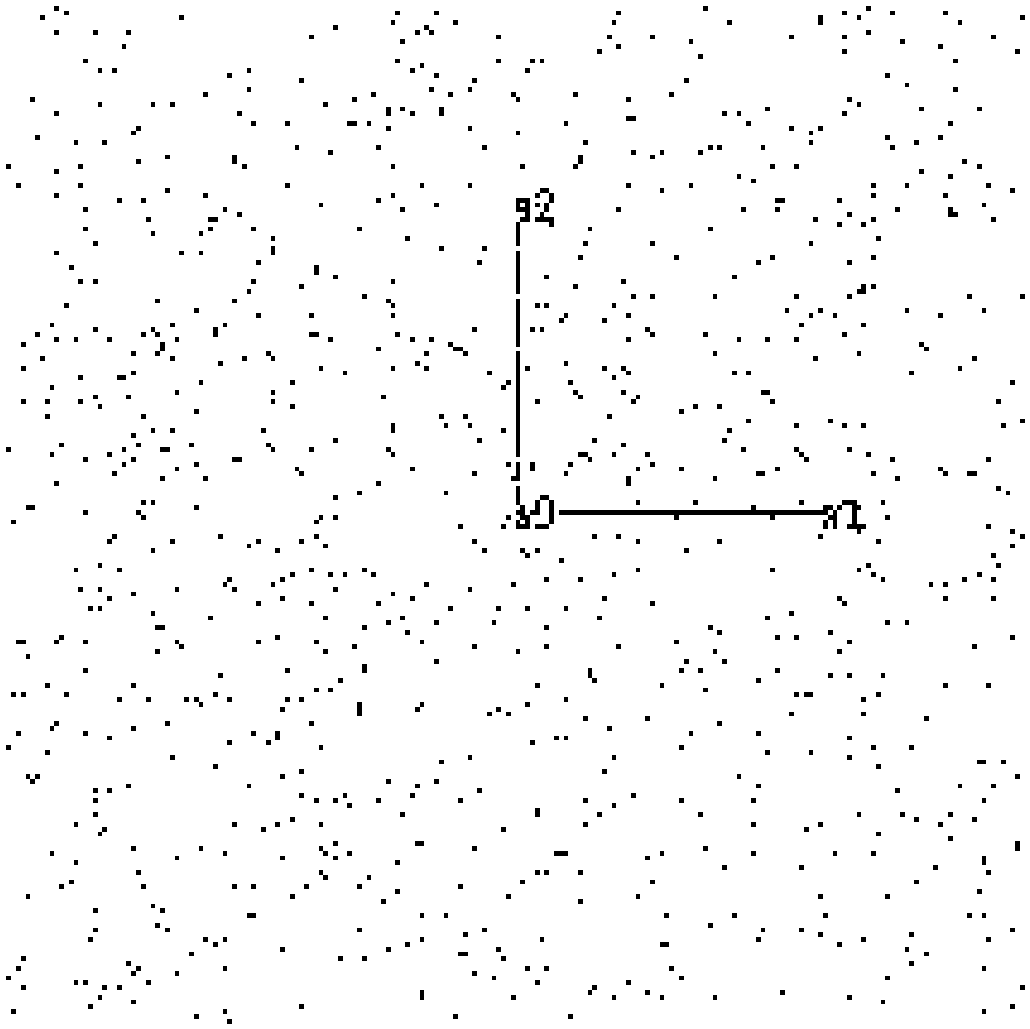
Every point is specified by d numbers.

Is there a more compact representation?

It depends on the distribution of the data points in R^d ...

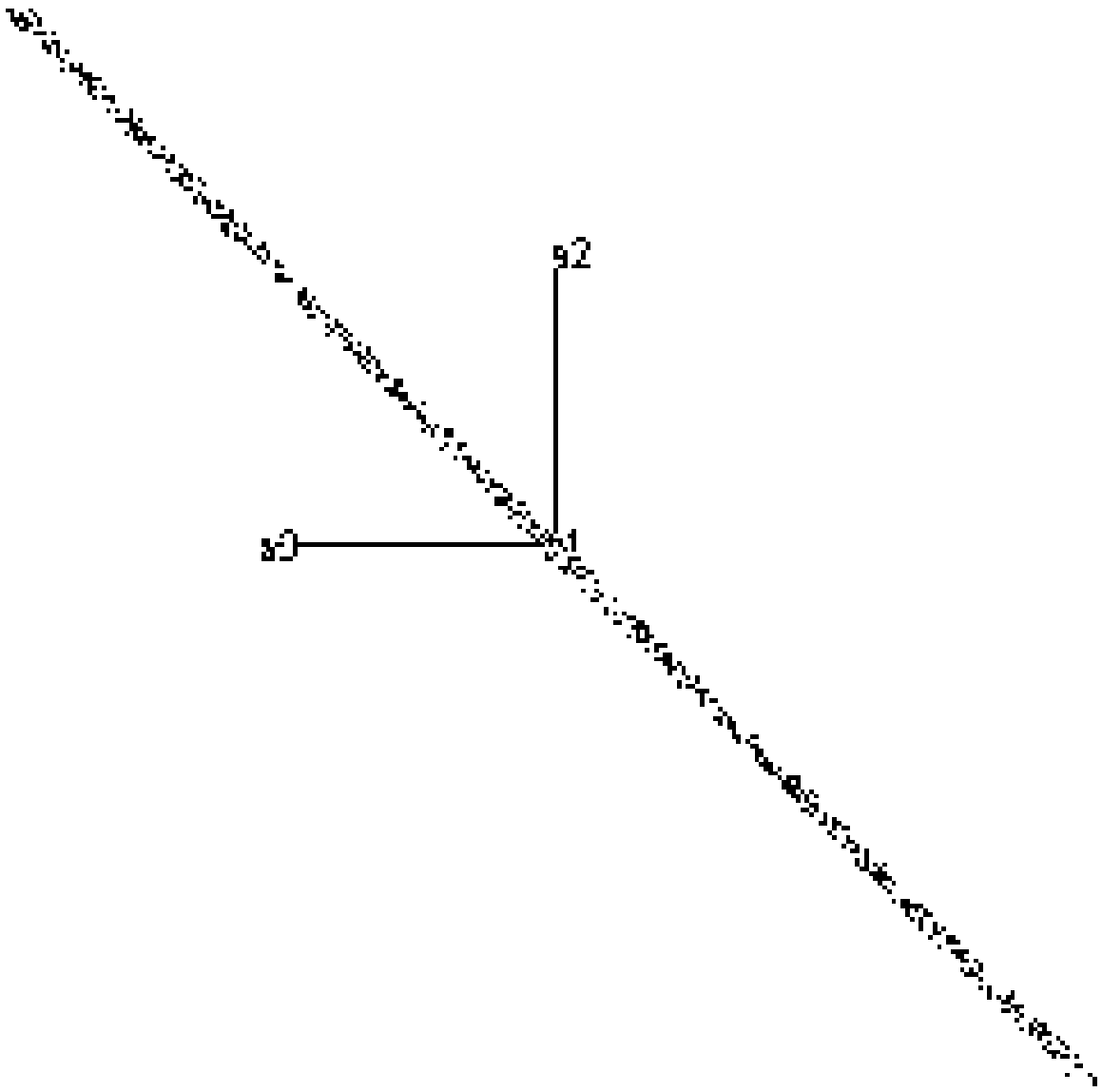
There are random points ...

```
0.058 0.567 0.147
0.337 0.918 0.5
0.005 0.273 0.678
0.112 0.688 0.821
0.948 0.91  0.961
0.867 0.475 0.117
0.013 0.554 0.912
0.655 0.244 0.708
0.405 0.091 0.628
0.285 0.385 0.032
0.034 0.972 0.003
0.529 0.816 0.542
0.973 0.14  0.692
0.263 0.266 0.283
0.403 0.504 0.425
0.498 0.069 0.019
0.172 0.06  0.01
0.352 0.46  0.835
```


x2
x0
x1

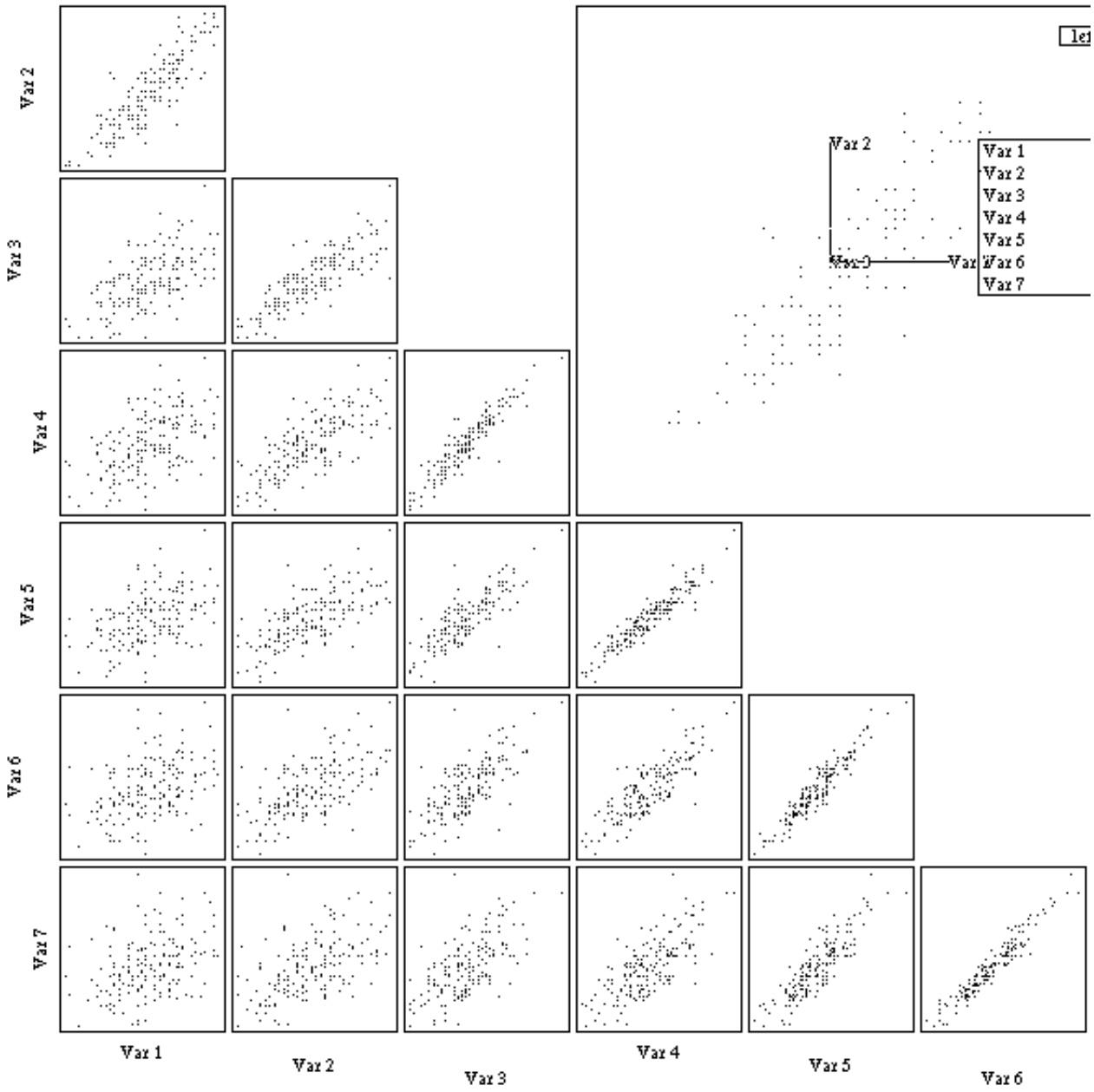
and random points ...

```
0.033 0.42 0.84
0.727 0.577 1.154
0.372 0.957 1.914
0.177 0.45  0.9
0.222 0.562 1.124
0.693 0.644 1.288
0.985 0.805 1.61
0.167 0.766 1.532
0.701 0.078 0.156
0.768 0.791 1.582
0.51 0.874  1.748
0.432 0.375 0.75
0.942 0.251 0.502
0.805 0.795 1.59
0.373 0.181 0.362
0.189 0.541 1.082
0.77 0.724  1.448
```


g2
g3

Nr.	Feature
1	pupil to nose vertical distance
2	pupil to mouth vertical distance
3	pupil to chin vertical distance
4	nose width
5	mouth width
6	face width halfway between nose tip and eyes
7	face width at nose position
8-13	chin radii
14	mouth height
15	upper lip thickness
16	lower lip thickness
17	pupil to eyebrow separation
18	eyebrow thickness

Features 8 to 14



Principal Component Analysis

The data might lie (at least approximately) in a k-dimensional linear subspace of R^d, with $k << d$:

$$\mathbf{x}_\alpha \approx \sum_{i=1}^{k} (\mathbf{x}_\alpha \cdot \mathbf{d}_i)\mathbf{d}_i$$

where $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^k$ is an orthonormal set of vectors:

$$\mathbf{d}_i \cdot \mathbf{d}_j = \delta_{ij}$$

Principal Component Analysis let us find *the best orthonormal set of vectors* $\{\mathbf{d}_i\}_{i=1}^k$, which are called the *first k principal components*.

The orthonormal basis $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^k$ is found by solving the following *least squares* problem:

$$\min_{\mathcal{D}} \sum_{\alpha=1}^{N} (\mathbf{x}_\alpha - \sum_{i=1}^{k} (\mathbf{x}_\alpha \cdot \mathbf{d}_i)\mathbf{d}_i)^2 =$$

$$\max_{\mathcal{D}} \sum_{\alpha=1}^{N} \sum_{i=1}^{k} (\mathbf{x}_\alpha \cdot \mathbf{d}_i)^2 =$$

$$= \frac{1}{N} \max_{\mathcal{D}} \sum_{i=1}^{k} \mathbf{d}_i \cdot C\mathbf{d}_i$$

where C is the *correlation matrix*:

$$C_{ij} = \frac{1}{N} \sum_{\alpha=1}^{N} x_\alpha^i x_\alpha^j$$

Let us consider the case $k = 1$, and $\mathbf{d}_1 = \mathbf{d}$. The problem is now to solve:

$$\frac{1}{N} \max_{\|\mathbf{d}\|=1} \mathbf{d} \cdot C\mathbf{d}$$

where C is a symmetrix matrix (correlation matrix). Setting

$$C = R^T \Lambda R \ , \quad \mathbf{d}^* = R\mathbf{d}$$

with R a rotation matrix and Λ diagonal, we have now to solve:

$$\frac{1}{N} \max_{\|\mathbf{d}^*\|=1} \mathbf{d}^* \cdot \Lambda\mathbf{d}^*$$

Notice that, since $\|\mathbf{d}^*\| = 1$, then:

$$\mathbf{d}^* \cdot \Lambda \mathbf{d}^* \leq \lambda_{max}$$

If we choose the vector $\mathbf{d}^*_{max}$ such that $\mathbf{d}^*_{max} = (0, 0, \ldots, 1, 0, 0 \ldots, 0)$ and

$$\Lambda \mathbf{d}^*_{max} = \lambda_{max} \mathbf{d}^*_{max}$$

then

$$\mathbf{d}^*_{max} \cdot \Lambda \mathbf{d}^*_{max} = \lambda_{max}$$

and $\mathbf{d}^*_{max}$ maximizes the quadratic form $\mathbf{d}^* \cdot \Lambda \mathbf{d}^*$.

The first principal component, that is the vector $\mathbf{d}_{max}$ that maximizes the quadratic form $\mathbf{d} \cdot C\mathbf{d}$, is therefore

$$\mathbf{d}_{max} = R^T \mathbf{d}^*_{max}$$

and it can be easily seen that satisfies the eigenvector equation:

$$C\mathbf{d}_{max} = \lambda_{max}\mathbf{d}_{max}$$

The first principal component is therefore the eigenvector of the correlation matrix corresponding to the maximum eigenvalue.

Case $k > 1$

It can be shown that *the first k principal components are the eigenvectors of the correlation matrix C corresponding to the first k largest eigenvalues*.

Try this at home:

The approximation error of the first k principal components $\{\mathbf{d}_i\}_{i=1}^{k}$

$$E(k) = \sum_{\alpha=1}^{N} (\mathbf{x}_\alpha - \sum_{i=1}^{k} (\mathbf{x}_\alpha \cdot \mathbf{d}_i)\mathbf{d}_i)^2$$

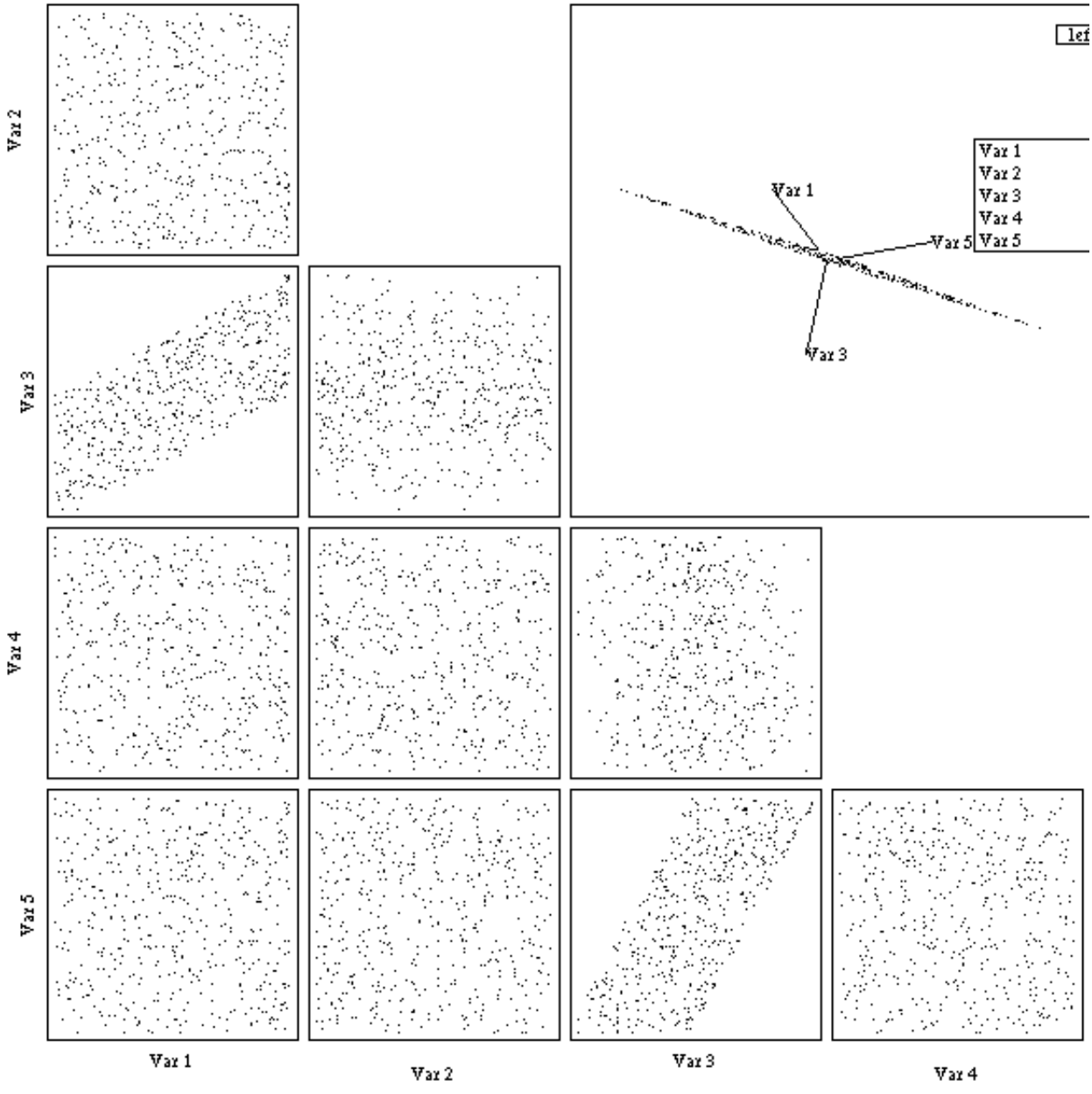
is given by

$$E(k) = \sum_{i=k+1}^{N} \lambda_i$$

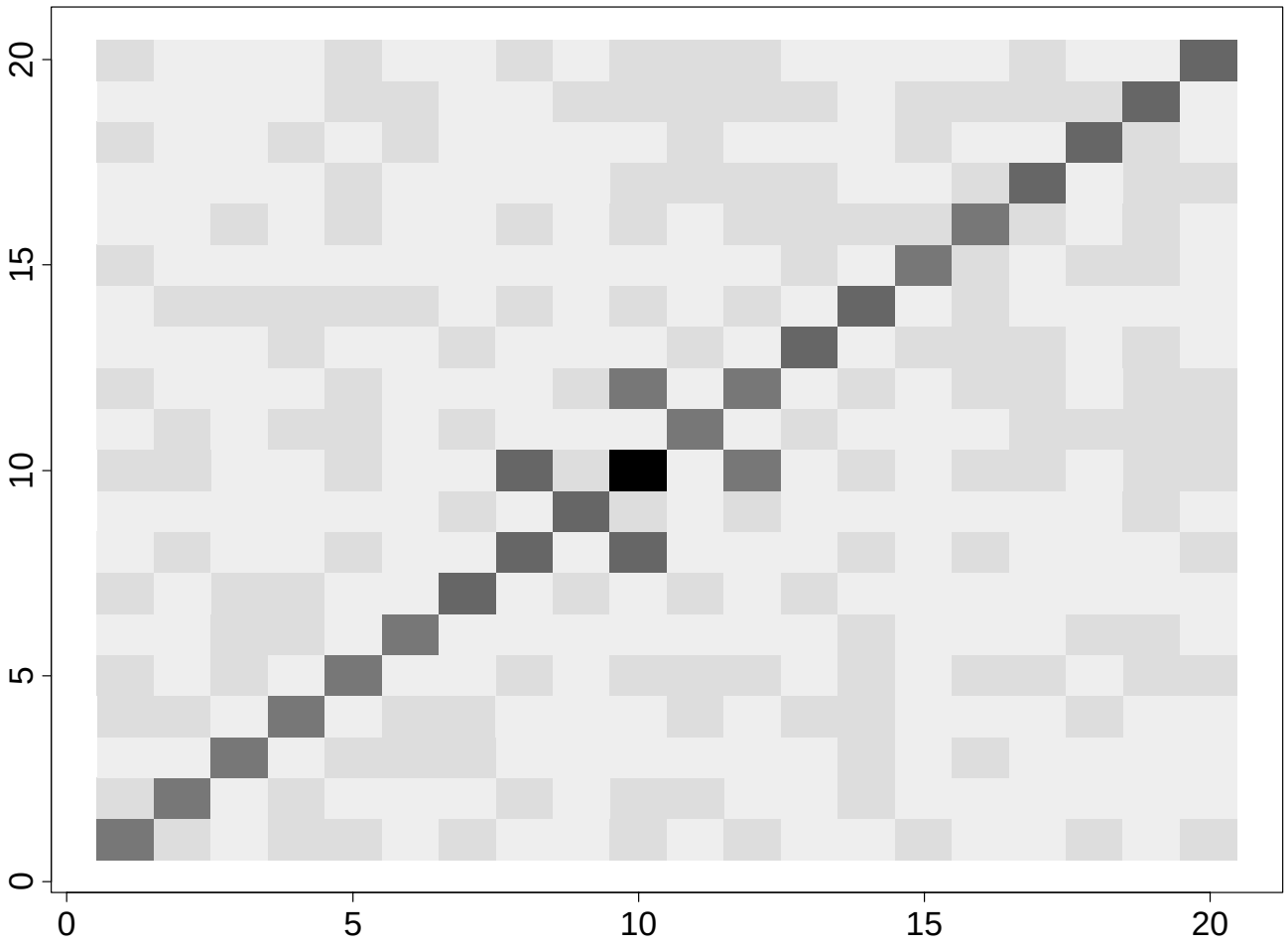
where λ_i are the eigenvalues of the correlation matrix C.

Hint: when $k = N$ the error is zero because the correlation matrix C is symmetric, and its eigenvectors form a complete basis.

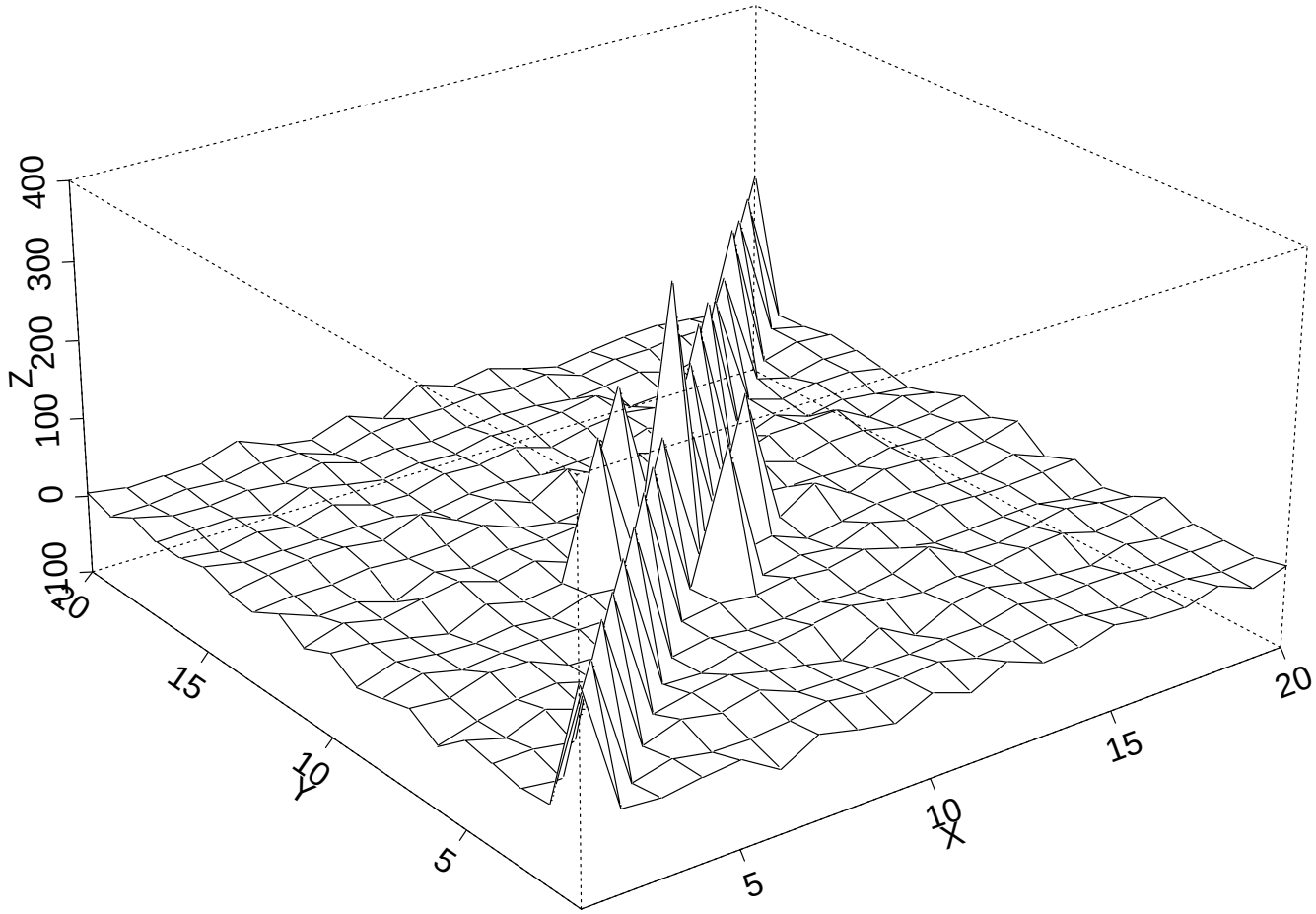
An example in 20 dimensions: $\mathbf{x} \in R^{20}$, $x^{10} = x^8 + x^{12}$, 500 points.



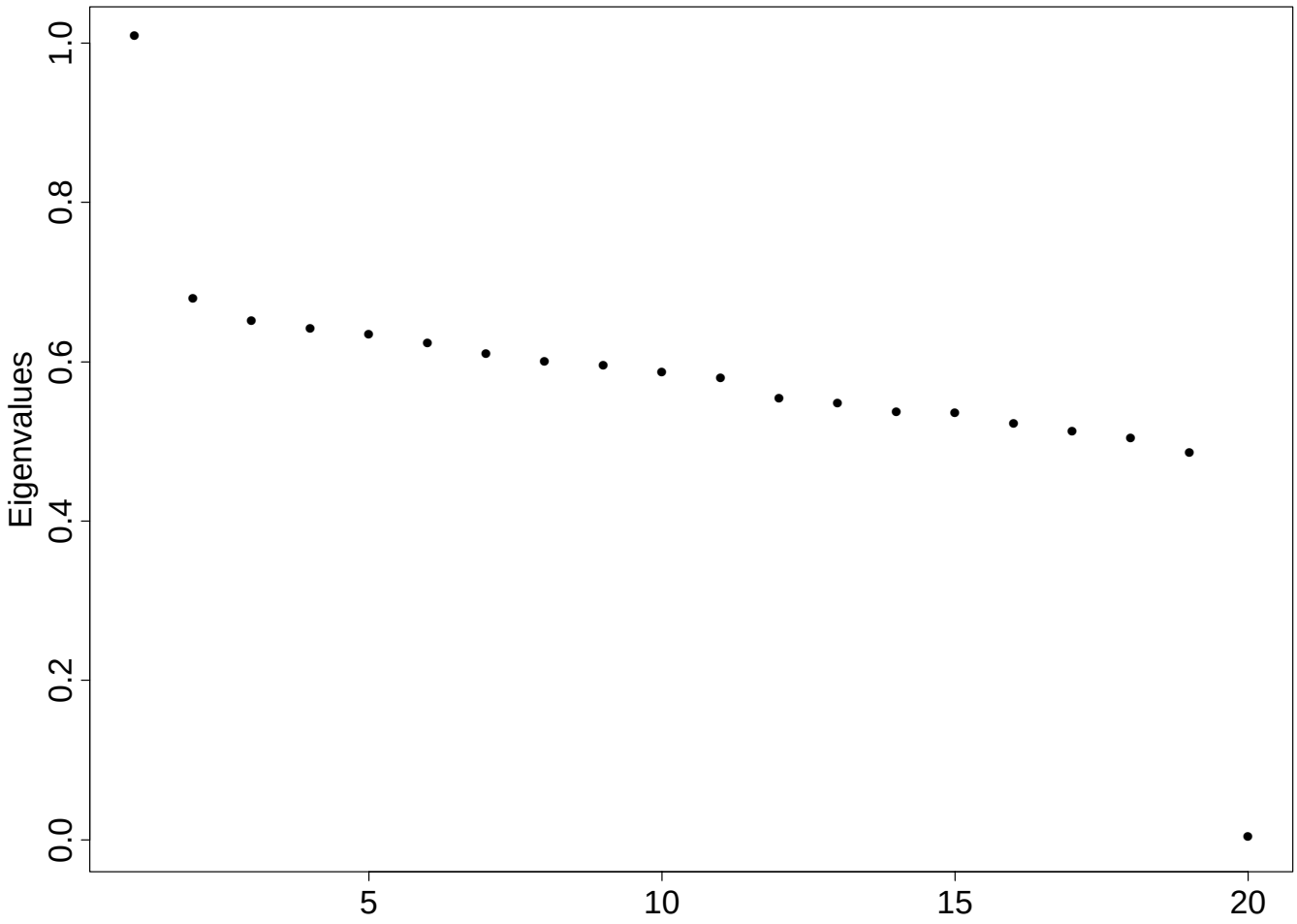
Correlation matrix



Correlation matrix



Eigenvalues of the correlation matrix



Another point of view on PCA

Let $\{\mathbf{x}_\alpha\}_{\alpha=1}^N$ be a set of points in R^d.

Problem: *find the unit vector* $\mathbf{d}$ *for which the projection of the data on* $\mathbf{d}$ *has the maximum variance*.

Assuming that the data points have zero mean, this means:

$$\max_{\|\mathbf{d}\|=1} \frac{1}{N} \sum_{\alpha=1}^{N} (\mathbf{x}_\alpha \cdot \mathbf{d})^2$$

This is formally the same as finding the *first principal component*.

The second principal component is the projection of maximum variance in a subspace orthogonal to the first principal component.

...

The k-th principal component is the projection of maximum variance in a subspace wich is orthogonal to the subspace spanned by the first $k-1$ principal components.

Extensions of Radial Basis Functions

- Different variables can have different scales: $f(x, y) = y^2 \sin(100x)$;

- Different variables could have different units of measure $f = f(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$;

- Not all the variables are independent or relevant: $f(x, y, z, t) = g(x, y, z(x, y))$;

- Only some linear combinations of the variables are relevant: $f(x, y, z) = \sin(x + y + z)$;

Extensions of regularization theory

A priori knowledge:

- the relevant variables are linear combination of the original ones:

  $$\mathbf{z} = \mathbf{W}\mathbf{x}$$

  for some (possibly rectangular) matrix $\mathbf{W}$;

- $f(\mathbf{x}) = g(\mathbf{W}\mathbf{x}) = g(\mathbf{z})$ and the function g is smooth;

The regularization functional is now

$$H[g] = \sum_{i=1}^{N} (y_i - g(\mathbf{z}_i))^2 + \lambda\phi[g]$$

where $\mathbf{z}_i = \mathbf{W}\mathbf{x}_i$.

Extensions of regularization theory (continue)

The solution is

$$g(\mathbf{z}) = \sum_{i=1}^{N} c_i G(\mathbf{z} - \mathbf{z}_i) \ .$$

Therefore the solution for f is:

$$f(\mathbf{x}) = g(\mathbf{W}\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{x}_i)$$

If the matrix **W** were known, the coefficients could be computed as in the radial case:

$$(G + \lambda I)\mathbf{c} = \mathbf{y}$$

where

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_i = c_i \ , (G)_{ij} = G(\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j)$$

and the same criticisms of the Regularization Networks technique apply, leading to *Generalized Regularization Networks*:

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{t}_\alpha)$$

Since $\mathbf{W}$ is not known, it could be found by *least squares*. Define

$$E(c_1, \cdots, c_n, \mathbf{W}) = \sum_{i=1}^{N} (y_i - f^*(\mathbf{x}_i))^2$$

Then we can solve:

$$\min_{c_\alpha, \mathbf{W}} \; E(c_1, \cdots, c_n, \mathbf{W})$$

The problem is not convex and quadratic anymore: expect multiple local minima.

From RBF to HyperBF

When the basis function is radial the Generalized Regularization Networks becomes

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\|\mathbf{x} - \mathbf{t}_\alpha\|_{\mathbf{W}})$$

that is a *non radial basis function* technique

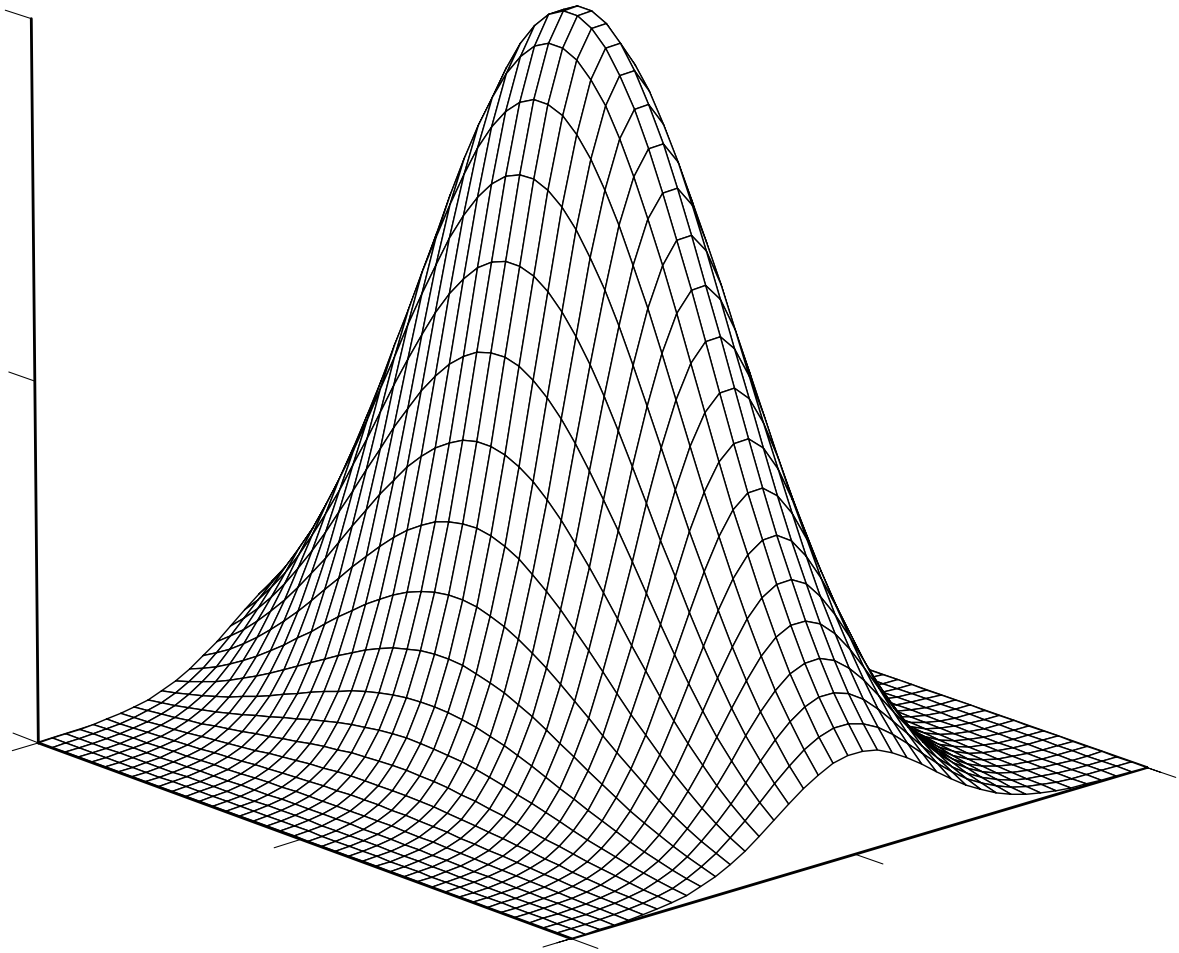
Least Squares

1. $\min_{c_\alpha} \; E(c_1, \ldots, c_n)$

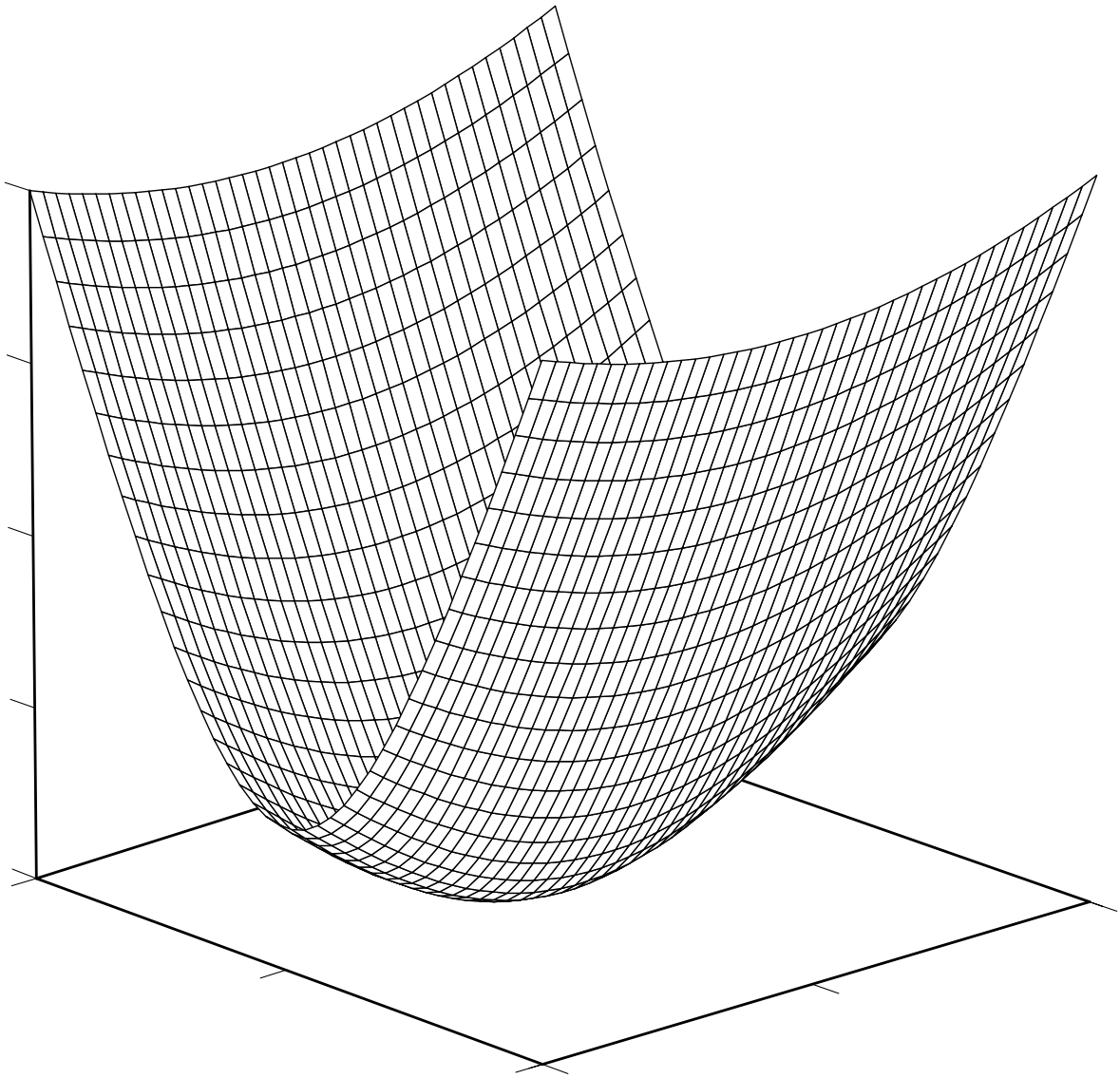
2. $\min_{c_\alpha, \mathbf{t}_\alpha} \; E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n)$

3. $\min_{c_\alpha, \mathbf{W}} \; E(c_1, \cdots, c_n, \mathbf{W})$

4. $\min_{c_\alpha, \mathbf{t}_\alpha, \mathbf{W}} E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n, \mathbf{W})$



A non radial gaussian function



A non radial multiquadric function

Additive models

An additive model has the form

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu)$$

where

$$f_\mu(x^\mu) = \sum_{i=1}^{N} c_i^\mu G(x^\mu - x_i^\mu)$$

In other words

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} \sum_{i=1}^{N} c_i^\mu G(x^\mu - x_i^\mu)$$

Extensions of Additive Models

If we have less centers than examples we obtain:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu)$$

where

$$f_\mu(x^\mu) = \sum_{\alpha=1}^{N} c_\alpha^\mu G(x^\mu - t_\alpha^\mu)$$

In other words

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} \sum_{\alpha=1}^{N} c_\alpha^\mu G(x^\mu - t_\alpha^\mu)$$

Extensions of Additive Models

If we now allow for an arbitrary linear transformation of the inputs:

$$\mathbf{x} \rightarrow \mathbf{W}\mathbf{x}$$

where $\mathbf{W}$ is a $d' \times d$ matrix, we obtain:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} \sum_{\alpha=1}^{n} c_\alpha^\mu G(\mathbf{x} \cdot \mathbf{w}_\mu - t_\alpha^\mu)$$

where $\mathbf{w}_\mu$ is the μ-th row of the matrix $\mathbf{W}$,

Extensions of Additive Models

The expression

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} \sum_{\alpha=1}^{n} c_\alpha^\mu G(\mathbf{x} \cdot \mathbf{w}_\mu - t_\alpha^\mu)$$

can be written as

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} h_\mu(\mathbf{x} \cdot \mathbf{w}_\mu)$$

where

$$h_\mu(y) = \sum_{\alpha=1}^{n} c_\alpha^\mu G(y - t_\alpha^\mu)$$

This form of approximation is called **ridge approximation**

From the extension of additive models we can therefore justify an approximation technique of the form

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} \sum_{\alpha=1}^{n} c_{\alpha}^{\mu} G(\mathbf{x} \cdot \mathbf{w}_{\mu} - t_{\alpha}^{\mu})$$

Particular case: $n = 1$ (one center). Then we derive the following technique:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} c^{\mu} G(\mathbf{x} \cdot \mathbf{w}_{\mu} - t\mu)$$

which is a Multilayer Perceptron with a Radial Basis Functions G instead of the sigmoid function.

Notice that the sigmoid function is not a Radial Basis Functions

Regularization Networks