ELEC2870 - Machine learning: regression and dimensionality reduction

Nonlinear regression with Radial-Basis Function Networks

Michel Verleysen

Machine Learning Group

Université catholique de Louvain

Louvain-la-Neuve, Belgium

michel.verleysen@uclouvain.be

Outline

- Origin: Cover's theorem
- Interpolation problem
- Regularization theory
- Generalized RBFN
  - Universal approximation
  - RBFN = kernel regression
  - Comparison with MLP
- Learning
  - Centers
  - Widths
  - Multiplying factors
  - Other forms

Origin: Covers' theorem

- Covers' theorem on separability of patterns (1965)
- $x^1, x^2, \ldots, x^P$ assigned to two classes C^1 C^2
- φ-separability:

$$\exists w \;\Big|\; \begin{cases} w^T \varphi(x) > 0 & x \in C^1 \\ w^T \varphi(x) < 0 & x \in C^2 \end{cases}$$

- Cover's theorem:
  - if functions $\varphi_i(x)$ are nonlinear
  - if number of functions $\varphi_i(x)$ > dimension input space
  - $\rightarrow$ then probability of separability closer to 1

- Quite a « natural » theorem:
  - imagine that you take many (really, a lot…) of random transformations $\varphi_i(x)$. They form a (very) high-dimensional space, and with some chance, some features in that space will be linearly separable!

Outline

- Origin: Cover's theorem
- Interpolation problem
- Regularization theory
- Generalized RBFN
  - Universal approximation
  - RBFN = kernel regression
  - Comparison with MLP
- Learning
  - Centers
  - Widths
  - Multiplying factors
  - Other forms

Interpolation problem

- Given points (x^p, t^p), $x^p \in \Re^D$, $t^p \in \Re$, $1 \leq p \leq P$:

- Find $F : \Re^D \rightarrow \Re$ that satisfies

$$F\left(x^p\right) = t^p, \;\; p = 1 \ldots P$$

(so we don't care for regularization, for now)

- RBF technique (Powell, 1988):

$$F(x) = \sum_{p=1}^{P} w_p \varphi\left(\left\| x - x^p \right\|\right)$$

  - $\varphi\left(\left\| x - x^p \right\|\right)$ are arbitrary non-linear functions (RBF)
  - as many functions as data points
  - centers fixed at known points x^p

Interpolation problem

$$F\left(x^p\right) = t^p \qquad F(x) = \sum_{p=1}^{P} w_k \varphi\left(\left\|x - x^p\right\|\right)$$

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1P} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{P1} & \varphi_{P2} & \cdots & \varphi_{PP} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_P \end{bmatrix} = \begin{bmatrix} t^1 \\ t^2 \\ \vdots \\ t^P \end{bmatrix}$$

where

$$\varphi_{kl} = \varphi\left(\left\|x^k - x^l\right\|\right)$$

- Into matrix form: $\Phi w = t \longrightarrow w = \Phi^{-1} t$
- Vital question: is Φ non-singular ?

Michelli's theorem

- If points $\boldsymbol{x}^k$ are distinct, Φ is non-singular (regardless of the dimension of the input space)
- Valid for a large class of RBF functions:

$$\varphi(x,c) = \sqrt{\|x-c\|^2 + l^2} \qquad (l > 0)$$

non-localized function

$$\varphi(x,c) = \frac{1}{\sqrt{\|x-c\|^2 + l^2}}$$

$$\varphi(x,c) = \exp\left(-\frac{\|x-c\|^2}{2\sigma^2}\right) \qquad (\sigma > 0)$$

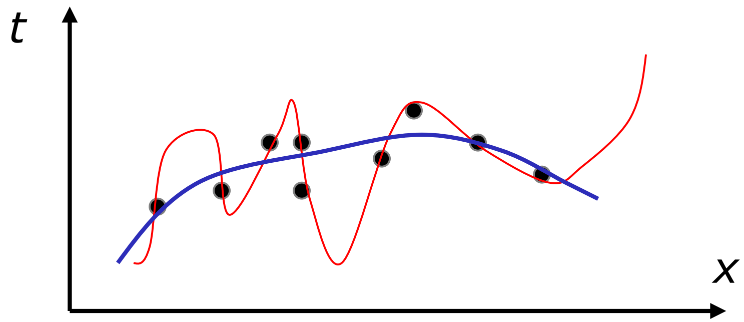
localized functions

Outline

- Origin: Cover's theorem
- Interpolation problem
- Regularization theory
- Generalized RBFN
  - Universal approximation
  - RBFN = kernel regression
  - Comparison with MLP
- Learning
  - Centers
  - Widths
  - Multiplying factors
  - Other forms

But…

- Remember: learning is all in-posed problem
- Nobody cares (or should…) about learning; the challenge is generaliziation!



- Error criterion: $$E(F) = \underbrace{\frac{1}{2P}\sum_{p=1}^{P}\left(t^p - F\left(x^p\right)\right)^2}_{\text{MSE}} + \lambda \underbrace{\frac{1}{2}C(w)}_{\text{regularization}}$$

Solution to the regularization problem

- Poggio & Girosi (1990):
  - if $C(w)$ is a (problem-dependent) linear differential operator, the solution to

$$E(F) = \frac{1}{2P}\sum_{p=1}^{P}\left(t^p - F\left(x^p\right)\right) + \lambda\frac{1}{2}C(w)$$

is of the following form:

$$F(x) = \sum_{p=1}^{P} w_p G\left(x, x^p\right)$$

where $G()$ is a Green's function,

$$w = \left(G + \lambda I\right)^{-1} t$$

$$G_{kl} = G(x^k, x^l)$$

Interpolation - regularization

- Interpolation

$$F(x) = \sum_{p=1}^{P} w_p \varphi\left(\left\| x - x^p \right\|\right)$$

$$w = \Phi^{-1} t$$

  - Exact interpolator
  - Possible RBF:

$$\varphi\left(\left\| x, x^p \right\|\right) = \exp\left(- \frac{\left\| x - x^p \right\|^2}{2\sigma^2} \right)$$

- Regularization

$$F(x) = \sum_{p=1}^{P} w_p G\left(x, x^p\right)$$

$$w = \left(G + \lambda I\right)^{-1} t$$

  - Exact interpolator
  - Equal to the interpolation solution iff $\lambda=0$
  - Example of Green function:

$$G\left(x, x^p\right) = \exp\left(- \frac{\left\| x - x^p \right\|^2}{2\sigma^2} \right)$$

$\boxed{\text{One RBF / Green's function for each learning pattern!}}$

Outline

- Origin: Cover's theorem
- Interpolation problem
- Regularization theory
- Generalized RBFN
  - Universal approximation
  - RBFN = kernel regression
  - Comparison with MLP
- Learning
  - Centers
  - Widths
  - Multiplying factors
  - Other forms

Generalized RBFN (GRBFN – RBFN)

- As many radial functions as learning patterns:
  - computationally (too) intensive (inversion of PxP matrix grows with P^3)
  - ill-conditioned matrix
  - regularization not easy (problem-specific)

→ *Generalized* RBFN approach!

$$F(x) = \sum_{i=1}^{K} w_i \varphi\left(\left\| x - c_i \right\|\right)$$

Typically:

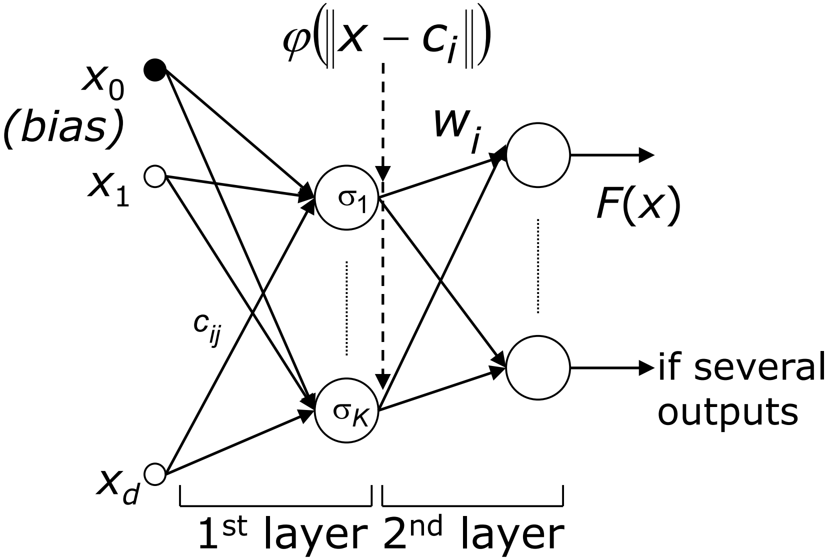
- $K << P$
- $\varphi\left(\left\| \boldsymbol{x} - \boldsymbol{c}_i \right\|\right) = \exp\left(- \frac{\left\| \boldsymbol{x} - \boldsymbol{c}_i \right\|^2}{2\sigma_i^2} \right)$

Parameters:
$\boldsymbol{c}_i$, σ_i, w_i

Radial-Basis Function Networks (RBFN)

$$F(x) = \sum_{i=1}^{K} w_i \varphi(\|x - c_i\|)$$

$$\varphi(\|x - c_i\|) = \exp\left(- \frac{\|x - c_i\|^2}{2\sigma_i^2} \right)$$



- Possibilities:
  - several outputs (common hidden layer)
  - bias (recommended) (see extensions)

RBFN: universal approximation

- Park & Sandberg 1991:
  - *For any continuous input-output mapping function f(**x**)*

$$\exists \;\; F(x) = \sum_{i=1}^{K} w_i \varphi(\|x - c_i\|) \;\Big|\; L_p(f(x), F(x)) < \varepsilon \qquad (\varepsilon > 0, p \in [1, \infty])$$

  - The theorem is stronger (radial symmetry not needed)
  - K not specified
  - Provides a theoretical basis for *practical* RBFN!

RBFN and kernel regression

- A digression about (probabilistic) kernel regression:
- non-linear regression model

$$t^p = f(x^p) + \varepsilon^p = y^p + \varepsilon^p, \ 1 \le p \le P$$

- estimation of $f(\boldsymbol{x})$: average of t around $\boldsymbol{x}$. More precisely:

$$\begin{aligned} f(x) &= E[y|x] \\ &= \int_{-\infty}^{\infty} y f_Y(y|x) dy \\ &= \frac{\int_{-\infty}^{\infty} y f_{X,Y}(x,y) dy}{f_X(x)} \end{aligned}$$

- Need for estimates of $f_{X,Y}(x,y)$ and $f_X(x)$
  → Parzen-Rosenblatt density estimator

Parzen-Rosenblatt density estimator

$$\hat{f}_X(x) = \frac{1}{Ph^d} \sum_{p=1}^{P} K\left(\frac{x - x^p}{h} \right)$$

with $K()$ continuous, bounded, symmetric about the origin, with maximum value at 0, and with unit integral,

is consistent (asymptotically unbiased).

- Estimation of $f_{X,Y}(x,y)$

$$\hat{f}_{X,Y}(x,y) = \frac{1}{Ph^{d+1}} \sum_{p=1}^{P} K\left(\frac{x - x^p}{h} \right) K\left(\frac{y - y^p}{h} \right)$$

RBFN and kernel regression

$$f(x) = \frac{\int_{-\infty}^{\infty} y f_{X,Y}(x,y) dy}{f_X(x)}$$

$$\hat{f}(x) = \frac{\int_{-\infty}^{\infty} y \hat{f}_{X,Y}(x,y) dy}{\hat{f}_X(x)}$$

$$= \frac{\sum_{p=1}^{P} y^p K\left(\frac{x - x^p}{h}\right)}{\sum_{p=1}^{P} K\left(\frac{x - x^p}{h}\right)}$$

- Weighted average of y^i
- called Nadaraya-Watson estimator (1964)
- equivalent to *Normalized RBFN* in the unregularized context

RBFN - MLP

- RBFN
  - single hidden layer
  - non-linear hidden layer
    linear output layer
  - argument of hidden units:
    Euclidean norm
  - universal approximation
    property
  - local approximators
  - splitted learning

- MLP
  - single or multiple hidden layers
  - non-linear hidden layer
    linear or non-linear output layer
  - argument of hidden units:
    scalar product
  - universal approximation
    property
  - global approximators
  - global learning

Outline

- Origin: Cover’s theorem
- Interpolation problem
- Regularization theory
- Generalized RBFN
  - Universal approximation
  - RBFN = kernel regression
  - Comparison with MLP
- Learning
  - Centers
  - Widths
  - Multiplying factors
  - Other forms

RBFN learning

- This doesn't tell us how to *learn* in a RBFN

$$F(x) = \sum_{i=1}^{K} w_i \varphi(\|x - c_i\|) \qquad \varphi(\|x - c_i\|) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right)$$

- Remember: 3 set of parameters c_i, σ_i, w_i

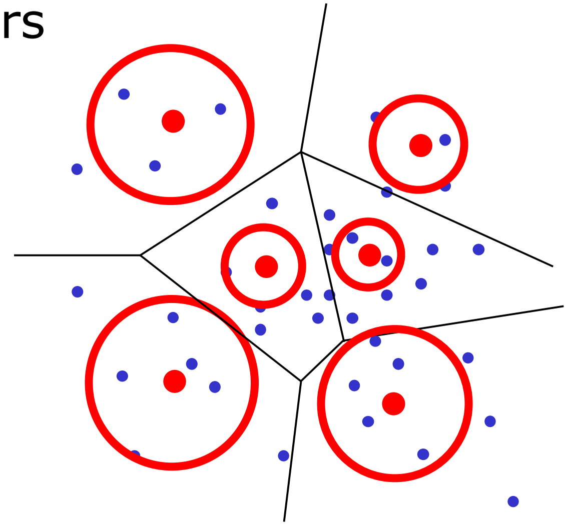
- Traditional learning strategy: splitted computation
  1. centers $\boldsymbol{c}_i$
  2. widths σ_i
  3. weights w_i

RBFN step 1: centers

- Idea: density of centers c_i must follow the density of learning points x^k
  → vector quantization
  - selected at random (in learning set)
  - competitive learning
  - frequency-sensitive learning
  - Kohonen maps
  - …
- This phase only uses the x^k information, not the t^k (it is *unsupervised*)

RBFN step 2: widths

- Universal approximation property holds with identical widths
- In practice (limited learning set): variable widths σ_i
- Idea: RBFN use *local* clusters



- choose σ_i *according* to standard deviation of clusters
- *According* means high stdv of clusters, high radius, but *doens't* mean equality (see further for details and ideas)

RBFN step 3: weights

$$F(x) = \sum_{i=1}^{K} w_i \varphi(\|x - c_i\|) \qquad \varphi(\|x - c_i\|) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right)$$

at x^p: constants !

- Problem becomes linear !
- Solution of least square criterion leads to

$$E(F) = \frac{1}{2P}\sum_{p=1}^{P}\left(t^p - F(x^p)\right)$$

$$w = \Phi^+ t = \left(\Phi^T \Phi\right)^{-1} \Phi^T t$$

where

$$\Phi \equiv \varphi_{ki} = \varphi\left(\left\|x^k - c_i\right\|\right)$$

- In practise: use SVD !

RBFN step 4: gradient descent

- After steps 1 to 3: all parameters are computed
  - But only the weights *w* are optimized with respect to the error criterion
  - Centers and widths are *reasonable choices*

$$F(x) = \sum_{i=1}^{K} \underbrace{w_i}_{\textbf{3}\text{ supervised}} \exp\left(- \frac{\|x - \overbrace{c_i}^{\textbf{1}\text{ unsupervised}}\|^2}{2\underbrace{\sigma_i}_{\textbf{2}\text{ unsupervised}}^2} \right)$$

- Optional step: possibility of gradient descent on *all* parameters

- Some improvement, but
  - learning speed
  - local minima
  - risk of non-local basis functions
  - etc.

More elaborated models

- Add constant and linear terms

$$F(x) = \sum_{i=1}^{K} w_i \exp\left(-\frac{\left\| x - c_i \right\|^2}{2\sigma_i^2} \right) + \sum_{i=1}^{D} w'_i x_i + w'_0$$

good idea (very difficult to approximate a constant with kernels…)

- Use normalized RBFN

$$F(x) = \sum_{i=1}^{K} w_i \frac{\exp\left(-\frac{\left\| x - c_i \right\|^2}{2\sigma_i^2} \right)}{\sum_{j=1}^{K} \exp\left(-\frac{\left\| x - c_j \right\|^2}{2\sigma_j^2} \right)}$$

basis functions are bouded [0,1] → can be interpreted as probability values (classification)

Back to the widths…

- choose σ_i according to standard deviation of clusters
- In the literature:
  - $\sigma = d_{\max} / \sqrt{2K}$ where $d_{\max}$ = maximum distance between centroids [1]
  - $\sigma_i = \frac{1}{q} \sqrt{\sum_{j=1}^{q} \left\| c_i - c_j \right\|^2}$ where index j scans the q nearest centroids to c_i [2]
  - $\sigma_i = r \min_j \left(\left\| c_i - c_j \right\| \right)$ where r is an overlap constant [3]
  - …..

[1] S. Haykin, *"Neural Networks a Comprehensive Foundation"*, Prentice-Hall Inc, second edition, 1999.

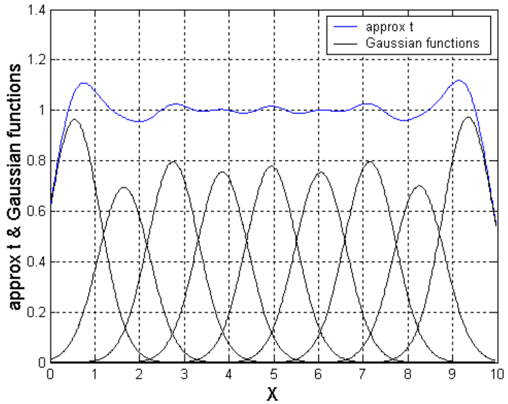
[2] J. Moody and C. J. Darken, *"Fast learning in networks of locally-tuned processing units"*, Neural Computation 1, pp. 281-294, 1989.

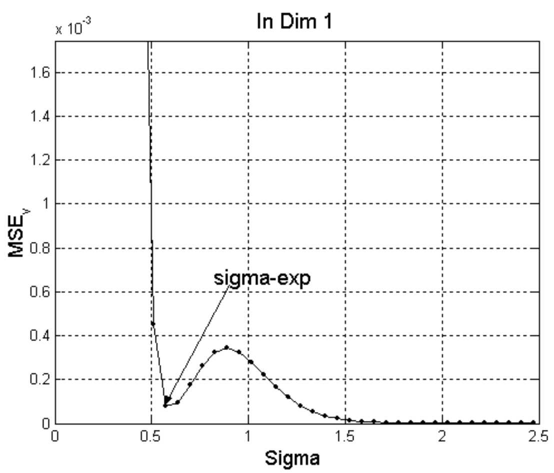
[3] A. Saha and J. D. Keeler, *''Algorithms for Better Representation and Faster Learning in Radial Basis Function Networks"*, Advances in Neural Information Processing Systems 2, Edited by David S. Touretzky, pp. 482-489, 1989.

Basic example

- Approximation of $f(\boldsymbol{x}) = 1$ with a d-dimensional RBFN

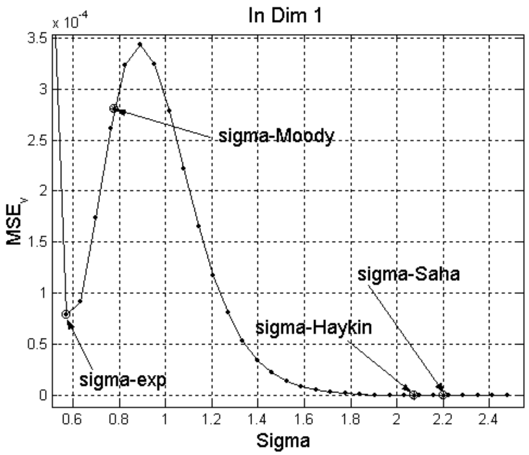
- In theory: identical w_i
- Experimentally: side effects
  → only middle taken into account

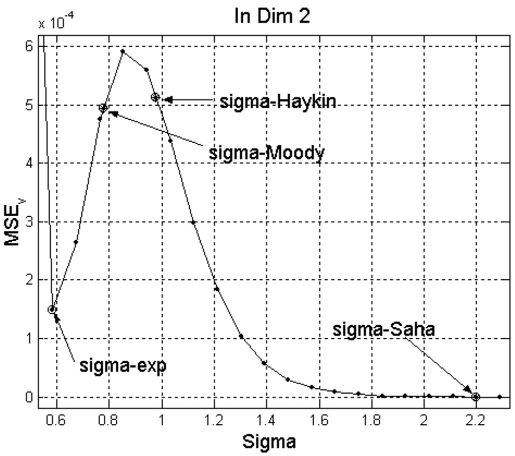


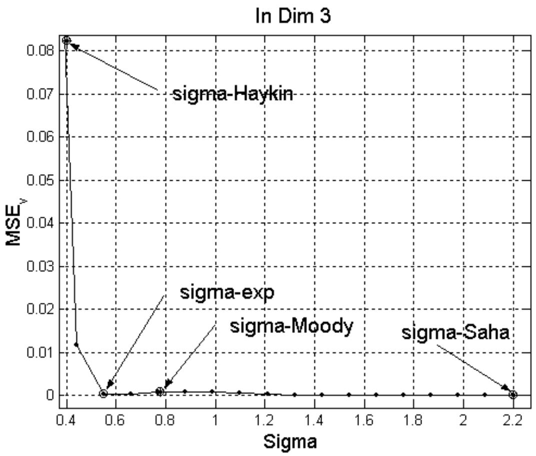


Error versus width

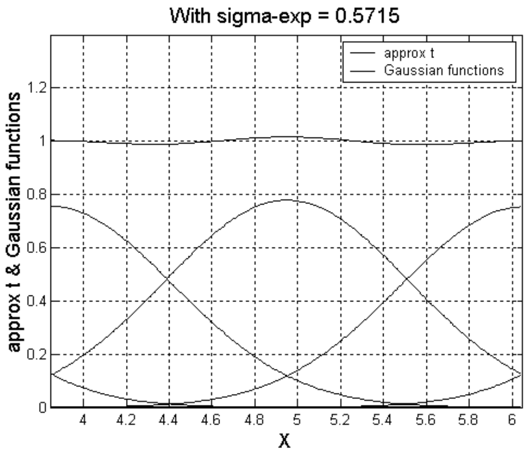
Basic example: error versus space dimension

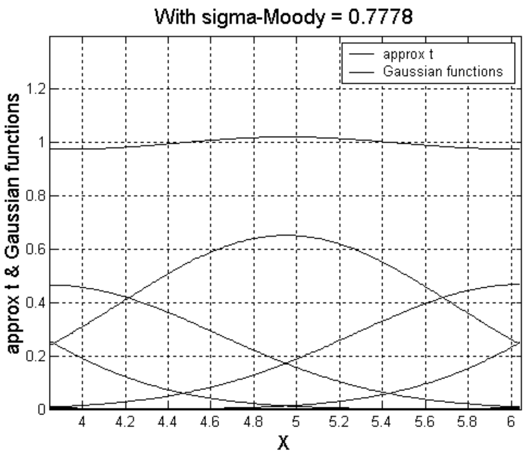


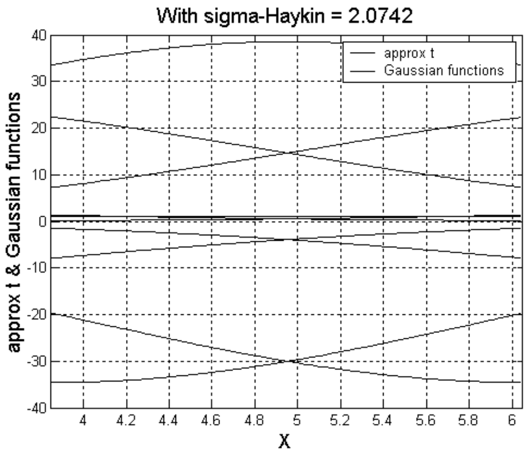




Basic example: local decomposition?

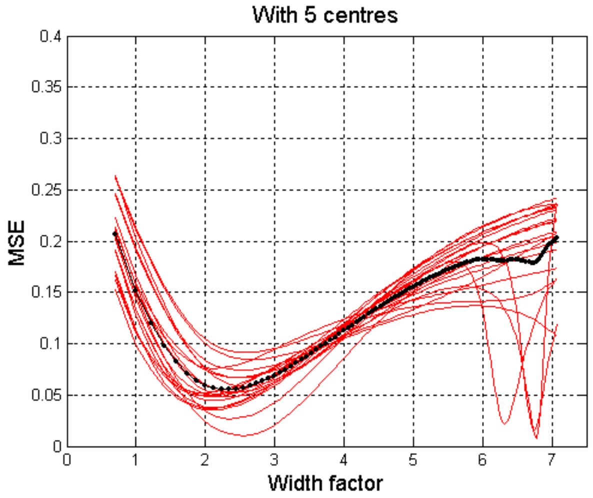






Multiple local minima in error curve

- Choose the first minimum to preserve the locality of clusters



- The first local minimum is usually less sensitive to variability (sparsity is good!)

Some concluding comments

- RBFN: easy learning (compared to MLP)
  - in a cross-validation scheme: important!
- Many RBFN models
- Even more RBFN learning schemes…
- Results not very sensitive to unsupervised part of learning (c_i, σ_i)
- Open work for a priori (proble-dependent) choice of widths σ_i

Sources and references

- Most of the basic concepts developed in these slides come from the excellent book:
  - Neural networks – a comprehensive foundation, S. Haykin, Macmillan College Publishing Company, 1994.
- Some supplementary comments come from the tutorial on RBF:
  - An overview of Radial Basis Function Networks, J. Ghosh & A. Nag, in: Radial Basis Function Networks 2, R.J. Howlett & L.C. Jain eds., Physica-Verlag, 2001.