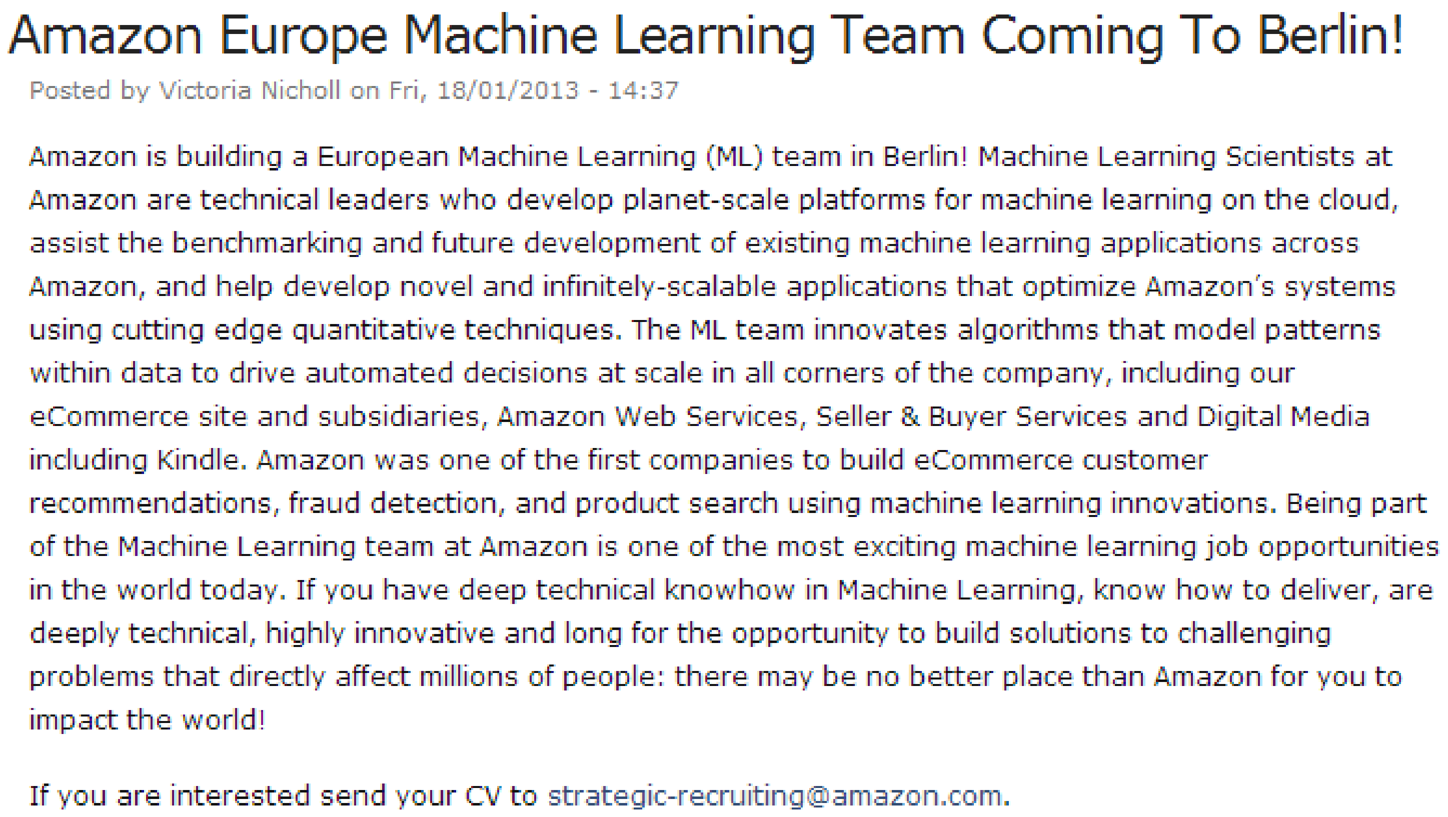
Basis Functions

Volker Tresp
Summer 2014

I am an AI optimist. We've got a lot of work in machine learning, which is sort of the polite term for AI nowadays because it got so broad that it's not that well defined.

`Bill Gates (Scientific American Interview, 2004)`

"If you invent a breakthrough in artificial intelligence, so machines can learn," Mr. Gates responded, "that is worth 10 Microsofts." (Quoted in NY Times, Monday March 3, 2004)



Amazon Europe Machine Learning Team Coming To Berlin!

Posted by Victoria Nicholl on Fri, 18/01/2013 - 14:37

Amazon is building a European Machine Learning (ML) team in Berlin! Machine Learning Scientists at Amazon are technical leaders who develop planet-scale platforms for machine learning on the cloud, assist the benchmarking and future development of existing machine learning applications across Amazon, and help develop novel and infinitely-scalable applications that optimize Amazon's systems using cutting edge quantitative techniques. The ML team innovates algorithms that model patterns within data to drive automated decisions at scale in all corners of the company, including our eCommerce site and subsidiaries, Amazon Web Services, Seller & Buyer Services and Digital Media including Kindle. Amazon was one of the first companies to build eCommerce customer recommendations, fraud detection, and product search using machine learning innovations. Being part of the Machine Learning team at Amazon is one of the most exciting machine learning job opportunities in the world today. If you have deep technical knowhow in Machine Learning, know how to deliver, are deeply technical, highly innovative and long for the opportunity to build solutions to challenging problems that directly affect millions of people: there may be no better place than Amazon for you to impact the world!

If you are interested send your CV to strategic-recruiting@amazon.com.

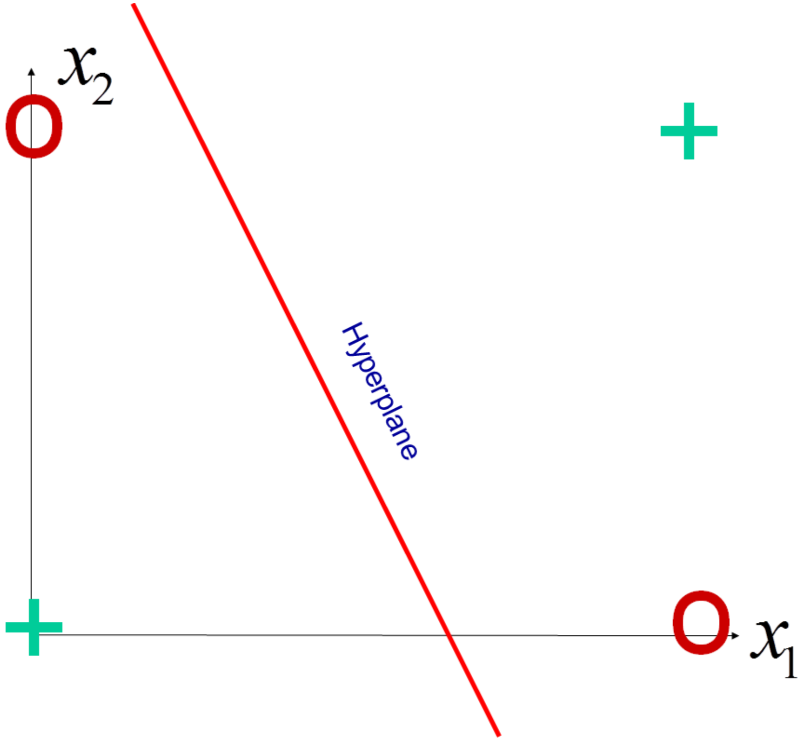
Nonlinear Mappings and Nonlinear Classifiers

- Regression:
  - Linearity is often a good assumption when many inputs influence the output
  - Some natural laws are (approximately) linear $F = ma$
  - But in general, it is rather unlikely that a true function is linear
- Classification:
  - Similarly, it is often not reasonable to assume that the classification boundaries are linear hyper planes

Trick

- We simply transform the input into a high-dimensional space where the regression/classification is again linear!
- Other view: let's define appropriate features
- Other view: let's define appropriate basis functions

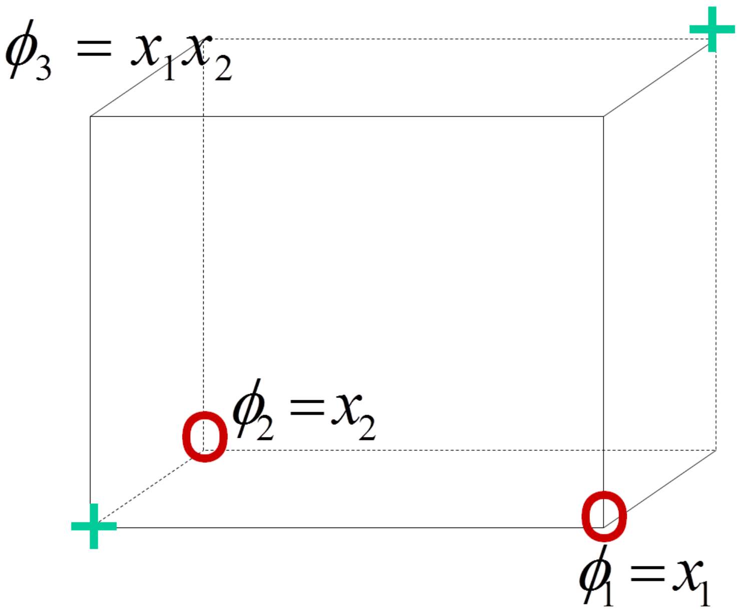
XOR is not linearly separable



Trick: Let's add Basis Functions

- Linear Model: input vector: $1, x_1, x_2$
- Let's consider $x_1 x_2$ in addition
- The interaction term $x_1 x_2$ couples two inputs nonlinearly

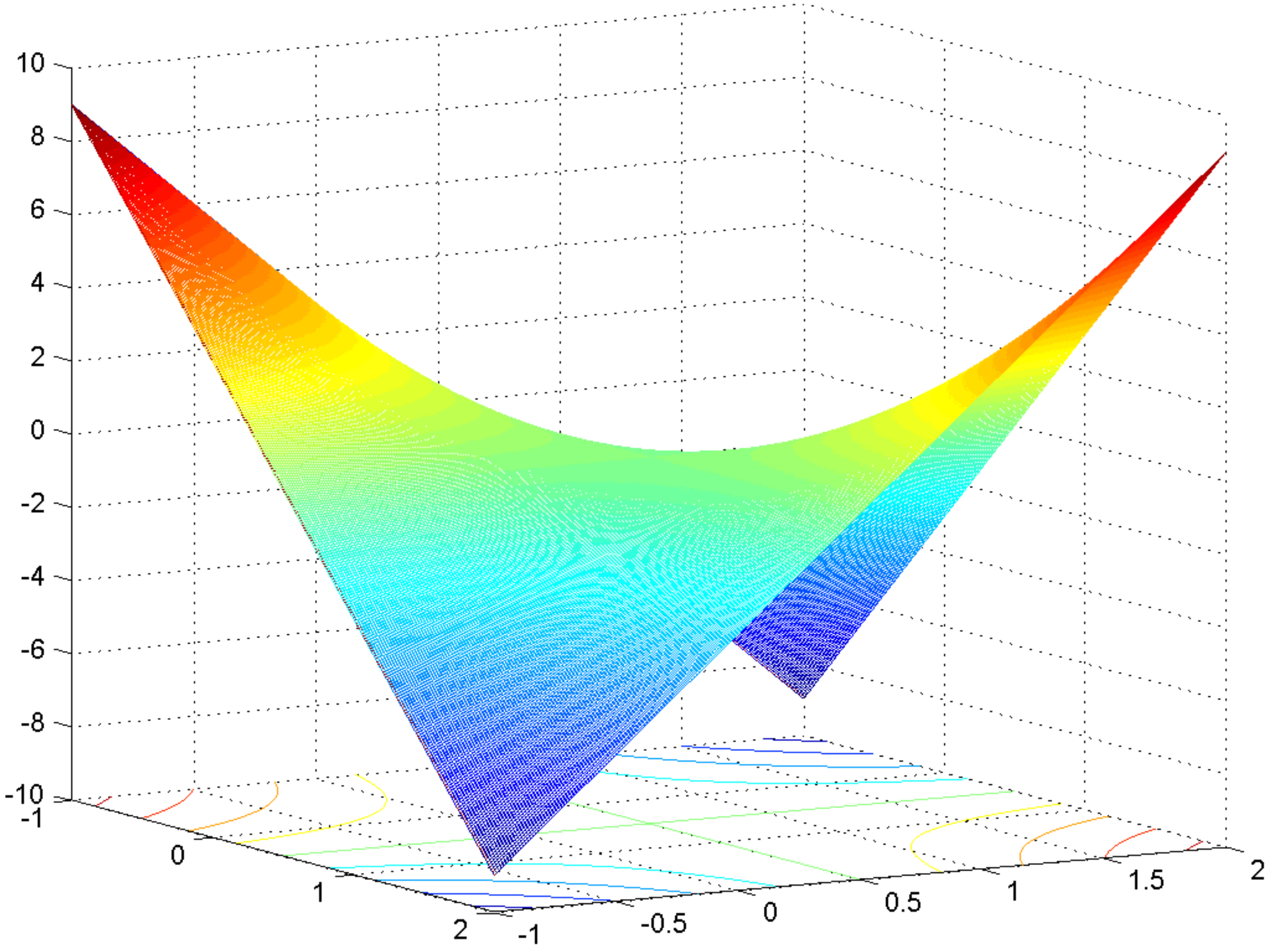
With a Third Input $z_3 = x_1x_2$ the XOR Becomes Linearly Separable



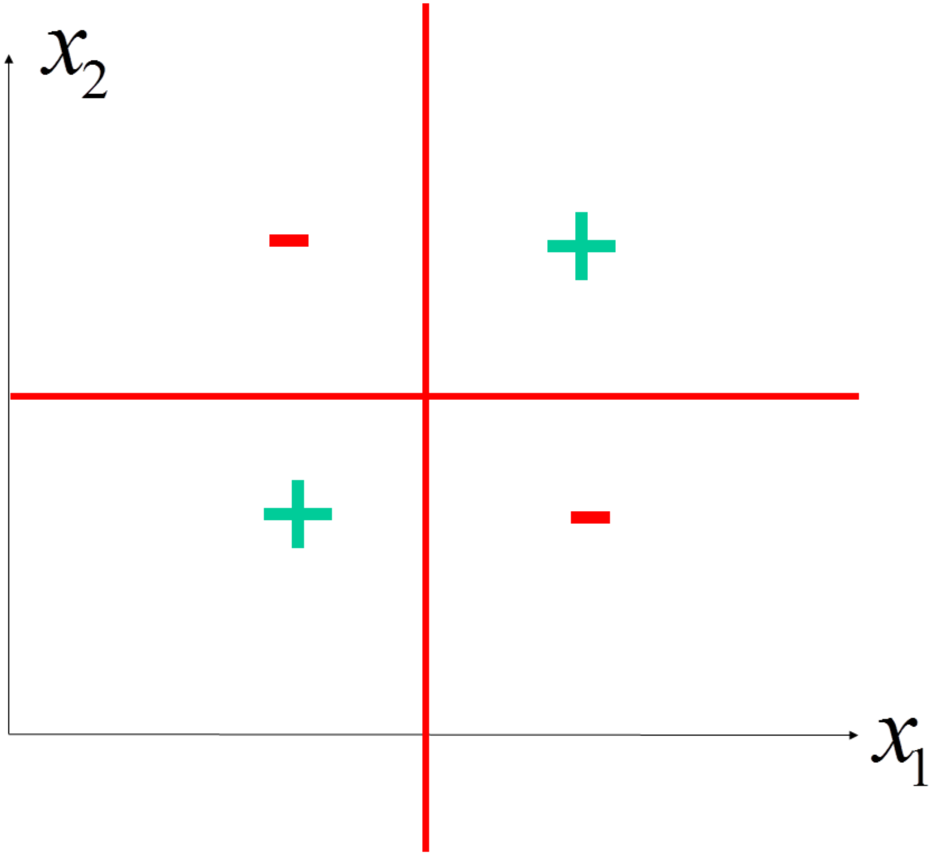
$$f(\mathbf{x}) = 1 - 2x_1 - 2x_2 + 4x_1x_2 = \phi_1(x) - 2\phi_2(x) - 2\phi_3(x) + 4\phi_4(x)$$

with $\phi_1(x) = 1, \phi_2(x) = x_1, \phi_3(x) = x_2, \phi_4(x) = x_1x_2$

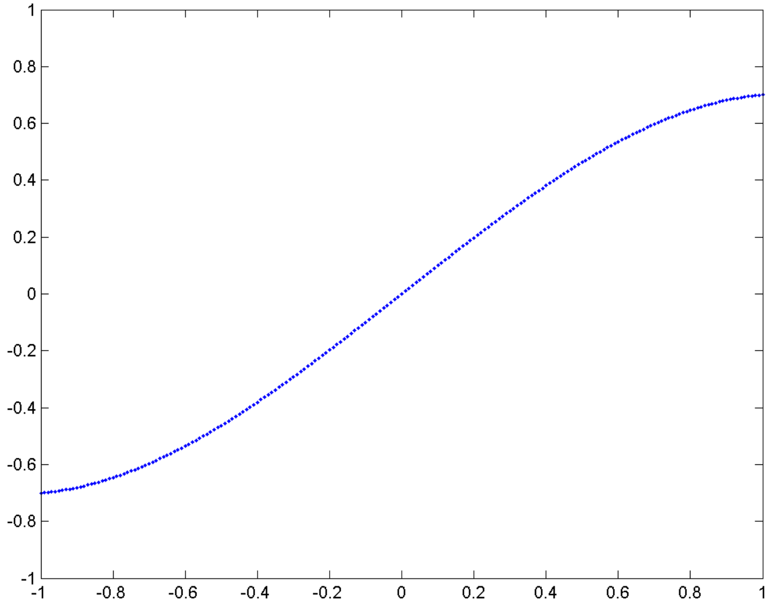
$$f(\mathbf{x}) = 1 - 2x_1 - 2x_2 + 4x_1x_2$$



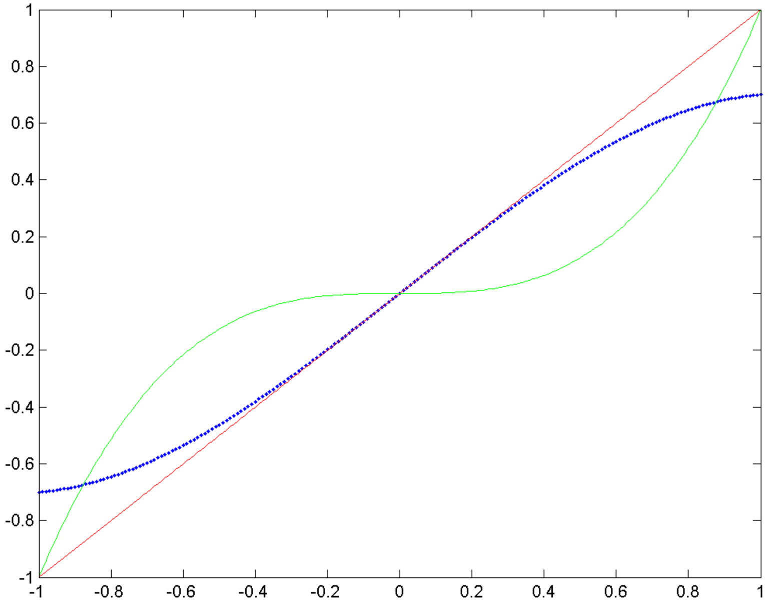
Separating Planes



A Nonlinear Function



$$f(x) = x - 0.3x^3$$



Basis functions $\phi_1(x) = 1, \phi_2(x) = x, \phi_3(x) = x^2, \phi_4(x) = x^3$ und $\mathbf{w} = (0, 1, 0, -0.3)$

Basic Idea

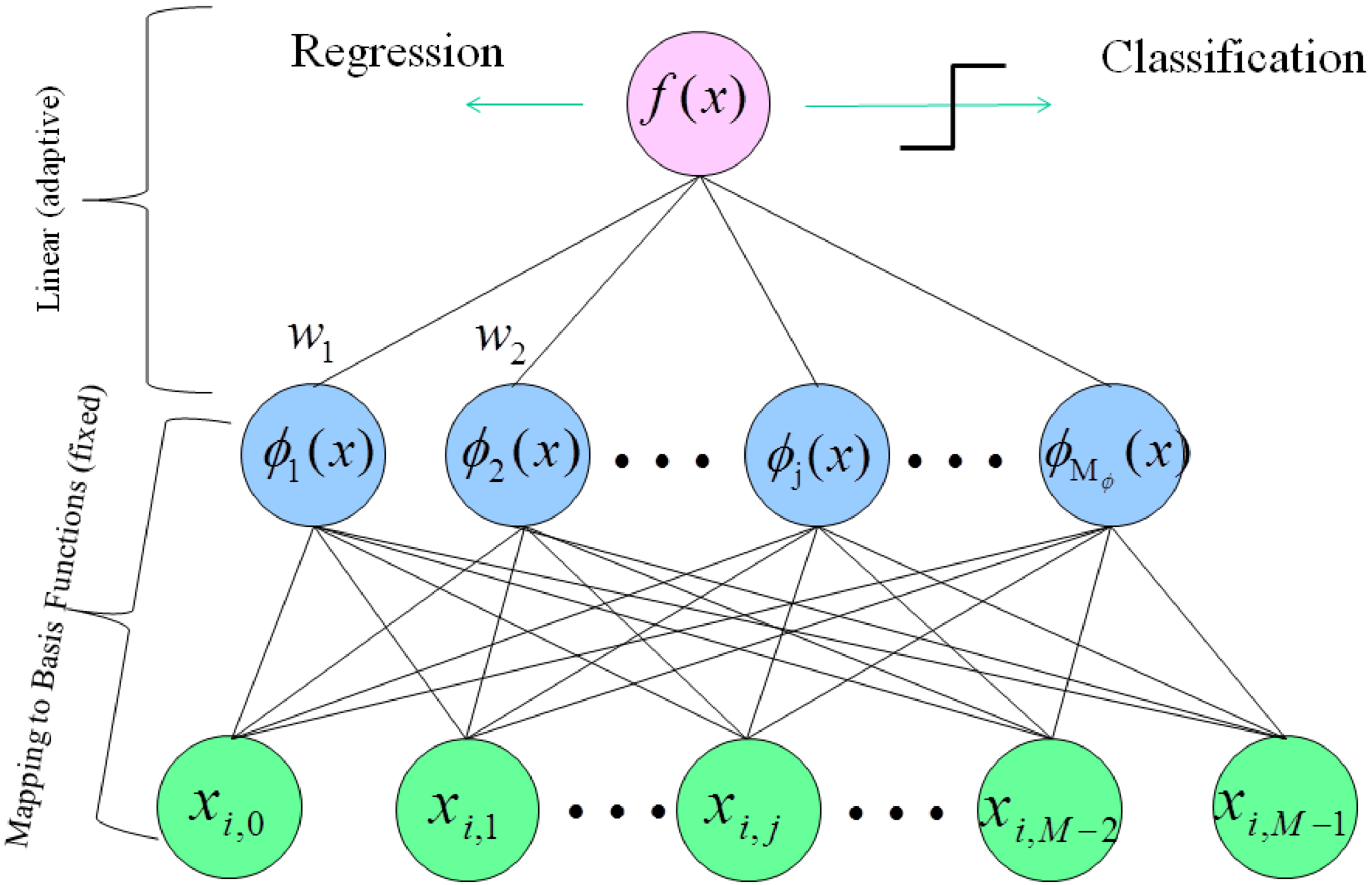
- The simple idea: in addition to the original inputs, we add inputs that are calculated as deterministic functions of the existing inputs and treat them as additional inputs
- Example: Polynomial Basis Functions

$$\{1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1^2, x_2^2, x_3^2\}$$

- Basis functions $\{\phi_h(\mathbf{x})\}_{h=1}^{M_\phi}$
- In the example:

$$\phi_1(\mathbf{x}) = 1 \quad \phi_2(\mathbf{x}) = x_1 \quad \phi_6(\mathbf{x}) = x_1x_3 \quad \ldots$$

- Independent of the choice of basis functions, the regression parameters are calculated using the well-known equations for linear regression


Linear (adaptive)
Mapping to Basis Functions (fixed)
Regression
$f(x)$
Classification
w_1
w_2
$\phi_1(x)$
$\phi_2(x)$
$\phi_j(x)$
$\phi_{M_\phi}(x)$
$x_{i,0}$
$x_{i,1}$
$x_{i,j}$
$x_{i,M-2}$
$x_{i,M-1}$

Review: Penalized LS for Linear Regression

- Multidimensional Linear Model:

$$f(\mathbf{x}_i, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_{i,j} = \mathbf{x}_i^T \mathbf{w}$$

- Regularized cost function

$$\text{cost}^{pen}(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + \lambda \sum_{i=0}^{M-1} w_i^2$$

- Die PLS-Solution

$$\hat{\mathbf{w}}_{pen} = \left(\mathbf{X}^T \mathbf{X} + \lambda I\right)^{-1} \mathbf{X}^T \mathbf{y} \quad \text{with} \quad \mathbf{X} = \begin{pmatrix} x_{1,0} & \ldots & x_{1,M-1} \\ \ldots & \ldots & \ldots \\ x_{N,0} & \ldots & x_{N,M-1} \end{pmatrix}$$

Regression with Basis Functions

- Model with basis functions:

$$f(\mathbf{x}_i, \mathbf{w}) = \sum_{j=1}^{M_\phi} w_j \phi_j(\mathbf{x}_i)$$

- Regularized cost function

$$J_N^{pen}(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + \lambda \sum_{i=1}^{M_{\mathbf{\Phi}}} w_i^2$$

- The PLS-solution

$$\hat{\mathbf{w}}_{pen} = \left(\mathbf{\Phi}^T \mathbf{\Phi} + \lambda I\right)^{-1} \mathbf{\Phi}^T \mathbf{y}$$

with

$$\mathbf{\Phi} = \left(\begin{array}{ccc} \phi_1(\mathbf{x}_1) & \dots & \phi_{M_\phi}(\mathbf{x}_1) \\ \dots & \dots & \dots \\ \phi_1(\mathbf{x}_N) & \dots & \phi_{M_\phi}(\mathbf{x}_N) \end{array} \right)$$

Nonlinear Models for Regression and Classification

- Regression:

$$f(\mathbf{x}) = \sum_{j=1}^{M_\phi} w_j \phi_i(\mathbf{x})$$

  As discussed, the weights can be calculated via PLS

- Classification:

$$\hat{y} = \text{sign}(f(\mathbf{x})) = \text{sign}\left(\sum_{j=1}^{M_\phi} w_j \phi_j(\mathbf{x})\right)$$

  The Perceptron learning rules can be applied, if we replace $1, x_{i,1}, x_{i,2}, \ldots$ with $\phi_1(x_i), \phi_2(x_i), \ldots$

Which Basis Functions?

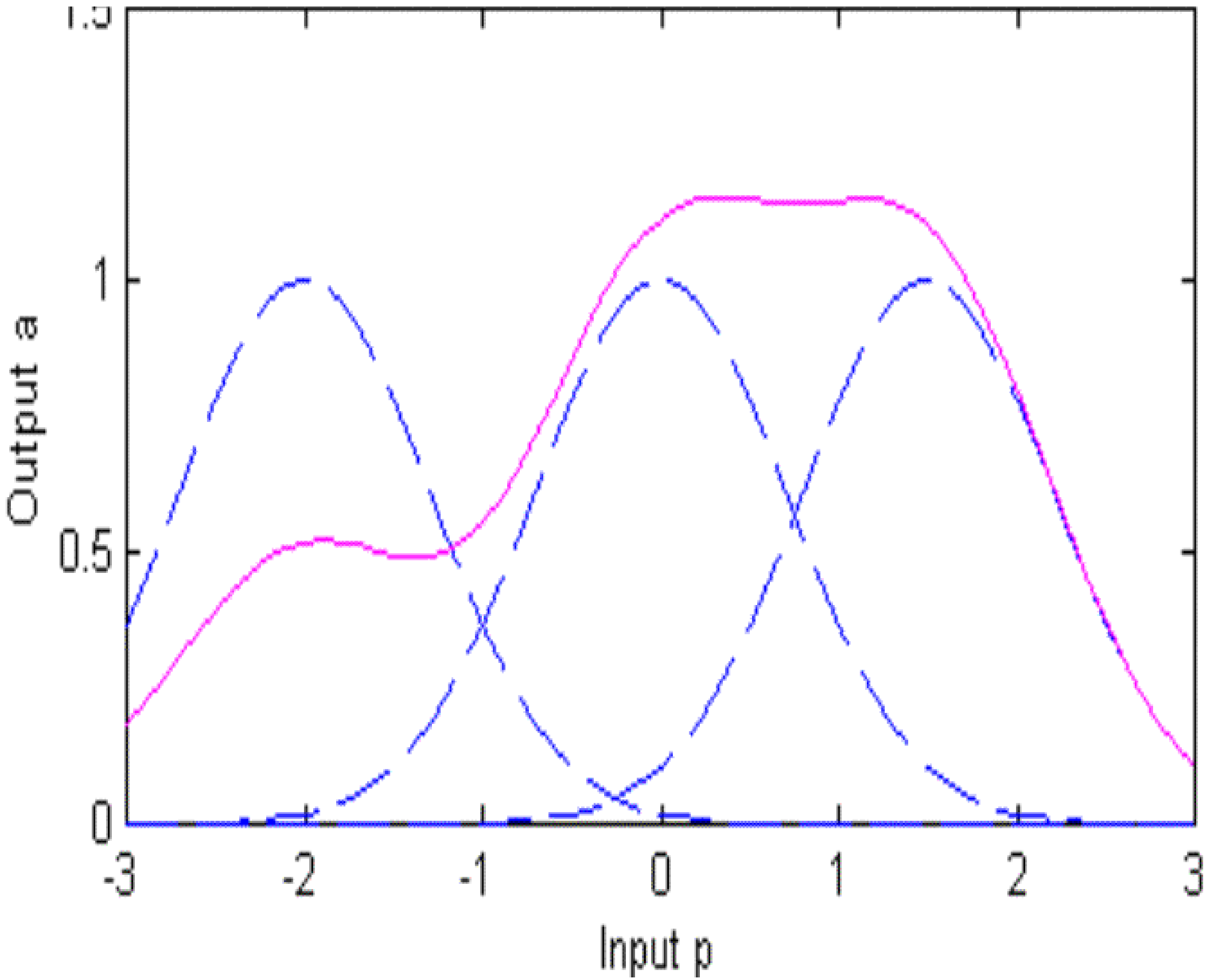
- The challenge is to find problem specific basis functions which are able to effectively model the true mapping

Radial Basis Function (RBF)

- We already have learned about polynomial basis functions
- Another class are radial basis functions (RBF). Typical representatives are Gaussian basis functions

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2s_j^2}|\mathbf{x} - \mathbf{c}_j|^2\right)$$

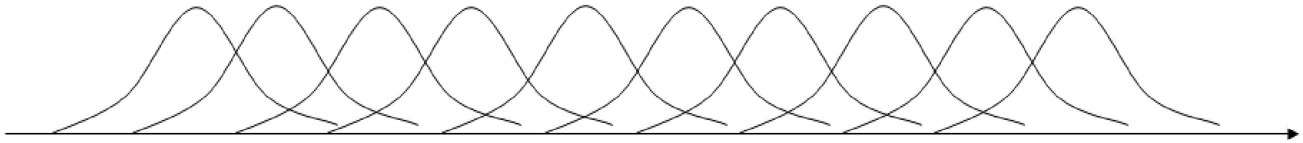
Three RBFs (blue) form $f(\mathrm{x})$ (pink)

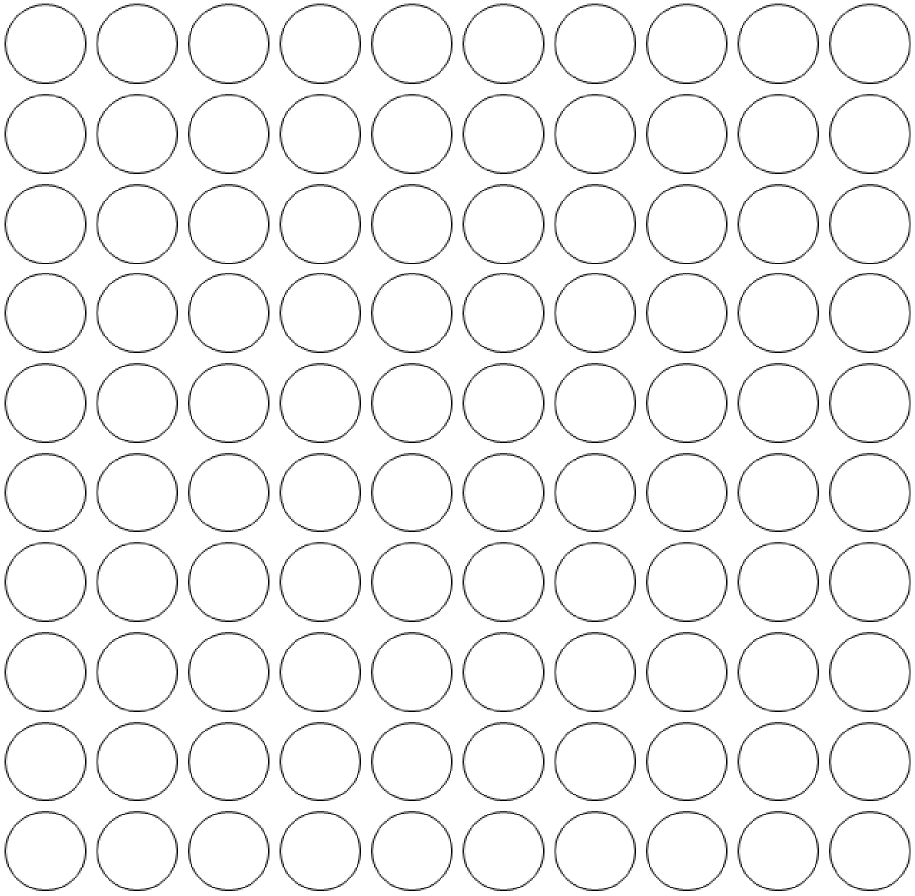


Optimal Basis Functions

- So far all seems to be too simple
- Here is the catch: the number of “sensible” basis functions increases exponential with the number of inputs
- If I am willing to use K basis functions per dimension. then I need K^M RBFs in M dimensions
- We get a similar exponential increase for polynomial basis functions
- *The most important challenge: How can I get a small number of relevant basis functions*

10 RBFs in one dimension





100 RBFs in two dimensions

Model Selection: Polynomial Basis functionen

- First we only work with the original inputs and forma linear model
- Then we sequentially stepwise add basis functions that improve the model significantly
- Alternative: we start with many polynomial basis functions and remove the ones whose removeal does not deteriorate performance significantly
- Polynomklassifikatoren: Siemens-Dematic OCR, J. Schürmann):
  - Pixel-based image features
  - Dimensional reduction via PCA
  - Additional basis functions (significant polynomials)
  - Linear Classification

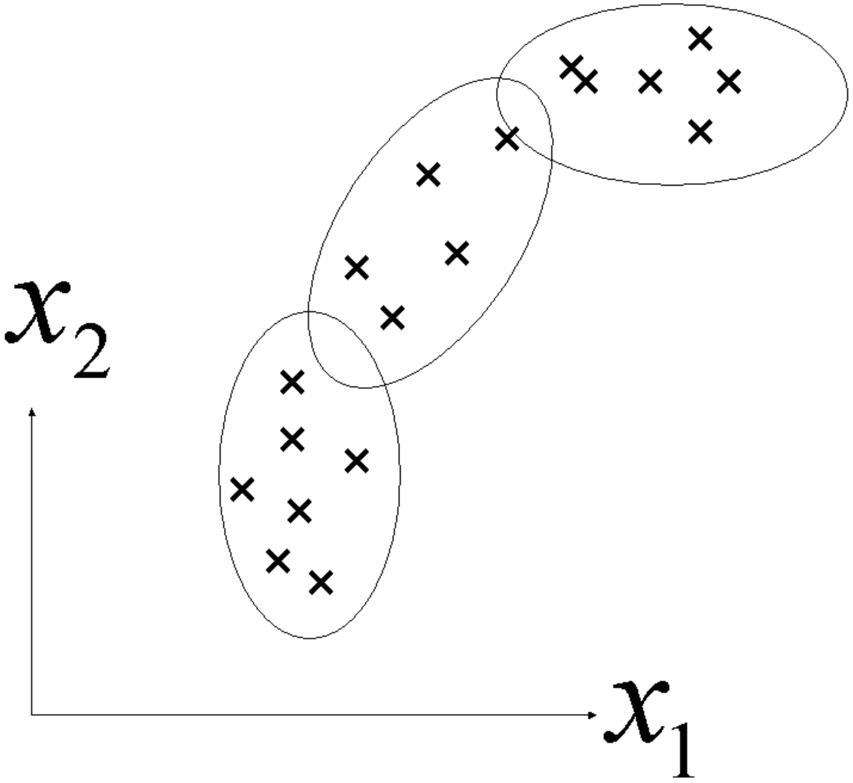
Model Selection: RBFs

- Sometimes it is sensible to first group (cluster) data in input space and to then use the cluster centers as positions for the Gaussian basis functions

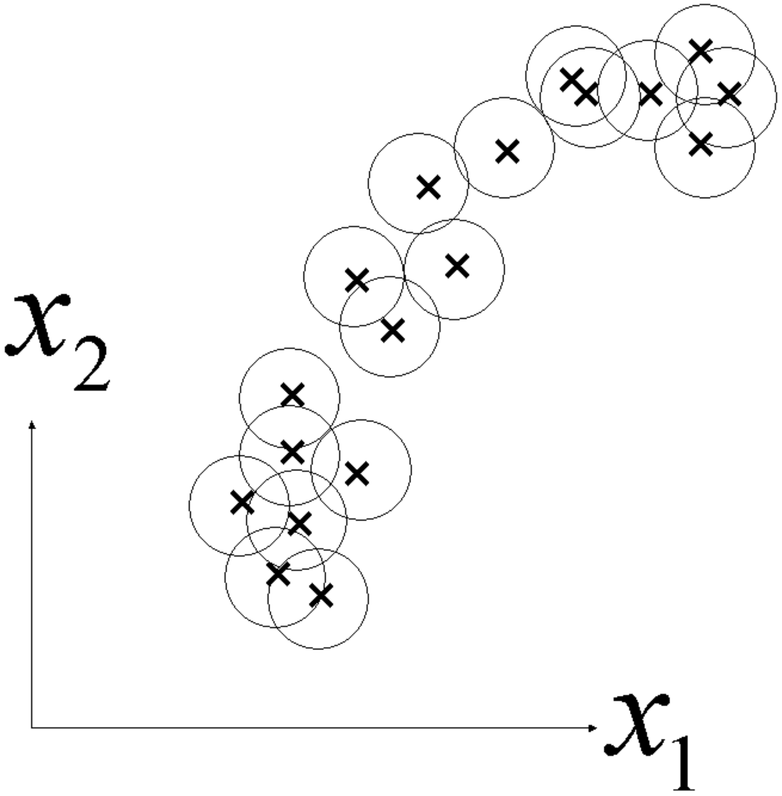
- The widths of the Gaussian basis functions might be derived from the variances of the data in the cluster

- An alternative is to use one RBF per data point. The centers of the RBFs are simply the data points themselves and the widths are determined via some heuristics (or via cross validation, see later lecture)

RBFs via Clustering



One Basis Function per Data Point



Application-Specific Features

- Often the basis functions can be derived from sensible application features
  - Given an image with $256 \times 256 = 65536$ pixels. The pixels form the input vector for a linear classifier. This representation would not work well for face recognition
  - With fewer than 100 appropriate features one can achieve very good results (example: PCA features, see later lecture)
- The definition of suitable features for documents, images, gene sequences, ... is a very active research area
- If the feature extraction already delivers many features, it is likely that a linear model will solve the problem and no additional basis functions need to be calculated
- This is quite remarkable: learning problems can become simpler in high-dimensions, in apparent contradiction to the famous "curse of dimensionality" (Bellman)

Appendix: Detour on Function Spaces

Vectors

- To describe a vector f we need basis vectors ϕ_i that define the orthogonal unit vectors in a coordinate system and the coordinates of a vector w_i, and $f = \sum_i w_i \phi_i$
- Orthogonality of basis vectors: $\left\langle \phi_i, \phi_j \right\rangle_{\mathbf{\Phi}} = \delta_{i,j}$
- The coordinates of a vector in a coordinate system are defined by the inner product of the vector with the basis vectors $w_i = \langle \phi_i, f \rangle_{\mathbf{\Phi}}$
- The inner product of two vectors is then $\langle f, g \rangle_{\mathbf{\Phi}} = \sum_i w_{f,i} w_{g,i}$
- To move from one coordinate system to a reference coordinate system we need the coordinates of the basis vectors in the reference coordinate system

Functions are Vectors

- Functions are just like vectors in a vector space $f = \sum_i w_i \phi_i$
- The reference system is defined by delta functions $\delta(x - x')$. The coordinates are simply the functional values: $\langle \delta_x, f \rangle_\delta = w_x = f(x)$
- In this coordinate system, $\langle f, g \rangle_\delta = \int f(x) g(x) dx = \sum_{i,j} w_{f,i} w_{g,j} \left\langle \phi_i, \phi_j \right\rangle_\delta$
- The representation of another basis vector ϕ_i in the reference coordinate system is $\langle \delta_x, \phi_i \rangle_\delta = \phi_i(x)$. Thus $f(x) = \sum_i w_i \phi_i(x)$
- Similarly, we have $w_i = \langle \phi_i, f \rangle_\Phi$, and $\langle f, g \rangle_\Phi = \sum_i w_{f,i} w_{g,i}$
- Note, that in general: $\langle f, g \rangle_\delta \neq \langle f, g \rangle_\Phi$

Rewriting the Cost Function

- Also note that

$$f(x) = \sum_i w_i \phi_i(x)$$

can be thought of as an inner product between the function $f(x') = \sum w_i \phi_i(x')$ and the function $k(x, x') = \sum \phi_i(x)\phi_i(x')$, thus

$$f(x) = \langle f, k_x \rangle_{\Phi}$$

- Here, $k(x, x')$ is a kernel function and is called the *reproducing kernel*

- With all of this, we can write our cost function as

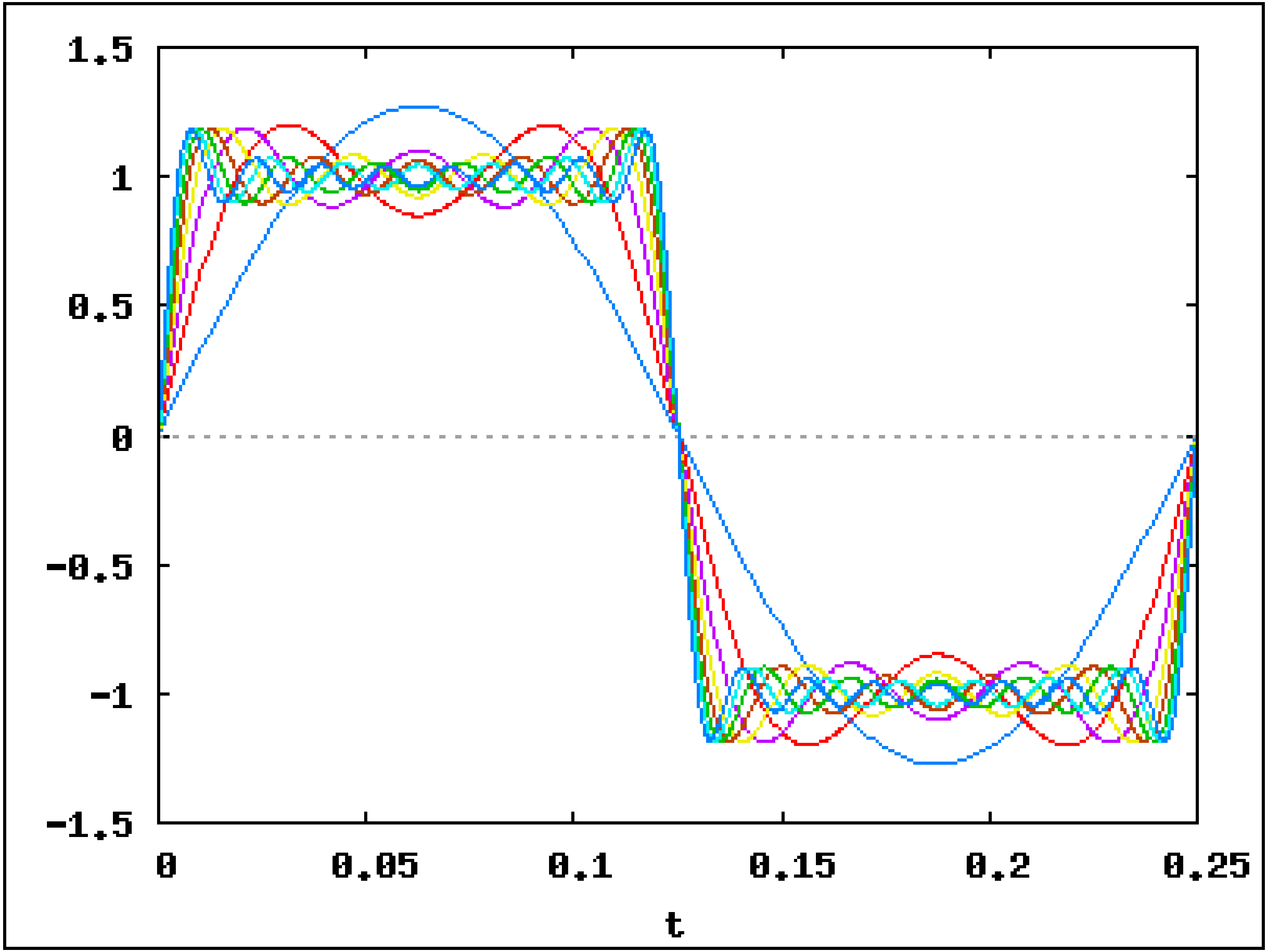
$$\text{cost}^{pen} = \sum_{i=1}^{N} \left(y_i - \langle f, k_{x_i} \rangle_{\Phi}\right)^2 + \lambda \langle f, f \rangle_{\Phi}$$

Fourier Basis Functions

- A common set of basis functions (in 1-D or 2-D) are Fourier basis functions $\phi_{c,\omega_i} = \cos(\omega_i x)$, $\phi_{s,\omega_i} = \sin(\omega_i x)$
- They are orthogonal in the basis function space, but also in the reference space

$$\left\langle \phi_{\omega_i}, \phi_{\omega_j} \right\rangle_{\Phi} = \left\langle \phi_{\omega_i}, \phi_{\omega_j} \right\rangle_{\delta} = \delta_{i,j}$$

- Thus we can write $f(x) = \sum_i w_{c,i} \cos(\omega_i x) + w_{s,i} \sin(\omega_i x)$ and the $w_{c,i}$ and the $w_{s,i}$ form the spectrum


1.5
1
0.5
0
-0.5
-1
-1.5
0
0.05
0.1
0.15
0.2
0.25
t

An Interesting Connection to Quantum Mechanics

- The state is described by a (complex valued) wave function ψ
- In the reference system, the basis function for location are $\delta(x - x')$ and the weight is called $w_x(x) = \psi(x)$
- The basis function for momentum p is ϕ_p and its representation in location space is ($\hbar = h/(2\pi)$ where h is the Planck constant, $i = \sqrt{-1}$)

$$\psi_p(x) = \frac{1}{\sqrt{2\pi\hbar}} \exp(ipx/\hbar)$$

- Given, ψ, the probability that the particle is measured in location x is

$$|w_x|^2 = |\psi(x)|^2$$

- Given, ψ, the probability that the particle is measured with momentum p is

$$|w_p|^2$$

Collapse of the Wave function

- What if I do another measurement, would I get the same probabilities? The answer is no! After I do a measurement on the particle, ψ become identical to the basis function associated with the measurement (collapse of the wave function)
- Thus if I measure the particle at location x, the wave function changes to
$$\psi(x) = \delta_x$$
- Thus if I measure the particle with momentum p, the wave function changes to ψ_p with
$$\psi_p(x) = \frac{1}{\sqrt{2\pi\hbar}} \exp(ipx/\hbar)$$
- This collapse of the wave function is still a big riddle and has let to different interpretations of the quantum theory (Copenhagen, Many-world, ...)

Uncertainty Principle

- Note that w_x and w_p are Fourier transforms of one another (The complex exponential is a convenient way of writing cosine and sine)
- This means that if I measure location, then momentum is flat (all p have same probability) and if I measure momentum, then location is flat (all x have same probability)
- This is the uncertainty principal: I cannot measure location and momentum of a particle at the same time!
- If I make location x more blurred, I can get a more focussed p, but

$$\sigma_x \sigma_p \geq \frac{\hbar}{2}$$