Special Course in Computer and Information Science II

Machine Learning : Basic Principles

KERNEL METHODS

Jayaprakash Rajasekharan

19-02-2007

Kernels

- Input Output sets X , Y
- Training set $(x_{1,} y_1), (x_{2,} y_2), \ldots\ldots, (x_n , y_n) \in X \times Y$
- Generalization : given unseen $x \in X$, find $y \in Y$
- (x , y) should be “similar” to $(x_{1,} y_1), \ldots\ldots, (x_n , y_n)$
- How to measure similarity..??
  - Outputs - Loss Function
  - Inputs - “Kernels”

Similarity of Inputs

- Symmetric Function $k : X \times X \to R$

$$(x, x') \to k(x, x')$$

- If $X = R^N$, Dot product gives a linear kernel

$$k(x, x') = x^T x'$$

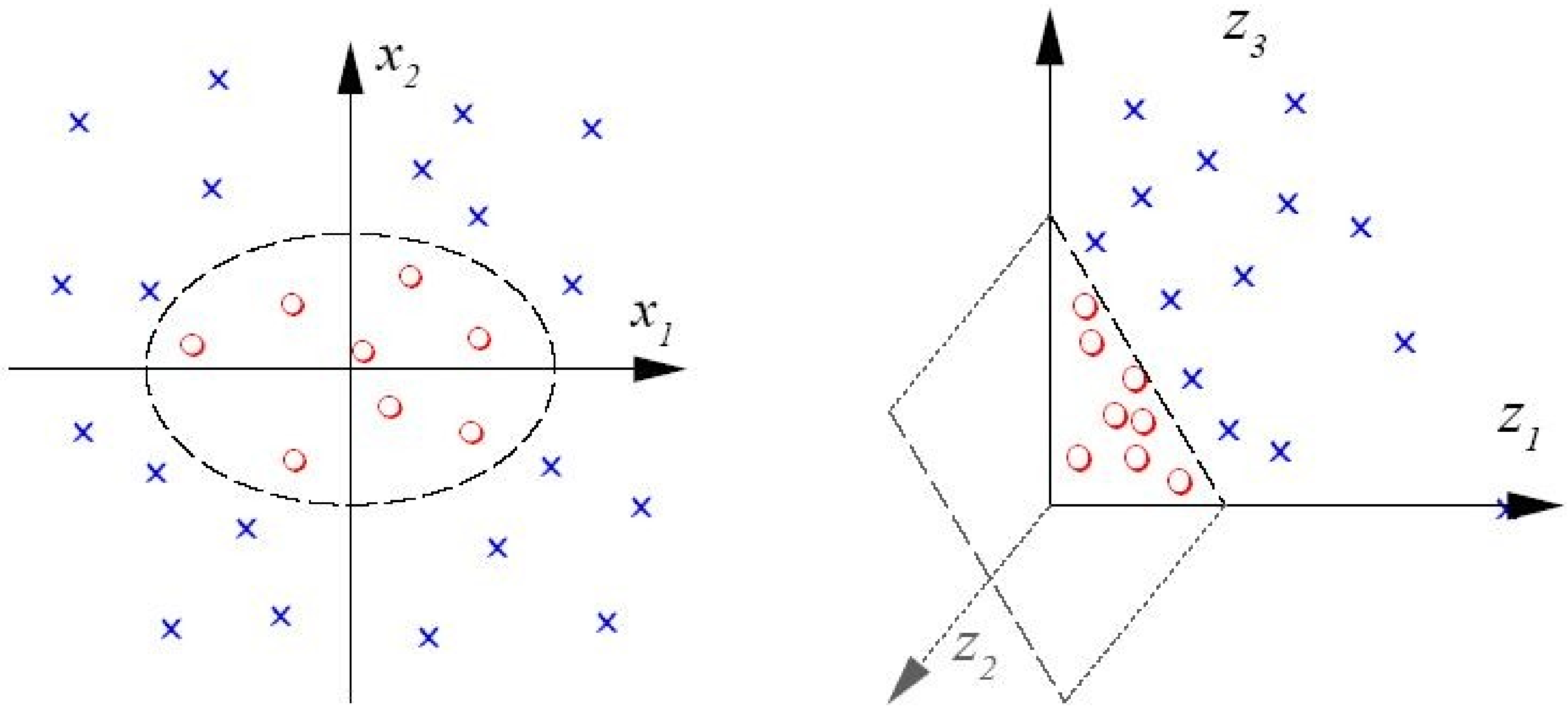
- For instance, in R^2 we can collect monomial features extractors of degree 2 in a nonlinear map

$$\Phi : R^2 \to R^3$$

$$(x, x') \to (x^2, x'^2, xx')$$

Kernel Algorithm

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$



Kernel Methods

- For a N dimensional input, we have N_F monomials

$$N_F = \frac{(N+d-1)!}{d!(N-1)!}$$

[For a 16x16 pixel input image and monomial degree d = 5 → $N_F = 10^{10}$]

- If X is not a dot product space, then there exists a map $\Phi : X \rightarrow H$ such that $k(x, x') = \phi(x)^T \phi(x')$ where H is a linearized feature space.
- Compute dot products in high dimensional feature spaces without explicitly mapping into them by means of kernels that are nonlinear in input space.

Kernel Methods

- Kernel Trick → If the input vector enters any algorithm only in the form of scalar products, then replace the scalar product by some appropriate kernel.
  - Kernel PCA
  - Kernel Fisher Discriminant
- Linear Parametric Model → Dual Representations
- Can also be applied to symbolic inputs such as strings, sets, graphs, text documents etc..

Dual Representations

- Linear Regression Model → Minimize the regularized sum of squares error function

$$J(w)=\frac{1}{2}\sum_{n=1}^{N}\{w^T\phi(x_n)-t_n\}^2+\frac{\lambda}{2}w^Tw$$

- Formulate Gram Matrix $K=\Phi\Phi^T$

$N\times N$ symmetric matrix $K_{nm}=\phi(x_n)^T\phi(x_m)=k(x_n,x_m)$

- Prediction for new input x

$$y(x)=w^T\phi(x)=a^T\Phi\phi(x)=k(x)^T(K+\lambda I_N)^{-1}\boldsymbol{t}$$

Types of Kernels

- Stationary Kernels $k(x, x') = k(x - x')$
- Homogeneous Kernels - Radial Basis Functions

$$k(x, x') = k(\|x - x'\|)$$

- Positive Definite Kernels - For Symmetric kernels

$k(x, x') = k(x', x)$, the Gram Matrix **K** with

elements $K_{nm} = k(x_n, x_m)$ is positive definite, i.e.,

$a_n^T K a_m > 0$

Examples of Kernels

- Simple Polynomial Kernel – terms of degree 2

$$k(x, x') = (x^T x)^2$$

- Generalized Polynomial kernel – degree M

$$k(x, x') = (x^T x + c)^M, \ c > 0$$

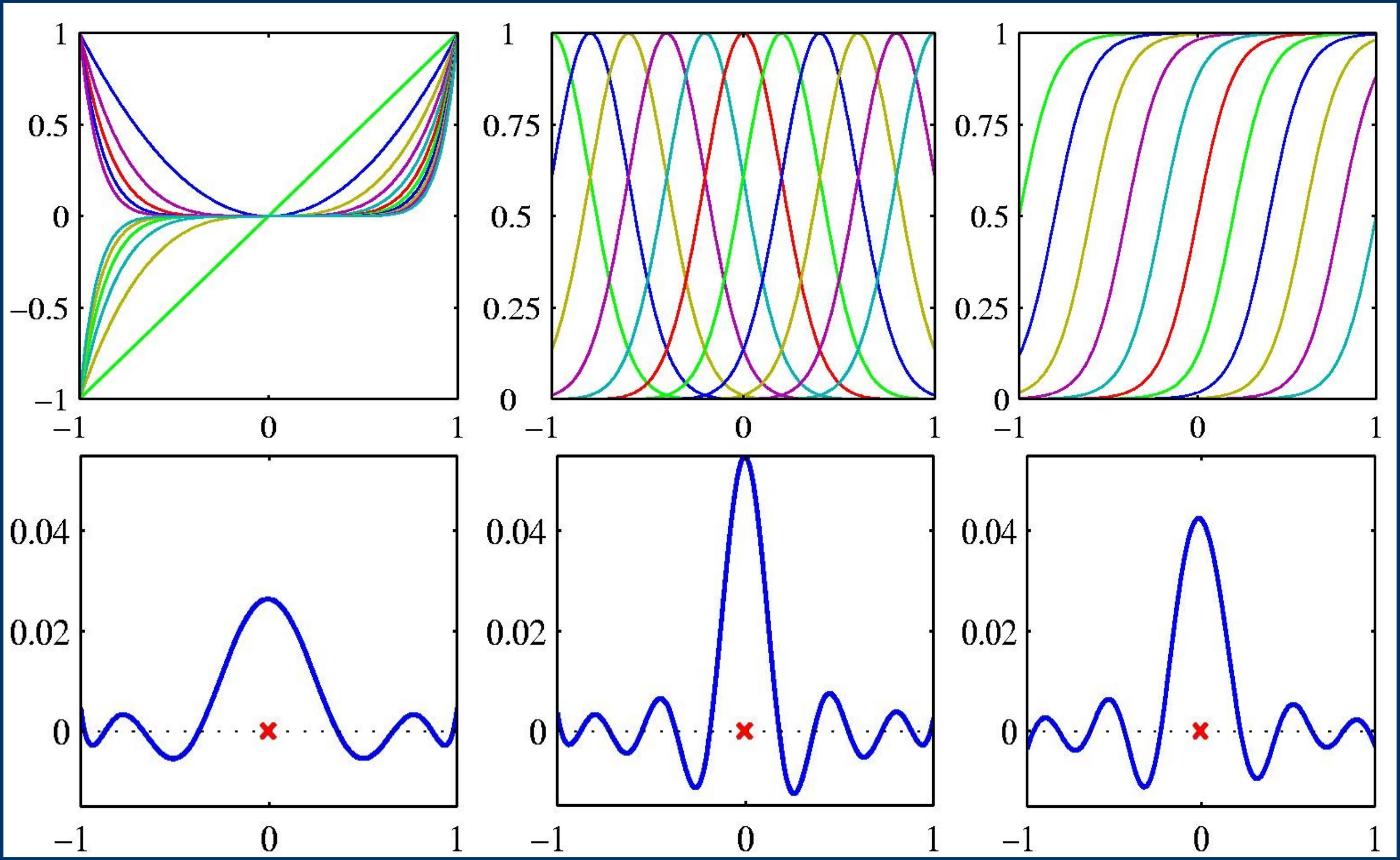
- Gaussian Kernels – not related to gaussian pdf !

$$k(x, x') = \exp(-\|x - x'\|^2 / 2\sigma^2)$$

- Sigmoidal Kernels – Gram Matrix is not p.d

$$k(x, x') = \tanh(a\, x^T x + b)$$

Examples of Kernels



Construction of Kernels

- Given valid kernels $k_1(x, x')$ and $k_2(x, x')$, the following are also valid kernels

$$k_2(x, x') = c\,k_1(x, x'),\ c > 0$$

$$k_2(x, x') = f(x)k_1(x, x')f(x')$$

$$k_2(x, x') = \exp(k_1(x, x'))$$

$$k_2(x, x') = k_1(x, x') + k_2(x, x')$$

$$k_2(x, x') = k_1(x, x')k_2(x, x')$$

Construction of Kernels

- Kernels from probabilistic generative model
  - Can be used in discriminative setting
- Given a generative model $p(x)$, define a kernel by

$$k(x, x') = p(x)\, p(x')$$

  - Can be interpreted as inner product in the one dimensional feature space defined by mapping $p(x)$
  - Two inputs x and x' are similar if they both have high probabilities

Construction of Kernels

- We can further extend this class of kernels by considering sums over products of different probability distributions with positive weighting coefficients

$$k(x, x') = \sum_i p(x/i)\, p(x'/i)\, p(i)$$

- For continuous latent variables, we have

$$k(x, x') = \int p(x/z)\, p(x'/z)\, p(z)\, dz$$

Construction of Kernels

- Consider parametric generative model $p(x/\theta)$
- Find a kernel that measures similarity of two input vectors induced by the generative model.
- Consider the gradient w.r.t parameter θ that defines a vector in feature space having the same dimensionality as the parameter vector θ.
- Fisher Score $g(\theta, x) = \Delta_\theta \ln p(x/\theta)$
- Fisher Kernel $k(x, x') = g(\theta, x)^T F^{-1} g(\theta, x')$
- Fisher Information Matrix $F = E_x[g(\theta, x) g(\theta, x)^T]$

Radial Basis Functions

- Basis function depends only on the radial distance
- Developed for exact function interpolation
- Linear combination of radial basis functions, one centered on every data point.

$$f(x) = \sum_{n=1}^{N} w_n h(\|x - x_n\|)$$

- Interpolation when input variables are noisy. Noise on the input variable x is described by a variable e having a distribution v(e)

Radial Basis Functions

- Sum of squares error function is given by

$$E = \frac{1}{2}\sum_{n=1}^{N}\int \{y(x_n+e)-t_n\}^2 v(e)\,de$$

- Using calculus of variations, optimize w.r.t to the function f(x) to give $y(x_n) = \sum_{n=1}^{N} t_n h(x-x_n)$
- The basis functions are given by

$$h(x_n) = \frac{v(x-x_n)}{\sum_{n=1}^{N} v(x-x_n)}$$

Radial Basis Functions

- One basis function centered on every data point
- Basis functions are normalized.
- Computationally costly to evaluate when making predictions for new data points
- Choice of basis functions centers
  - Use any randomly chosen subset of the data points
  - Systematic approach – Orthogonal Least Squares
    - Sequential selection process based on sum of squares error

Nadaraya - Watson Model

- We have training set $\{x_n, t_n\}$. We use a parzen density estimator to model the joint distribution

$$p(x,t) = \frac{1}{N}\sum_{n=1}^{N} f(x - x_n, t - t_n)$$

- f(x,t) is the component density function and there is one such component centered on each data point
- Find the regression function y(x) corresponding to the conditional average of the target variable conditioned on the input variable.

Nadaraya - Watson Model

- Regression function

$$y(x) = E[t/x] = \int_{-\infty}^{\infty} t\, p(t/x)\, dt$$

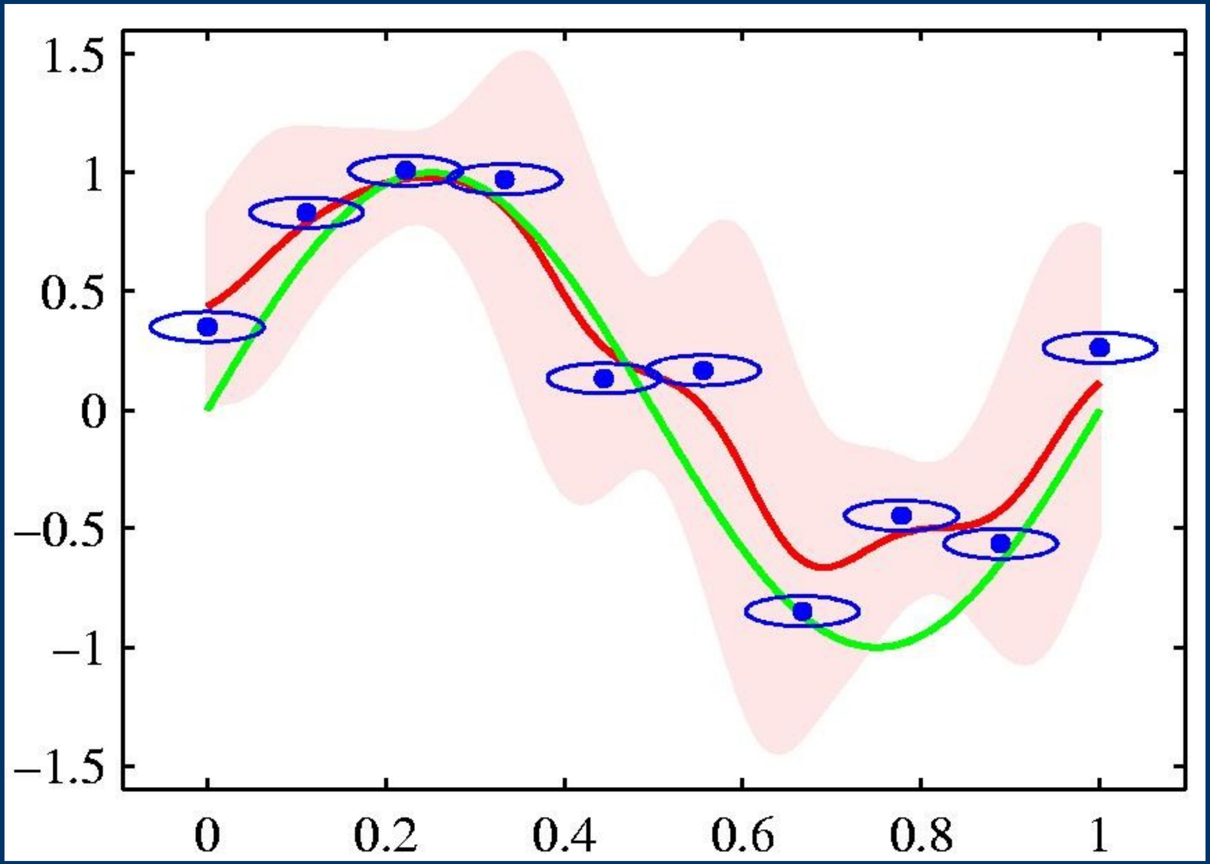
- Watson - Nadaraya Model

$$y(x) = \frac{\sum_n g(x - x_n) t_n}{\sum_m g(x - x_m)} = \sum_n k(x - x_n) t_n \quad \text{, where}$$

$$k(x - x_n) = \frac{g(x - x_n)}{\sum_m g(x - x_m)} \quad \text{and} \quad g(x) = \int_{-\infty}^{\infty} f(x, t)\, dt$$

Nadaraya - Watson Model

- Nadaraya – Watson Kernel Regression model using isotropic gaussian kernels for the sinusoidal data set



Gaussian Processes

- Dispose the parametric model in regression.
- Define a prior probability distribution over the functions directly
- Gaussian process is defined as the probability distribution over functions $y(x)$ such that the values of $y(x)$ evaluated at an arbitrary set of points $x_1, x_2, \ldots x_N$ jointly have a gaussian distribution.

Gaussian Processes for Regression

- Observed target values are noisy $t_n = y_n + \epsilon_n$
- Noise distribution $p(t_n / y_n) = N(t_n / y_n, \beta^{-1})$
- Joint distribution of target values conditioned on y

$$p(\boldsymbol{t} / \boldsymbol{y}) = N(\boldsymbol{t} / \boldsymbol{y}, \beta^{-1} \boldsymbol{I}_N)$$

- Marginal distribution $p(\boldsymbol{y}) = N(\boldsymbol{y} / \boldsymbol{0}, \boldsymbol{K})$
- Marginal Distribution conditioned on input values

$$p(\boldsymbol{t}) = \int p(\boldsymbol{t} / \boldsymbol{y}) p(\boldsymbol{y}) d\,\boldsymbol{y} = N(\boldsymbol{t} / \boldsymbol{0}, \boldsymbol{C})$$

$$C(x_n, x_m) = k(x_n, x_m) + \beta^{-1} \delta_{nm}$$

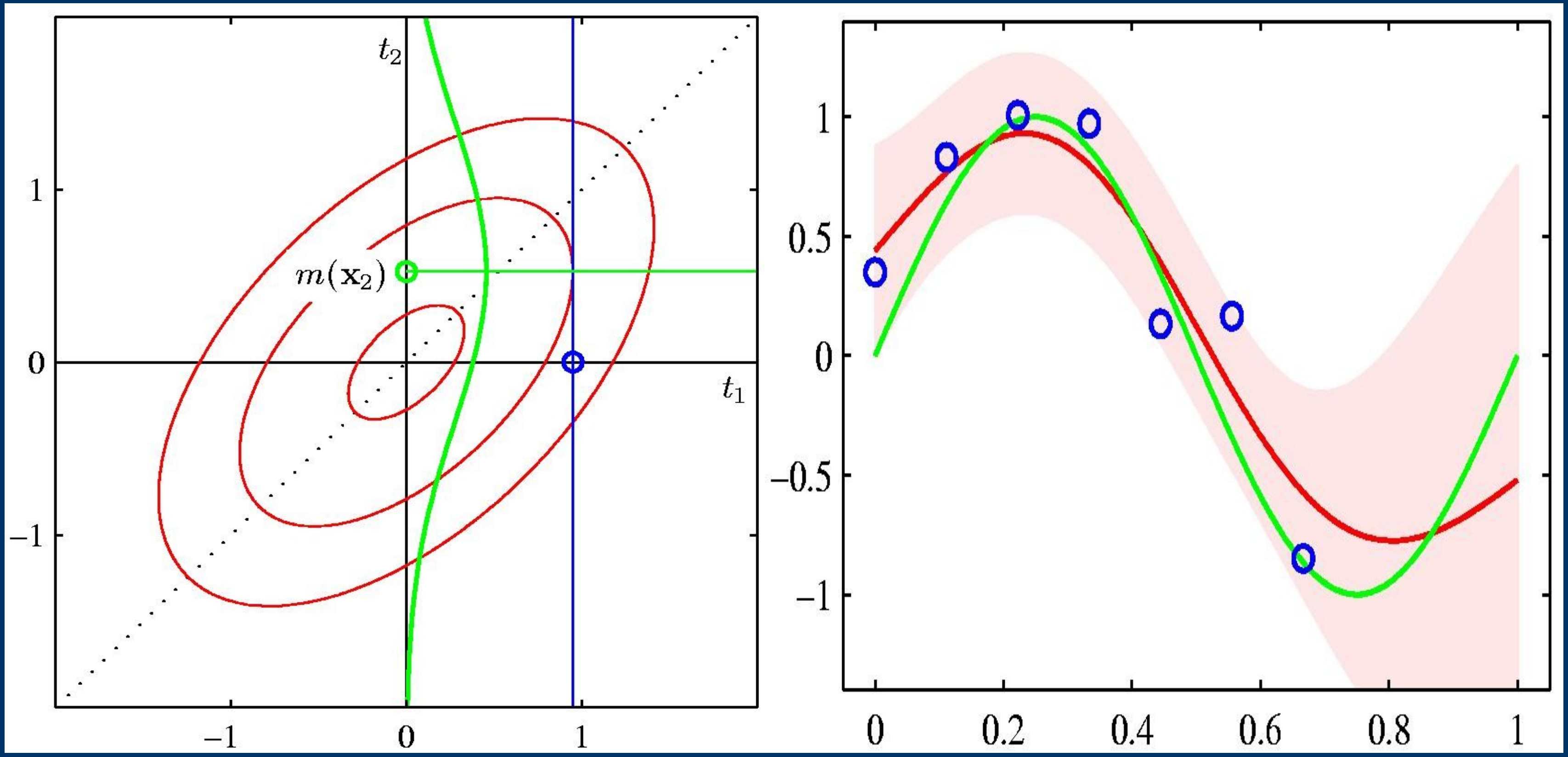
Gaussian Processes for Regression

- Predictive distribution $p(t_{N+1}/\mathbf{t}_N)$
- Joint distribution $p(\mathbf{t}_{N+1}) = N(\mathbf{t}_{N+1}/\mathbf{0}, \mathbf{C}_{N+1})$
- Partitioned Covariance Matrix $\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k} & c \end{pmatrix}$

  where k has elements $k(x_n, x_{N+1})$ and the scalar $c = k(x_{N+1}, x_{N+1}) + \beta^{-1}$
- The predictive distribution is gaussian distributed with mean and covariance given by $m(x_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}$

  $\sigma^2(x_{N+1}) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}$

Gaussian Processes for Regression



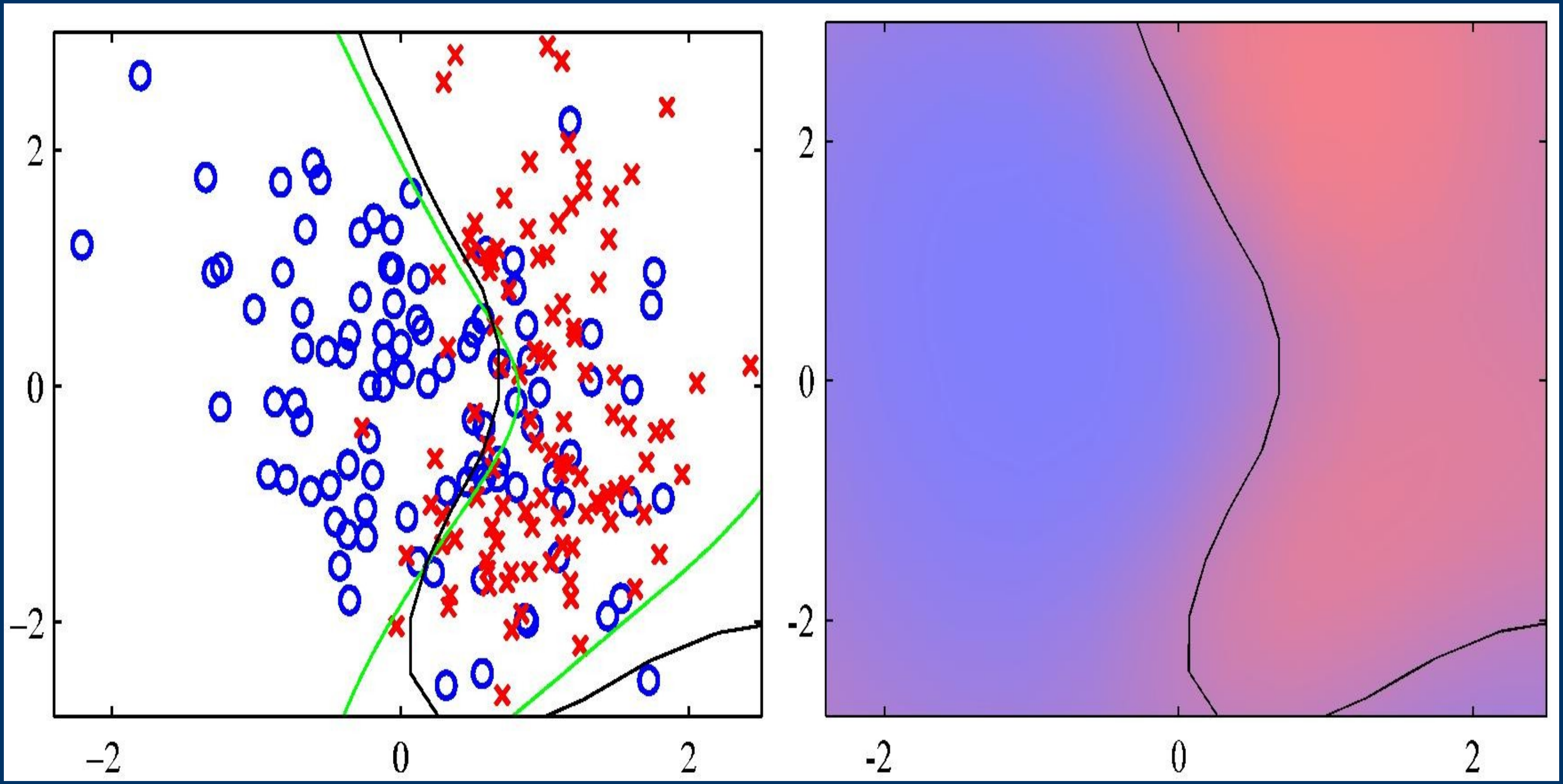
Gaussian Processes for Classification

- Probabilistic Approach → Model posterior probability → Values lie in the interval (0,1)
- Gaussian process model makes predictions that lie on the entire real axis
- Adapt gaussian processes to classification problems by transforming the output of the gaussian process using an appropriate non-linear activation function.

Gaussian Processes for Classification

- However, it is very difficult to arrive at a closed form analytical solution for the predictive distribution
- Consider approximation using sampling methods or analytical approximation
  - Variational Inference
  - Expectation Propagation
  - Laplace Approximation

Gaussian Processes for Classification



Connection to Neural Networks

- For a broad class of prior distributions over w, the distribution of functions generated by a neural network will tend to a gaussian process as $M \to \infty$
- In this limit, the output variables of the neural network become independent.
- Generally, the weights associated with each hidden unit in a neural network are influenced by all of the output variables , but this property is lost in the gaussian process limit.

References

- Pattern Recognition and Machine Learning
  - Christopher M. Bishop
- http://www.kernel-machines.org/
- http://www.learning-with-kernels.org/
- http://www.support-vector.net/
- http://www.gaussianprocess.org/
- Max Planck Institute for Biological Cybernetics
  - Bernhard Schölkopf, Prof. Dr.

Any Questions..??

Thank You..!!