Introduction to Statistical Machine Learning

Cheng Soon Ong & Christian Walder

Machine Learning Research Group
Data61 | CSIRO
and
College of Engineering and Computer Science
The Australian National University

Canberra
February – June 2017

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

Part I

Kernel Methods

Where are we?

- Basis function models (regression, classification)
- Flexible basis function models (neural networks) $\rightarrow$ last lecture

Ong & Walder
Data61 | CSIRO
The Australian National University

Where are we going?

- Why not use all training data to make predictions for the test inputs?
- Basic ideas:
  - Continuity : Mostly targets don't change abruptly.
  - Similarity : Each training pair (input, target) tells us something about the possible targets in the neighbourhood of the input.
- Kernels formalise those ideas.

Ong & Walder
Data61 | CSIRO
The Australian National University

How are we going there?

- Kernels for density estimation
  - Nonparametric density estimation
- Kernels for classification
  - Basis functions and the kernel trick
  - Constructing kernels

Warning: The term 'kernel' is also used for all vectors mapping under some matrix A to zero. This is a different concept. Don't get confused!

©2016
Ong & Walder
Data61 | CSIRO
The Australian National University

Kernel Density Estimation

Ong & Walder
Data61 | CSIRO
The Australian National University

Density Estimation

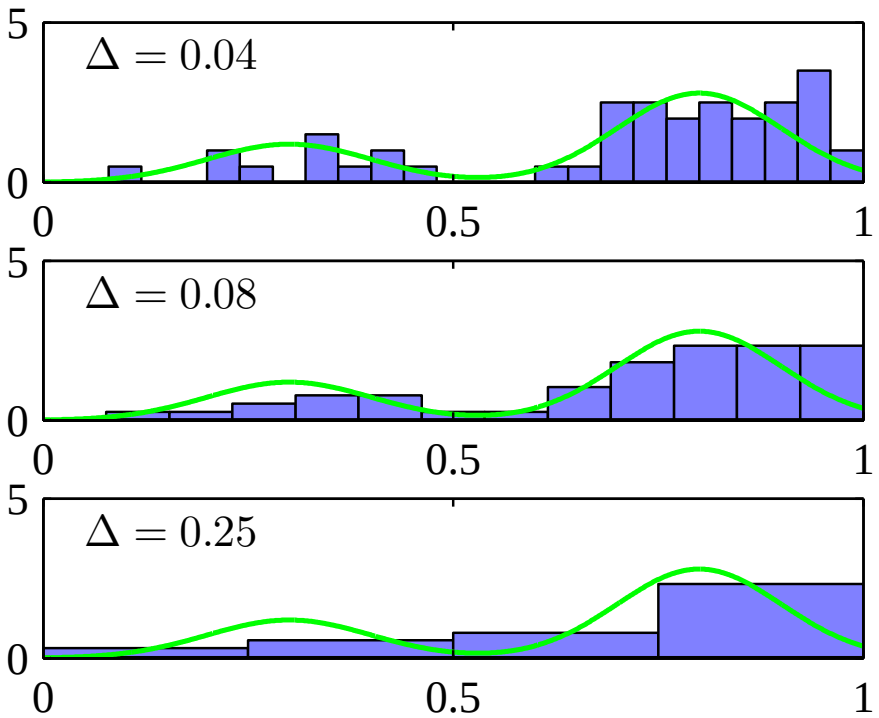
- Suppose we observe data points $\{\mathbf{x}_n\}_{n=1}^N$
  - e.g. just N real numbers
- Suppose we believe these are drawn independently from some distribution $p(\mathbf{x})$
  - e.g. $p(\mathbf{x})$ is Gaussian with unknown mean and variance
- **Density estimation problem**: Estimate $p(\mathbf{x})$ from data

Ong & Walder
Data61 | CSIRO
The Australian National University

Nonparametric Density Estimation – Histogram

- Partition the space into bins of width Δ_i.
- Count the number n_i of samples falling into each bin i.
- Normalise.

$$p_i = \frac{n_i}{N\Delta_i}$$



Histogram of 50 data points generated from the distribution shown by the green curve for varying common bin width Δ

Ong & Walder
Data61 | CSIRO
The Australian National University

Nonparametric Density Estimation – Histogram

Advantages:

- Data can be discarded after calculating the p_i.
- Algorithm can be applied to sequentially arriving data.

Disadvantages:

- Dependency on bin width Δ_i.
- Discontinuities due to the bin edges.
- Exponential scaling with the dimensionality D of the data. Need M^D bins for D dimensions and M bins per dimension.

Data61 | CSIRO
The Australian National University

Nonparametric Density Estimation - Refined

- Draw data from some unknown probability distribution $p(\mathbf{x})$ in a D-dimensional space.
- Consider a small region $\mathcal{R}$ containing $\mathbf{x}$. Probability mass associated with this region

$$P = \int_{\mathcal{R}} p(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

- Data set of N observations drawn from $p(\mathbf{x})$. Total number K of points found inside of $\mathcal{R}$ is distributed according to the binomial distribution

$$\mathrm{Bin}(K \,|\, N, P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K}$$

- Expectation of K : $\mathbb{E}\left[K/N\right] = P$
- Variance of K : $\mathrm{var}[K/N] = P(1-P)/N$

Data61 | CSIRO
The Australian National University

Nonparametric Density Estimation - Refined

- Expectation of K : $\mathbb{E}\left[K/N\right] = P$
- Variance of K : $\mathrm{var}[K/N] = P(1-P)$
- For large N, the distribution will be sharply peaked and therefore

$$K \approx NP$$

- Assuming also that the region has volume V and the region is small enough for $p(\mathbf{x})$ to be roughly constant, then

$$P \approx p(\mathbf{x})V$$

- Combining two contradictory assumptions
  - Region $\mathcal{R}$ is small enough for $p(\mathbf{x})$ to be roughly constant.
  - Region $\mathcal{R}$ is large enough to have enough K points falling into it to get a sharp peak for the binomial distribution.

$$p(\mathbf{x}) \approx \frac{K}{NV}$$

Data61 | CSIRO
The Australian National University

Nonparametric Density Estimation - Refined

- Two ways to exploit

$$p(\mathbf{x}) \approx \frac{K}{NV}$$

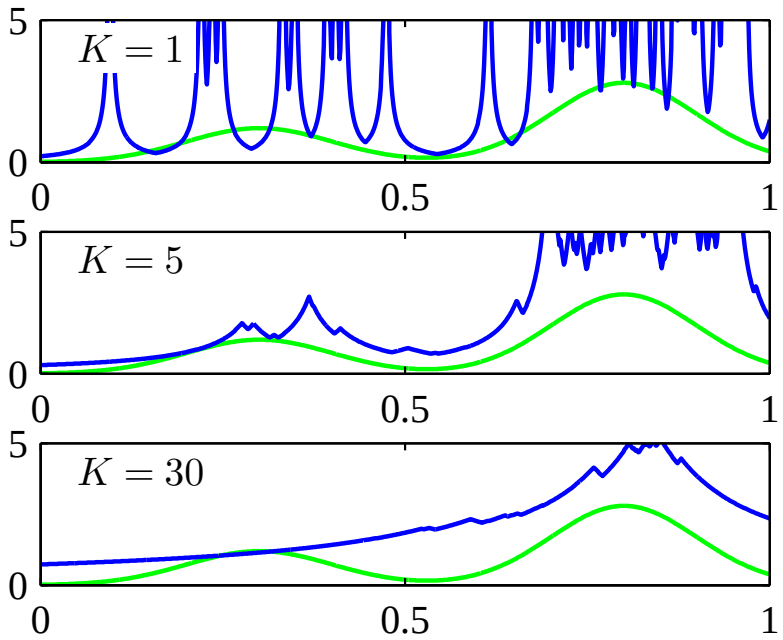
1. Fix K and determine the volume V from the data : K-nearest-neighbours density estimation
2. Fix V and determine K from the data : kernel density estimation

Ong & Walder
Data61 | CSIRO
The Australian National University

Nonparametric Estimation – Nearest Neighbour

- Fix K and find an appropriate value for V.
- Consider a small sphere around $\mathbf{x}$ and then allow the radius to increase until it contains exactly K data points.
- Calculate the probability by

$$p(\mathbf{x}) \approx \frac{K}{NV}$$



Nearest neighbour density model for different K.

Data61 | CSIRO
The Australian National University

Nonparametric Estimation – Parzen Estimator

- Define region $\mathcal{R}$ to be a small hypercube around $\mathbf{x}$
- Define Parzen window (kernel function)

$$k(\mathbf{u}) = \begin{cases} 1, & |u_i| \leq 1/2, \qquad i = 1, \ldots, D \\ 0, & \text{otherwise} \end{cases}$$

- Total number of data points inside of the hypercube centered at $\mathbf{x}$ with lengths h:

$$K = \sum_{n=1}^{N} k\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right)$$

- Density estimate for $p(\mathbf{x})$

$$p(\mathbf{x}) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^D} k\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right)$$

- Interpret as sum over N cubes centered at each of the $\mathbf{x}_n$.

Nonparametric Estimation – Parzen Estimator

- Remaining problem: Discontinuities because of the hypercube (either in or out).
- Choose a smoother kernel function (and normalise correctly).
- Common choice : Gaussian kernel

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{D/2}}\exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\right\}$$

- Can choose any other kernel function $k(\mathbf{u})$ obeying

$$k(\mathbf{u}) \geq 0,$$

$$\int k(\mathbf{u})\,\mathrm{d}\mathbf{u} = 1$$

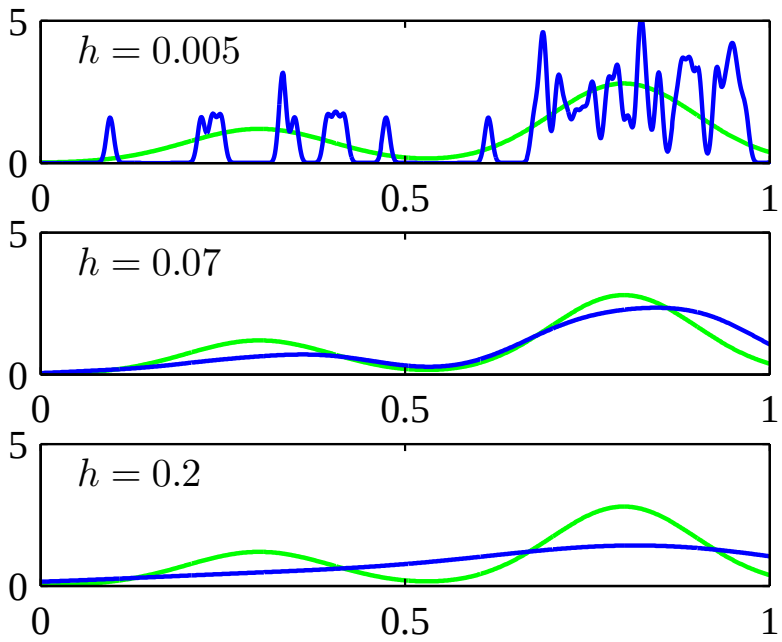
Data61 | CSIRO
The Australian National University

Nonparametric Estimation – Parzen Estimator

- Gaussian kernel

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{D/2}}\exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\right\}$$

- h controls the trade-off between sensitivity to noise and over-smoothing.



Kernel density model with Gaussian kernel for different h.

Data61 | CSIRO
The Australian National University

Kernel Methods for Classification

Ong & Walder
Data61 | CSIRO
The Australian National University

The Role of Training Data

- Parametric methods
  - Learn the model parameter $\mathbf{w}$ from the training data $\mathbf{t}$.
  - Discard the training data $\mathbf{t}$.
- Nonparametric methods
  - Use training data directly for prediction
  - k-nearest neighbours : use k-closest data from the 'training' set for classification
- Kernel methods
  - Base prediction on linear combination of kernel functions evaluated at the training data.

©2016 Ong & Walder Data61 | CSIRO The Australian National University

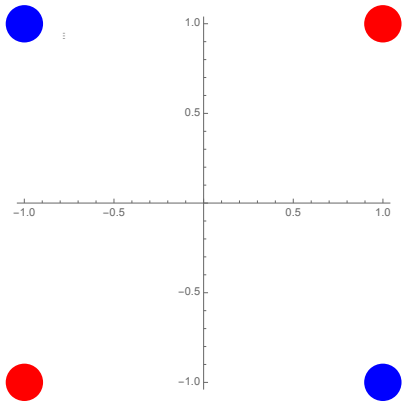
Features

- A feature is a measurable property of a phenomenon being observed or any derived property thereof
  - raw features: the original data
  - derived features: mappings of the original features to some other space (possibly high- or infinite dimensional, e.g., basis functions)
- Feature selection: which features matter for the problem at hand?
  - redundant features
  - problem dependent
- Feature extraction: can we combine the important features to a smaller set of new features?
  - compact representation versus ability to explain to a human

Ong & Walder
Data61 | CSIRO
The Australian National University

Very simple example - XOR

x_1	x_2	$y = x_1 \text{xor} x_2$
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

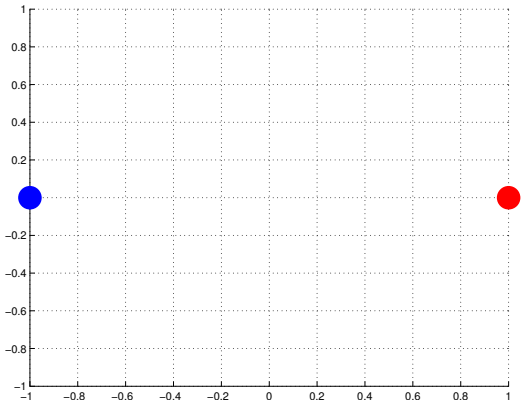


- not linearly separable (why?)
- raw features $\{(-1,-1),(-1,1),(1,-1),(1,1)\}$

Ong & Walder
Data61 | CSIRO
The Australian National University

Very simple example - XOR

x_1	x_2	$x_{\text{new}} = x_1 \cdot x_2$	$y = x_1 \text{xor} x_2$
-1	-1	1	1
-1	1	-1	-1
1	-1	-1	-1
1	1	1	1



- feature extraction: $x_{\text{new}} = x_1 \cdot x_2$
- data is now separable!
- All classification algorithms work also if we first apply a fixed nonlinear transformation of the inputs using a vector of basis functions $\phi(\mathbf{x})$.

Ong & Walder
Data61 | CSIRO
The Australian National University

Kernel methods in one slide

- Consider a labelled training set $\{\mathbf{x}_i, t_i\}_{i=1}^N$
- On a new point $\mathbf{x}$, we will predict

$$y(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \cdot K(\mathbf{x}, \mathbf{x}_i) \cdot t_i$$

  where $\{\alpha_i\}_{i=1}^N$ are weights to be determined, and $K(\cdot, \cdot)$ is a kernel function
- The kernel function measures the similarity between any two examples
  - Prediction is a weighted average of the training targets
  - Weights depend on the similarity of $\mathbf{x}$ to each training example

Ong & Walder
Data61 | CSIRO
The Australian National University

Kernel Methods - Intuition

- Suppose we perform linear regression with a feature matrix $\boldsymbol{\Phi}$ and target vector $\mathbf{t}$, where

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}_1)^T \\ \boldsymbol{\phi}(\mathbf{x}_2)^T \\ \dots \\ \boldsymbol{\phi}(\mathbf{x}_N)^T \end{bmatrix}$$

- Recall that the optimal (regularised) $\mathbf{w}^\star$ is

$$\mathbf{w}^\star = (\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{t}$$

- Thus, the prediction for feature vector of new point $\mathbf{x}$ with

$$y(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}^\star = \boldsymbol{\phi}(\mathbf{x})^T (\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{t}$$

Kernel Methods - Intuition

- Prediction with optimal (regularised) $\mathbf{w}^\star$

$$y(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}^\star = \boldsymbol{\phi}(\mathbf{x})^T (\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{t}$$

- Suppose that M is very large. Then, the inverse of an $M \times M$ matrix above will be expensive to compute.
- Consider however the following trick:

$$\boldsymbol{\phi}(\mathbf{x})^T (\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{t} = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\Phi}^T (\lambda \mathbf{I} + \boldsymbol{\Phi} \boldsymbol{\Phi}^T)^{-1} \mathbf{t}$$

Kernel Methods - Intuition

- We have thus written the prediction as

$$y(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{\Phi}^T (\lambda \mathbf{I} + \mathbf{\Phi}\mathbf{\Phi}^T)^{-1} \mathbf{t}$$

- Now, our prediction is determined by an $N \times N$ rather than $M \times M$ matrix
- $\mathbf{\Phi}\mathbf{\Phi}^T$ is known as the kernel matrix of the training data
  - Intuitively, measures the similarities between the training instances
  - Why? Because the inner product between two points is a measure of similarity:

$$\arg\max_{\mathbf{v}} \langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}.$$

©2016 Ong & Walder
Data61 | CSIRO
The Australian National University

Dual Representations

- Consider a linear regression model with regularised sum-of-squares error

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - t_n)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

  where $\lambda \geq 0$.
- We could also write this in more compact form as

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{t} - \boldsymbol{\Phi}\mathbf{w})^T(\mathbf{t} - \boldsymbol{\Phi}\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

  with the target vector $\mathbf{t} = (t_1, \ldots, t_N)^T$, and the design matrix

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \ldots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \ldots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \ldots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}.$$

Dual Representations

- Critical points for $J(\mathbf{w})$

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{t} - \mathbf{\Phi w})^T(\mathbf{t} - \mathbf{\Phi w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

satisfy

$$\begin{aligned}
\mathbf{w} &= (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\mathbf{\Phi}^T\mathbf{t} \\
(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I})\mathbf{w} &= \mathbf{\Phi}^T\mathbf{t} \\
\lambda\mathbf{w} &= \mathbf{\Phi}^T(\mathbf{t} - \mathbf{\Phi w}) \\
\mathbf{w} &= \mathbf{\Phi}^T\mathbf{a} \\
&= \sum_{n=1}^{N} \phi(\mathbf{x}_n)a_n
\end{aligned}$$

where $\mathbf{a} = (a_1, \ldots, a_N)^T$ with components

$$a_n = -\frac{1}{\lambda}\left\{\mathbf{w}^T\phi(\mathbf{x}_n) - t_n\right\}$$

Ong & Walder
Data61 | CSIRO
The Australian National University

Dual Representations

- Now express $J(\mathbf{w})$ as a function of this new variable $\mathbf{a}$ instead of $\mathbf{w}$ via the relation $\mathbf{w} = \mathbf{\Phi}^T\mathbf{a}$

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a} - \mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a}$$

  where again $\mathbf{t} = (t_1, \ldots, t_N)^T$.
  - Known as the dual representation
- Define the $N \times N$ Gram matrix $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$ with elements

$$K_{nm} = \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m).$$

- Express $J(\mathbf{a})$ now as

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^T\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^T\mathbf{K}\mathbf{a}.$$

Data61 | CSIRO
The Australian National University

Critical Points of $J(\mathbf{a})$

- Let's calculate the critical points for

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^T\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^T\mathbf{K}\mathbf{a}.$$

- Directional derivative

$$\mathcal{D}J(\mathbf{a})(\boldsymbol{\xi}) = \boldsymbol{\xi}^T\mathbf{K}\mathbf{K}\mathbf{a} - \boldsymbol{\xi}^T\mathbf{K}\mathbf{t} + \lambda\,\boldsymbol{\xi}^T\mathbf{K}\mathbf{a}$$

  should be zero in all possible directions $\boldsymbol{\xi}$.
- Therefore $\mathbf{K}(\mathbf{K}\mathbf{a} - \mathbf{t} + \lambda\,\mathbf{a}) = 0$ and so

$$\mathbf{a} = (\mathbf{K} + \lambda\,\mathbf{I}_N)^{-1}\mathbf{t}.$$

- Second directional derivative (using $\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$)

$$\mathcal{D}^2J(\mathbf{a})(\boldsymbol{\xi},\boldsymbol{\xi}) = \boldsymbol{\xi}^T\mathbf{K}\mathbf{K}\boldsymbol{\xi} + \lambda\,\boldsymbol{\xi}^T\mathbf{K}\boldsymbol{\xi} = \|\mathbf{K}\boldsymbol{\xi}\|^2 + \lambda\|\boldsymbol{\Phi}^T\boldsymbol{\xi}\| > 0.$$

- $\mathbf{a} = (\mathbf{K} + \lambda\,\mathbf{I}_N)^{-1}\mathbf{t}$ minimises $J(\mathbf{a})$.

Data61 | CSIRO
The Australian National University

Prediction for the Linear Regression Model

- Inserting the argument $\mathbf{a}$ which minimises the error $J(\mathbf{a})$ into the prediction model for the linear regression, we get for the prediction

$$\begin{aligned} y(\mathbf{x}) &= \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}^T \boldsymbol{\Phi} \boldsymbol{\phi}(\mathbf{x}) = (\boldsymbol{\Phi}\boldsymbol{\phi}(\mathbf{x}))^T \mathbf{a} \\ &= \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{t} \end{aligned}$$

  where we defined the vector $\mathbf{k}(\mathbf{x})$ with elements $k_n(\mathbf{x}) = k(\mathbf{x}_n, \mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}_n)^T \boldsymbol{\phi}(\mathbf{x})$.
- The prediction $y(\mathbf{x})$ can be expressed entirely in terms of the kernel function $k(\mathbf{x}, \mathbf{x}')$ evaluated at the training and test data.
- Looks familiar? See Bayesian Linear Regression.

The Kernel Function

- The kernel function is defined over two points, $\mathbf{x}$ and $\mathbf{x}'$, of the input space

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}').$$

- $k(\mathbf{x}, \mathbf{x}')$ is symmetric.
- It is an inner product of two vectors of basis functions

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle.$$

- For prediction, the kernel function will be evaluated at the training data points. (See next slides.)

Ong & Walder
Data61 | CSIRO
The Australian National University

Dual Representation

- What have we gained by the dual representation?
- Need to invert an $N \times N$ matrix now, where N is the number of data points. Can be large!
  - In the parameter space formulation, we "only" needed to invert an $M \times M$ matrix, where M was the number of basis functions.
  - But, a kernel corresponds to an inner product of basis functions. So we can use a large number of basis functions, even infinitely many.
- We can construct new valid kernels directly from given ones (whatever the corresponding basis functions of the new kernel might be).
- As a kernel defines a kind of 'similarity' between two points in the input space, we can define kernels over graphs, sets, strings, and text documents.

Data61 | CSIRO
The Australian National University

Kernel Construction

Kernels from Basis Functions

1. Choose a set of basis functions

$$\{\phi_1, \ldots, \phi_M\}$$

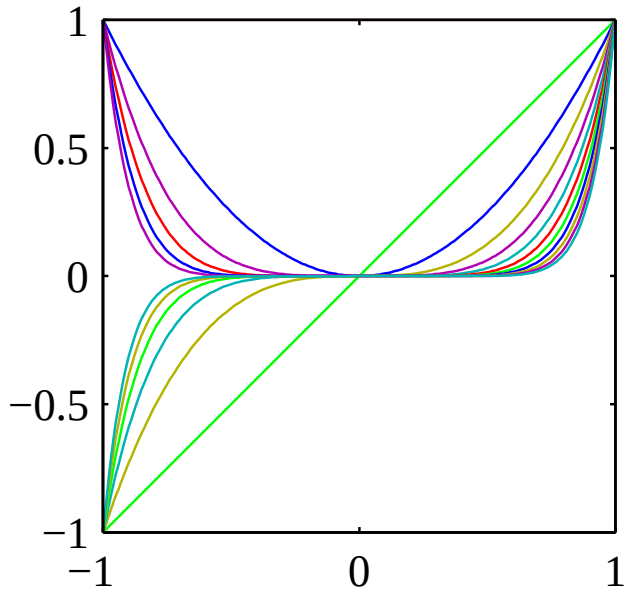
2. Find a new kernel as an inner product between vectors of basis functions evaluated at x and x'

$$k(x, x') = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x) = \sum_{i=1}^{M} \phi_i(x)\phi_i(x')$$

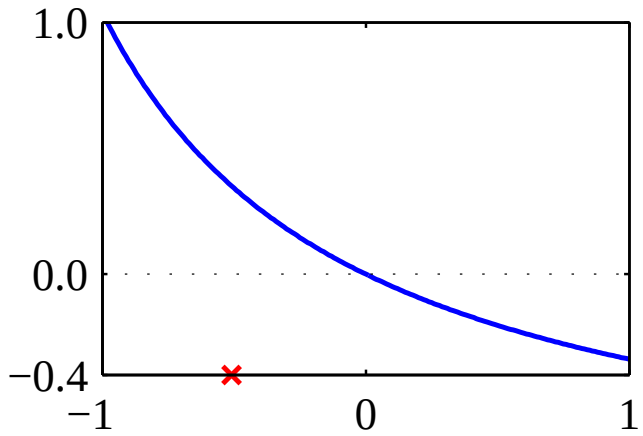
Ong & Walder
Data61 | CSIRO
The Australian National University

Kernels from Basis Functions

Polynomial basis functions



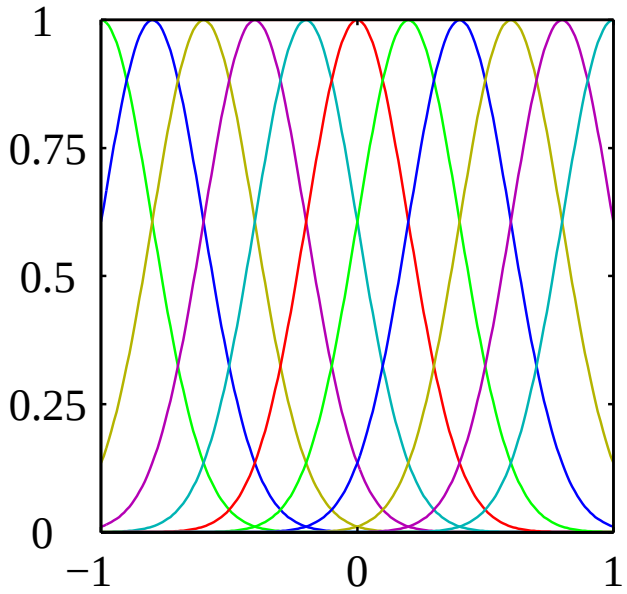
Corresponding kernel $k(x, x')$ as function of x for $x' = -0.5$ (red cross).



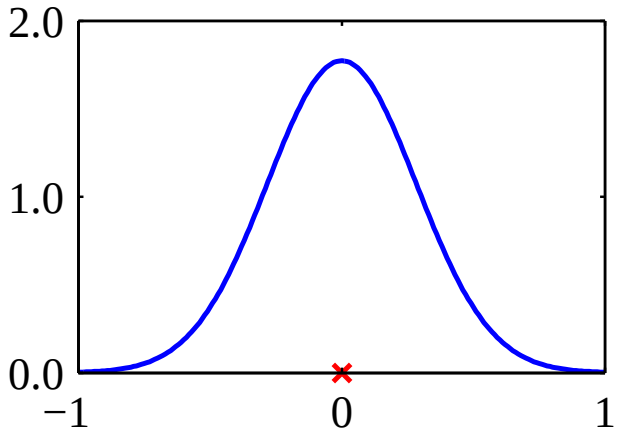
Ong & Walder
Data61 | CSIRO
The Australian National University

Kernels from Basis Functions

Gaussian basis functions



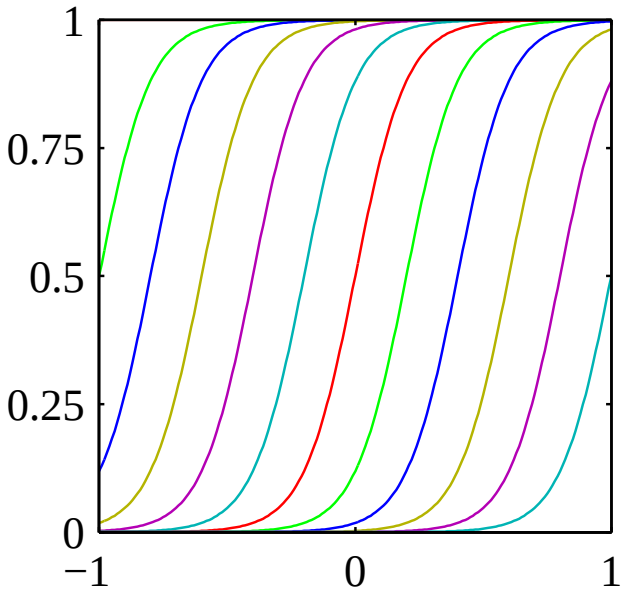
Corresponding kernel $k(x, x')$ as function of x for $x' = 0.0$ (red cross).



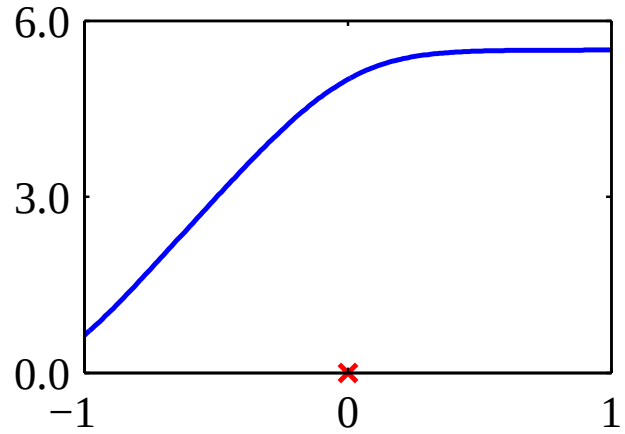
Ong & Walder
Data61 | CSIRO
The Australian National University

Kernels from Basis Functions

Logistic Sigmoid basis functions



Corresponding kernel $k(x, x')$ as function of x for $x' = 0.0$ (red cross).



Ong & Walder
Data61 | CSIRO
The Australian National University

Kernels by Guessing a Kernel Function

1. Choose a mapping from two points of the input space to a real number, which is symmetric in its arguments, e.g.

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2 = k(\mathbf{z}, \mathbf{x})$$

2. Try to write this as an inner product of a vector valued function evaluated at the arguments $\mathbf{x}$ and $\mathbf{z}$, e.g.

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^T \mathbf{z})^2 = (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2 x_1 z_2 x_2 z_2 + x_2^2 z_2^2 \\ &= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)(z_1^2, \sqrt{2} z_1 z_2, z_2^2)^T \\ &= \phi(\mathbf{x})^T \phi(\mathbf{z}) \end{aligned}$$

with the feature mapping $\phi(\mathbf{x}) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T$.

New Kernels From Theory

- A necessary and sufficient condition for $k(\mathbf{x}, \mathbf{x}')$ to be a valid kernel is that the kernel matrix $\mathbf{K}$, whose elements are $k(\mathbf{x}_n, \mathbf{x}_m)$, should be positive semidefinite for all possible choices of the set $\{\mathbf{x}_n\}$.
- Previously, we constructed $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$, which is automatically positive semidefinite (why?)
  - If we can explicitly construct the kernel via basis functions, we are good
  - Even if we cannot find the basis functions easily, we may be able to deduce $k(\mathbf{x}, \mathbf{x}')$ is a valid kernel

Data61 | CSIRO
The Australian National University

New Kernels From Other Kernels

Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, the following kernels are also valid:

$$k(\mathbf{x}, \mathbf{x}') = c\, k_1(\mathbf{x}, \mathbf{x}')$$
$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})\, k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')$$
$$k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}'))$$
$$k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}'))$$
$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$$
$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')\, k_2(\mathbf{x}, \mathbf{x}')$$
$$k(\mathbf{x}, \mathbf{x}') = k_3(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}'))$$
$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{A} \mathbf{x}'$$
$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b)$$
$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a)\, k_b(\mathbf{x}_b, \mathbf{x}'_b)$$

$c > 0$ constant

$f(\cdot)$ any function

$q(\cdot)$ polynomial with nonneg. coeff.

$\boldsymbol{\phi}(\mathbf{x})$ any function to $\mathbb{R}^M$

$k_3(\cdot, \cdot)$ valid kernel in $\mathbb{R}^M$

$\mathbf{A} = \mathbf{A}^T, \mathbf{A} \succeq 0$

$\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$

New Kernels From Other Kernels

Further examples of kernels

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')^M \qquad \text{only terms of degree } M$$

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^M \qquad \text{all terms up to degree } M$$

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2\right) \qquad \text{Gaussian kernel}$$

$$k(\mathbf{x}, \mathbf{x}') = \tanh\left(a\, \mathbf{x}^T \mathbf{x}' + b\right) \qquad \text{Sigmoidal kernel}$$

Generally, we call

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' \qquad \text{linear kernel}$$

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}') \qquad \text{stationary kernel}$$

$$k(\mathbf{x}, \mathbf{x}') = k(\|\mathbf{x} - \mathbf{x}'\|) \qquad \text{homogeneous kernel}$$

Kernels over Graphs, Sets, Strings, Texts

- We 'only' need an appropriate similarity measure $k(\mathbf{x}, \mathbf{x}')$ which is a kernel.
- Example: Given a set $\mathcal{A}$ and the set of all subsets of $\mathcal{A}$, called the power set $\mathcal{P}(\mathcal{A})$.
- For two subsets $\mathcal{A}_1, \mathcal{A}_2 \in \mathcal{P}(\mathcal{A})$, denote the number of elements of the intersection of $\mathcal{A}_1$ and $\mathcal{A}_2$ by $|\mathcal{A}_1 \cap \mathcal{A}_2|$.
- Then it can be shown that

$$k(\mathcal{A}_1, \mathcal{A}_2) = 2^{|\mathcal{A}_1 \cap \mathcal{A}_2|}$$

corresponds to an inner product in a feature space. Therefore, $k(\mathcal{A}_1, \mathcal{A}_2)$ is a valid kernel function.

Data61 | CSIRO
The Australian National University

Kernels from Probabilistic Generative Models

- Given $p(\mathbf{x})$, we can define a kernel

$$k(\mathbf{x}, \mathbf{x}') = p(\mathbf{x})\, p(\mathbf{x}'),$$

  which means two inputs $\mathbf{x}$ and $\mathbf{x}'$ are similar if they both have high probabilities.
- Include a weighting function $p(i)$ and extend the kernel to

$$k(\mathbf{x}, \mathbf{x}') = \sum_i p(\mathbf{x} \mid i)\, p(\mathbf{x}' \mid i) p(i).$$

- For a continous variable $\mathbf{z}$

$$k(\mathbf{x}, \mathbf{x}') = \int p(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{x}' \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z}.$$

- Hidden Markov Model with sequences of length L.

Data61 | CSIRO
The Australian National University

Kernels for Classification: Summary

- Pick a suitable kernel function $k(\mathbf{x}, \mathbf{x}')$
  - e.g. by computing inner product of some basis functions
- Make predictions by suitably combining $k(\mathbf{x}, \mathbf{x}_n)$ for each training example $\mathbf{x}_n$
  - implicitly, a linear model in some high-dimensional space
- For linear regression, we go from

$$y(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T(\lambda\mathbf{I} + \boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{t}$$

to

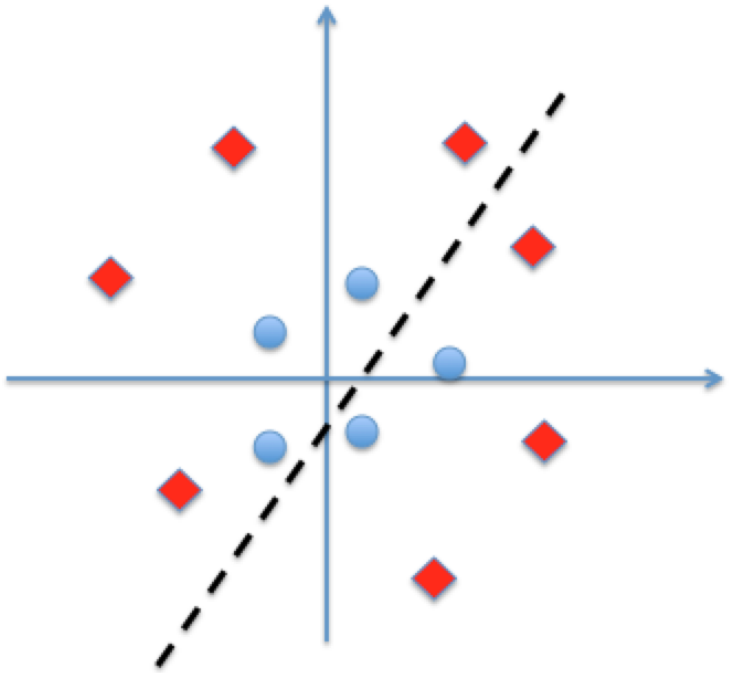
$$y(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\,\mathbf{I}_N)^{-1}\mathbf{t}$$

  - can plug in suitable kernel function to implicitly perform nonlinear transformation

Data61 | CSIRO
The Australian National University

Kernels for Classification: Summary

- Working with a nonlinear kernel, we are implicitly performing a nonlinear transformation of our data

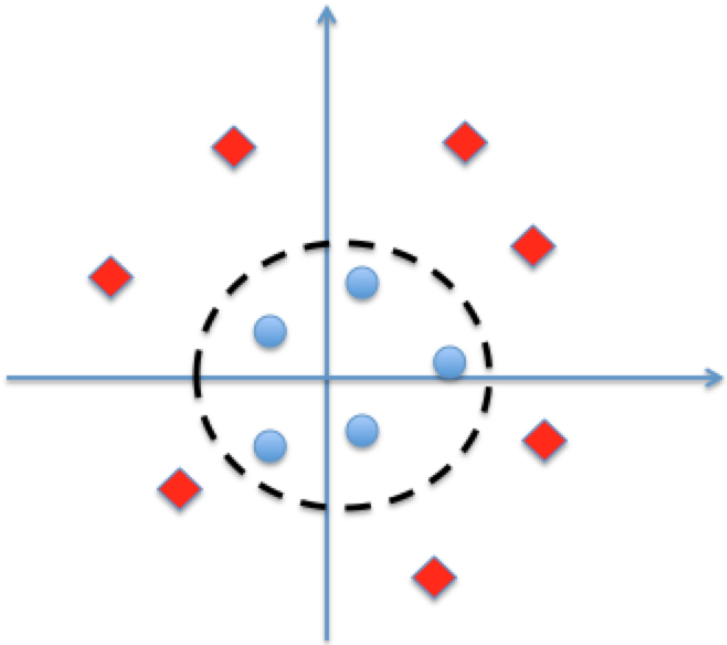


Classification with linear kernel $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$

Ong & Walder
Data61 | CSIRO
The Australian National University

Kernels for Classification: Summary

- Working with a nonlinear kernel, we are implicitly performing a nonlinear transformation of our data



Classification with nonlinear kernel $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')^2$

Data61 | CSIRO
The Australian National University