Introduction to Statistical Machine Learning

Christfried Webers

Statistical Machine Learning Group
NICTA
and
College of Engineering and Computer Science
The Australian National University

Canberra
February – June 2013

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

Outlines

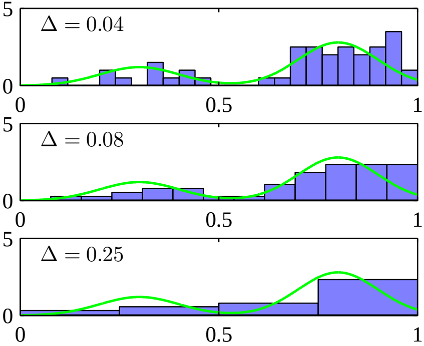
Part XI

Kernel Methods

Nonparametric Density Estimation – Histogram

- Partition the space x 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

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.

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

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

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

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.

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

Nonparametric Density Estimation - Refined

- Two ways to exploit

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

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

Nonparametric Estimation – Parzen Estimator

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

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

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

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

- 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

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

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

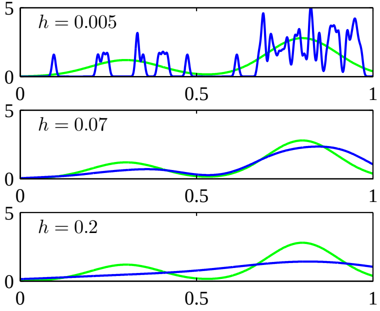
$$k(\mathbf{u}) \geq 0,$$
$$\int k(\mathbf{u}) \, \mathrm{d}\mathbf{u} = 1$$

Nonparametric Estimation – Parzen Estimator

- Gaussian kernel

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

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

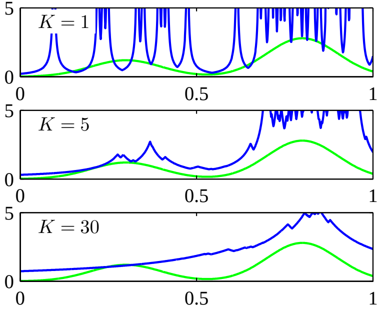


Kernel density model with Gaussian kernel for different h.

Nonparametric Estimation – Nearest Neighbour

- Now, 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.

The Role of Training Data

- Parametric methods
  1. Learn the model parameter $\mathbf{w}$ from the training data $\mathbf{t}$.
  2. 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
  - Parzen probability density model : set of functions centered on the training data
- Kernel methods
  - Base prediction on linear combination of kernel functions evaluated at the training data.

Dual Representations

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

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

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

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

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

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

Dual Representations

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

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

can be found as

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

by introducing the new vector $\mathbf{a} = (a_1, \dots, a_N)^T$ with components

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

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 $\mathbf{t} = (t_1, \dots, t_N)^T$.

- 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

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

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.)

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 as $\mathbf{K}$ has full rank

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

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

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

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

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

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

  where we defined the vector $\mathbf{k}(\mathbf{x})$ with elements $k_n(\mathbf{x}) = k(\mathbf{x}_n, \mathbf{x}) = \phi(\mathbf{x}_n)^T \phi(\mathbf{x})$.
- 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.

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 'distance' between two points in the input space, we can define kernels over graphs, sets, strings, and text documents.

Kernels from Basis Functions

1. Choose a set of basis functions

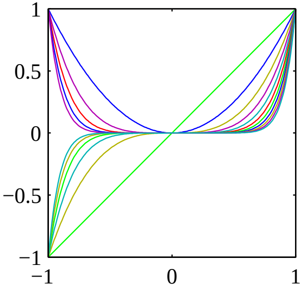
$$\{\phi_1, \dots, \phi_M\}$$

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

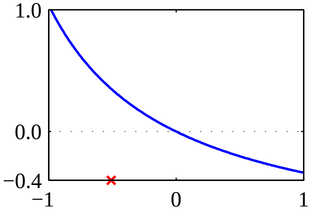
$$k(x, x') = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x) = \sum_{i=1}^{M} \phi_i(x) \phi_i(x')$$

Kernels from Basis Functions

Polynomial basis functions

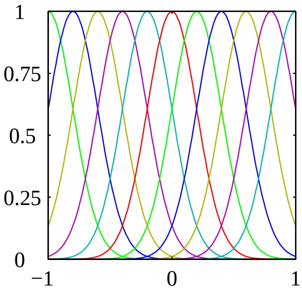


Corresponding kernel $k(x, x')$ as function of x for $x' = -0.5$ (red cross).

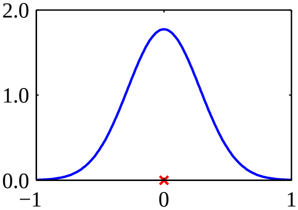


Kernels from Basis Functions

Gaussian basis functions

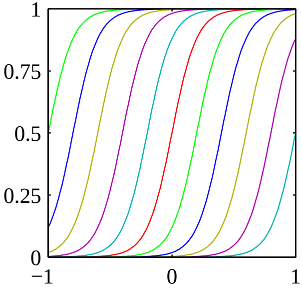


Corresponding kernel $k(x, x')$ as function of x for $x' = 0.0$ (red cross).

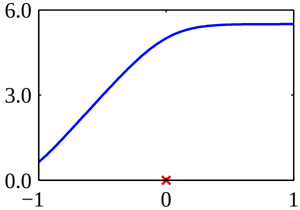


Kernels from Basis Functions

Logistic Sigmoid basis functions



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



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

1. A necessary and sufficient condition for $k(\mathbf{x}, \mathbf{x}')$ to be a valid kernel is that the Gram matrix $\mathbf{K}$, whose elements are $k(\mathbf{x}_n, \mathbf{x}_m)$, should be positive semidefinite for all possible choices of the set $\{\mathbf{x}_n\}$.

Note: The Gram matrix $\mathbf{K}$ was defined with the help of the input data $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$. The kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$ defines the entries in the Gram matrix $K_{nm} = k(\mathbf{x}_n, \mathbf{x}_m)$ depending on two input data points $\mathbf{x}_n$ and $\mathbf{x}_m$. The above therefore says, that $k(\mathbf{x}, \mathbf{x}')$ is a valid kernel if the Gram matrix is positive semidefinite for any set of input data.

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:

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

$c > 0$	constant
$f(\cdot)$	any function
$q(\cdot)$	polynomial with nonneg. coeff.
$\phi(\mathbf{x})$	any function to $\mathbb{R}^M$
$k_3(\cdot, \cdot)$	valid kernel in $\mathbb{R}^M$

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

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

New Kernels From Other Kernels

Further examples of kernels

$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')^M$	only terms of degree M
$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^M$	all terms up to degree M
$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\lVert\mathbf{x} - \mathbf{x}'\rVert^2 / 2\sigma^2\right)$	Gaussian kernel
$k(\mathbf{x}, \mathbf{x}') = \tanh\left(a\,\mathbf{x}^T \mathbf{x}' + b\right)$	Sigmoidal kernel

Generally, we call

$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$	linear kernel
$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$	stationary kernel
$k(\mathbf{x}, \mathbf{x}') = (\lVert\mathbf{x} - \mathbf{x}'\rVert)$	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.

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 $\mathbf{z}$

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

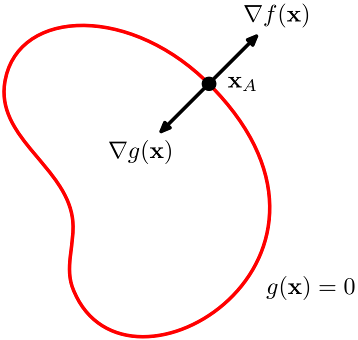
- Hidden Markov Model with sequences of length L.

Lagrange Multipliers

- Find the stationary points for a function $f(x_1, x_2)$ subject to one or more constraints on the variables x_1 and x_2 written in the form $g(x_1, x_2) = 0$.
- Direct approach
  1. Solve $g(x_1, x_2) = 0$ for one of the variables to get $x_2 = h(x_1)$.
  2. Insert the result into $f(x_1, x_2)$ to get a function of one variable $f(x_1, h(x_1))$.
  3. Find the stationary point(s) $x_1^\star$ of $f(x_1, h(x_1))$ with corresponding value $x_2^\star = h(x_1^\star)$.
- Finding $x_2 = h(x_1)$ may be hard.
- Symmetry in the variables x_1 and x_2 is lost.

Lagrange Multipliers

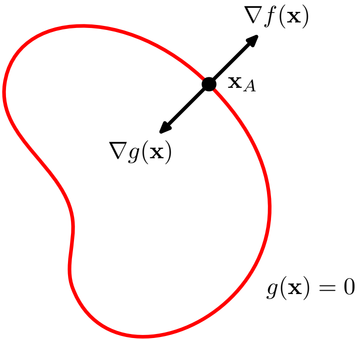
- Assume D-dimensional variable $\mathbf{x} = (x_1, \dots, x_D)^T$.
- The constraint $g(\mathbf{x}) = 0$ is a $(D-1)$-dimensional surface in the $\mathbf{x}$-space.
- The gradient $\nabla g(\mathbf{x})$ will be orthogonal to the surface because if both $g(\mathbf{x} + \boldsymbol{\epsilon})$ and $g(\mathbf{x})$ lie on the surface, then $g(\mathbf{x} + \boldsymbol{\epsilon}) \simeq g(\mathbf{x}) + \boldsymbol{\epsilon}^T \nabla g(\mathbf{x})$.



Lagrange Multipliers

- Assume $\mathbf{x}^\star$ maximises $f(\mathbf{x})$. Then $\nabla f(\mathbf{x}^\star)$ will be orthogonal to the surface. Otherwise we could increase the value by $f(\mathbf{x}^\star + \boldsymbol{\epsilon}) \simeq f(\mathbf{x}^\star) + \boldsymbol{\epsilon}^T \nabla f(\mathbf{x}^\star)$.
- Thus $\nabla f(\mathbf{x}^\star)$ and $\nabla g(\mathbf{x})$ must be parallel (or anti-parallel) and therefore at $\mathbf{x} = \mathbf{x}^\star$ we have with the Lagrange multiplier $\lambda \neq 0$,

$$\nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0.$$



Lagrange Multipliers

- Introduce the Lagrangian function

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

from which we get the constraint stationary conditions

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = \nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0$$

and the constraint itself

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda} = g(\mathbf{x}) = 0.$$

- This are D equations resulting from $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda)$ and one equation from $\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda}$, together determining $\mathbf{x}^{\star}$ and λ.

Lagrange Multipliers - Example

- Given $f(x_1, x_2) = 1 - x_1^2 - x_2^2$ subject to the constraint $g(x_1, x_2) = x_1 + x_2 - 1 = 0$.
- Define the Lagrangian function

$$L(\mathbf{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1).$$

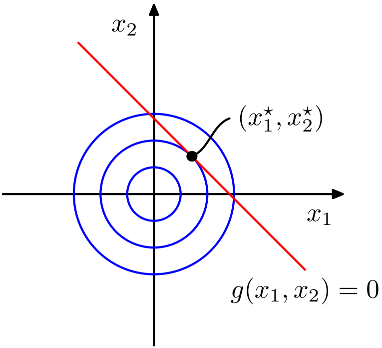
- A stationary solution with respect to x_1, x_2, and λ must satisfy

$$\begin{aligned} -2x_1 + \lambda &= 0 \\ -2x_2 + \lambda &= 0 \\ x_1 + x_2 - 1 &= 0. \end{aligned}$$

- Therefore $(x_1^\star, x_2^\star) = (\frac{1}{2}, \frac{1}{2})$ and $\lambda = 1$.

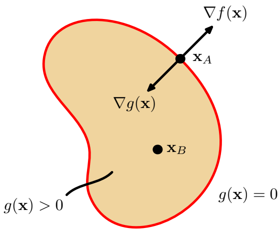
Lagrange Multipliers - Example

- Given $f(x_1, x_2) = 1 - x_1^2 - x_2^2$ subject to the constraint $g(x_1, x_2) = x_1 + x_2 - 1 = 0$.
- Lagrangian $L(\mathbf{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$
- $(x_1^\star, x_2^\star) = (\frac{1}{2}, \frac{1}{2})$.



Lagrange Multipliers for Inequality Constraints

- Inequality constraint $g(\mathbf{x}) \geq 0$.
- Two cases
  1. If $g(\mathbf{x}) > 0$, constraint is inactive. Constraint plays no role. Solution is $\nabla f(\mathbf{x}) = 0$. Corresponds to Lagrangian with $\lambda = 0$.
  2. If $g(\mathbf{x}) = 0$, constraint is active. Solution lies on the boundary, but now the sign of λ is crucial. Only a maximum if its gradient is oriented away from the region $g(\mathbf{x}) > 0$. Therefore, $\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$ for some $\lambda > 0$.
- For either of the cases $\lambda g(\mathbf{x})$ = 0.



Lagrange Multipliers for Inequality Constraints

- Maximise $f(\mathbf{x})$ subject to the constraint $g(\mathbf{x}) \geq 0$.
- Define the Lagrangian

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

- Solve for $\mathbf{x}$ and λ subject to the constraints (Karush-Kuhn-Tucker or KKT conditions)

$$\begin{aligned} g(\mathbf{x}) &\geq 0 \\ \lambda &\geq 0 \\ \lambda g(\mathbf{x}) &= 0 \end{aligned}$$

Lagrange Multipliers for General Case

- Maximise $f(\mathbf{x})$ subject to the constraints $g_j(\mathbf{x}) = 0$ for $j = 1, \dots, J$, and $h_k(\mathbf{x}) \geq 0$ for $k = 1, \dots, K$.
- Define the Lagrange multipliers $\{\lambda_j\}$ and $\{\mu_k\}$, and the Lagrangian

$$L(\mathbf{x}, \{\lambda_j\}, \{\mu_k\}) = f(\mathbf{x}) + \sum_{j=1}^{J} \lambda_j \, g_j(\mathbf{x}) + \sum_{k=1}^{K} \mu_k \, h_k(\mathbf{x}).$$

- Solve for $\mathbf{x}$, $\{\lambda_j\}$, and $\{\mu_k\}$ subject to the constraints (Karush-Kuhn-Tucker or KKT conditions)

$$\begin{aligned} \mu_k &\geq 0 \\ \mu_k \, h_k(\mathbf{x}) &= 0 \end{aligned}$$

  for $k = 1, \dots, K$.
- For minimisation of $f(\mathbf{x})$, change the sign in front of the Lagrange multipliers.