Lecture 3

- Homework

- Gaussian, Bishop 2.3
- Non-parametric, Bishop 2.5
- Linear regression 3.0-3.2

- Pod-cast lecture on-line

- Next lectures:
  - I posted a rough plan.
  - It is flexible though so please come with suggestions

Mark’s KL homework

Mark’s KL homework

Bayes for linear model

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$ $\quad \boldsymbol{n} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{C}_n)$ $\quad \mathbf{y} \sim \mathrm{N}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{C}_n)$ $\quad$ prior: $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{C}_x)$

$p(\boldsymbol{x}|\boldsymbol{y}) \sim p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}) \sim N(\boldsymbol{x}_p, \boldsymbol{C}_p)$

mean $\quad \boldsymbol{x}_p = \boldsymbol{C}_p \boldsymbol{A}^T \boldsymbol{C}_n^{-1} \boldsymbol{y}$

Covariance $\quad \boldsymbol{C}_p^{-1} = \boldsymbol{A}^T \boldsymbol{C}_n^{-1} \boldsymbol{A} + \boldsymbol{C}_x^{-1}$

Bayes' Theorem for Gaussian Variables

- Given

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right)$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}\right)$$

- we have

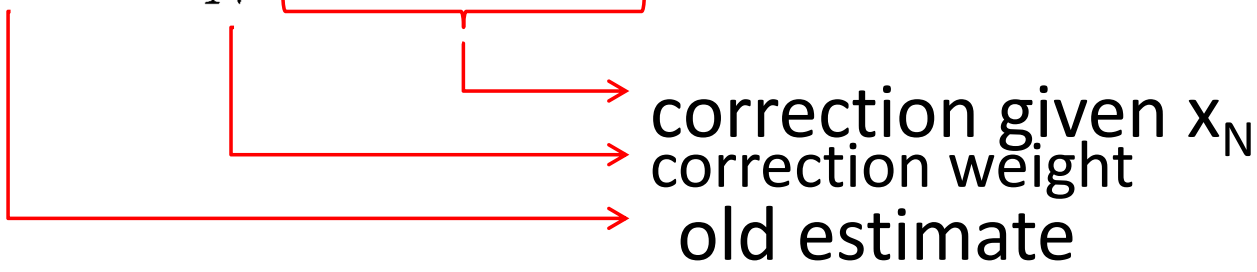
$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b}, \mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}})$$
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma})$$

- where $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}$

Sequential Estimation

Contribution of the Nth data point, x_N

$$\begin{aligned}
\boldsymbol{\mu}_{\mathrm{ML}}^{(N)} &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \\
&= \frac{1}{N}\mathbf{x}_N + \frac{1}{N}\sum_{n=1}^{N-1}\mathbf{x}_n \\
&= \frac{1}{N}\mathbf{x}_N + \frac{N-1}{N}\boldsymbol{\mu}_{\mathrm{ML}}^{(N-1)} \\
&= \boldsymbol{\mu}_{\mathrm{ML}}^{(N-1)} + \frac{1}{N}(\mathbf{x}_N - \boldsymbol{\mu}_{\mathrm{ML}}^{(N-1)})
\end{aligned}$$



Bayesian Inference for the Gaussian Bishop2.3.6

Assume σ^2 is known. Given i.i.d. data $\mathbf{x} = \{x_1, \ldots, x_N\}$ the likelihood function for μ is given by

$$p(\mathbf{x}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2\right\}.$$

- This has a Gaussian shape as a function of μ (but it is *not* a distribution over μ).

Bayesian Inference for the Gaussian Bishop2.3.6

- Combined with a **Gaussian prior over μ,** $p(\mu) = \mathcal{N}\left(\mu|\mu_0, \sigma_0^2\right).$
- this gives the posterior

$$p(\mu|\mathbf{x}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$

$$p(\mu|\mathbf{x}) \propto p(\mathbf{x}|\mu)p(\mu).$$

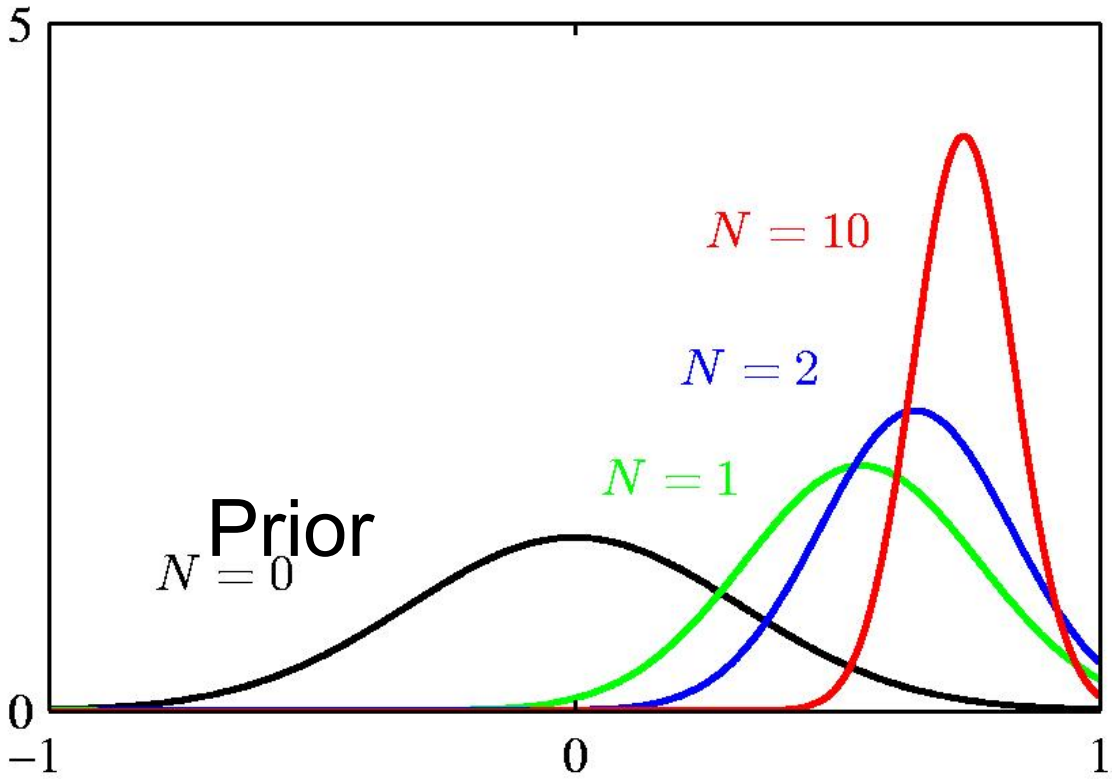
$$\begin{aligned}\mu_N &= \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{\mathrm{ML}}, \qquad \mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \\ \frac{1}{\sigma_N^2} &= \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}.\end{aligned}$$

	$N = 0$	$N \to \infty$
μ_N	μ_0	μ_{ML}
σ_N^2	σ_0^2	0

Bayesian Inference for the Gaussian (3)

- Example: for N = 0, 1, 2 and 10.

$$p(\mu|\mathbf{x}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$



Bayesian Inference for the Gaussian (4)

Sequential Estimation

$$
\begin{aligned}
p(\mu|\mathbf{x}) &\propto p(\mu)p(\mathbf{x}|\mu) \\
&= \left[p(\mu)\prod_{n=1}^{N-1} p(x_n|\mu)\right] p(x_N|\mu) \\
&\propto \mathcal{N}\left(\mu|\mu_{N-1},\sigma^2_{N-1}\right) p(x_N|\mu)
\end{aligned}
$$

The posterior obtained after observing N-1 data points becomes the prior when we observe the N^{th} data point.

Conjugate prior: posterior and prior are in the same family. The **prior** is called a **conjugate prior** for the likelihood function.

Nonparametric Methods (1) Bishop 2.5

- Parametric distribution models (… Gaussian) are restricted to specific forms, which may not always be suitable; for example, consider modelling a multimodal distribution with a single, unimodal model.

- Nonparametric approaches make few assumptions about the overall shape of the distribution being modelled.

- 1000 parameters versus 10 parameters

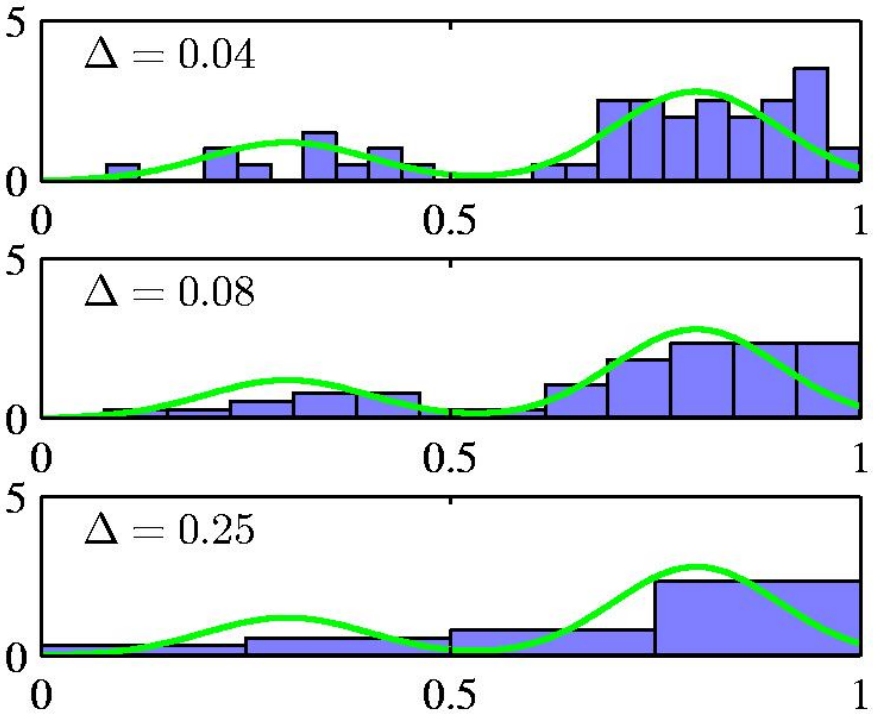
- Nonparametric models (not histograms) requires storing and computing with the entire data set.
- Parametric models, once fitted, are much more efficient in terms of storage and computation.

Nonparametric Methods (2)

Histogram methods partition the data space into distinct bins with widths ¢$_i$ and count the number of observations, n_i, in each bin.

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

- Often, the same width is used for all bins, $\Delta_i = \Delta$.
- Δ acts as a *smoothing* parameter.
- In a D-dimensional space, using M bins in each dimension will require $\mathbf{M^D}$ bins! => it only work for marginals.



Nonparametric Methods (3)

•Assume observations drawn from a density p(x) and consider a small region R containing x such that

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

•The probability that K out of N observations lie inside R is Bin(KjN,P) and if N is large

$$K \simeq NP.$$

If the volume of R, V, is sufficiently small, p(x) is approximately constant over R and

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

Thus

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

V small, yet K>0, therefore N large?

Nonparametric Methods (4)

Kernel Density Estimation: fix V, estimate K from the data. Let R be a hypercube centred on x and define the kernel function (Parzen window)

$$k((\mathbf{x}-\mathbf{x}_n)/h) = \begin{cases} 1, & |(x_i - x_{ni})/h| \leqslant 1/2, \qquad i = 1, \ldots, D, \\ 0, & \text{otherwise.} \end{cases}$$

- It follows that
- $K = \sum_{n=1}^{N} k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right)$ and hence $p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{h^D} k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right).$

Nonparametric Methods (5)

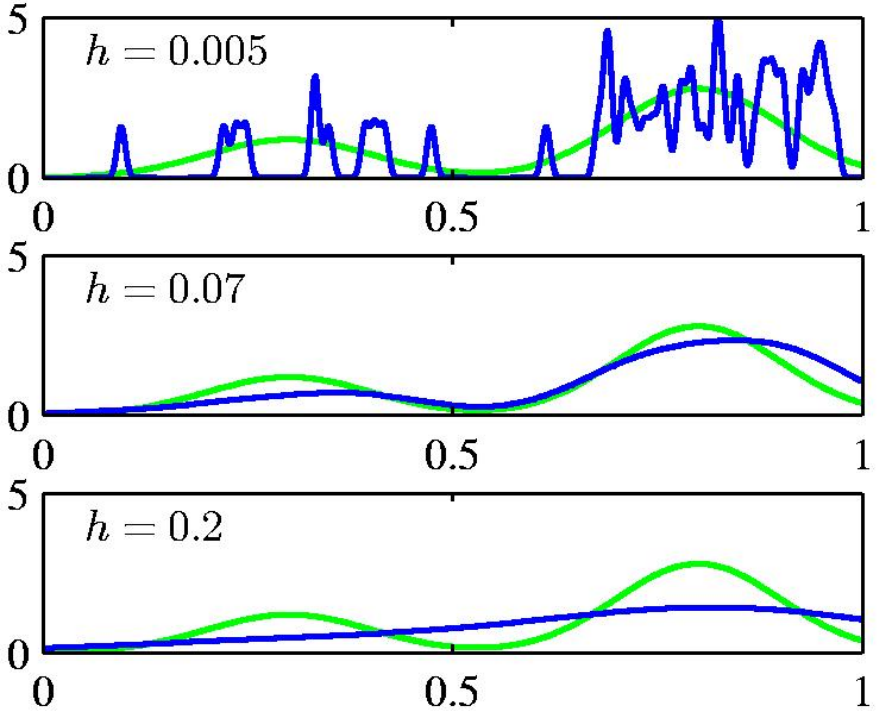
To avoid discontinuities in p(x), use a smooth kernel, e.g. a Gaussian

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

Any kernel such that

$$\begin{aligned} k(\mathbf{u}) &\geqslant 0, \\ \int k(\mathbf{u})\,\mathrm{d}\mathbf{u} &= 1 \end{aligned}$$

will work.

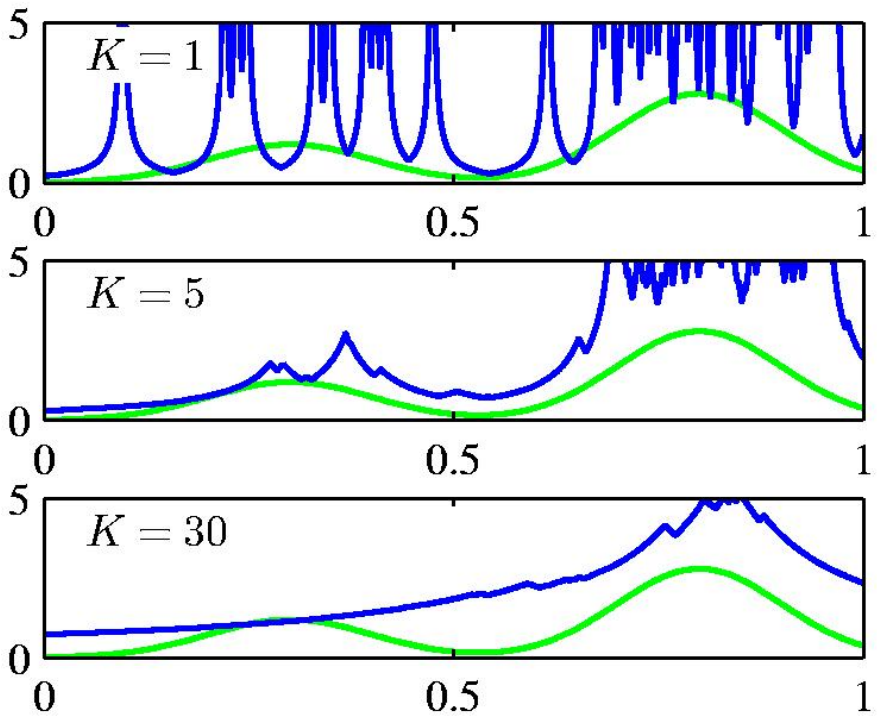


h acts as a smoother.

Nonparametric Methods (6)

Nearest Neighbour Density Estimation: fix K, estimate V from the data. Consider a hypersphere centred on x and let it grow to a volume, V ?, that includes K of the given N data points. Then

$$p(\mathbf{x}) \simeq \frac{K}{NV^{\star}}.$$



K acts as a smoother.

K-Nearest-Neighbours for Classification (1)

- Given a data set with N_k data points from class C_k and $\sum_k N_k = N$, we have

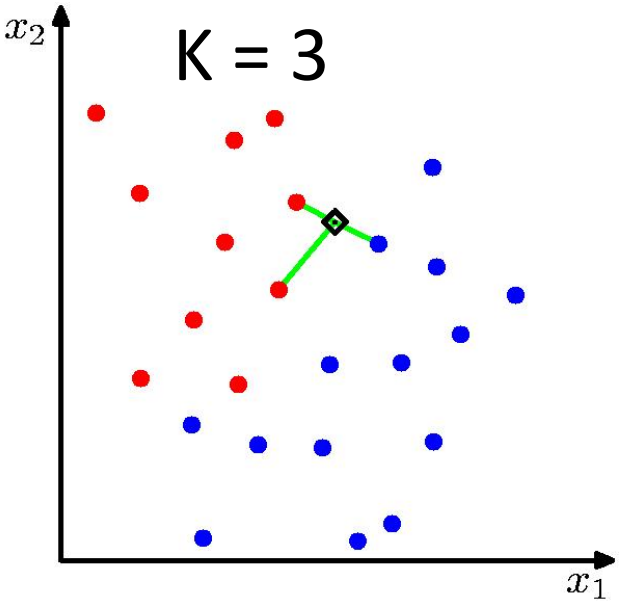
$$p(\mathbf{x}) = \frac{K}{NV}$$

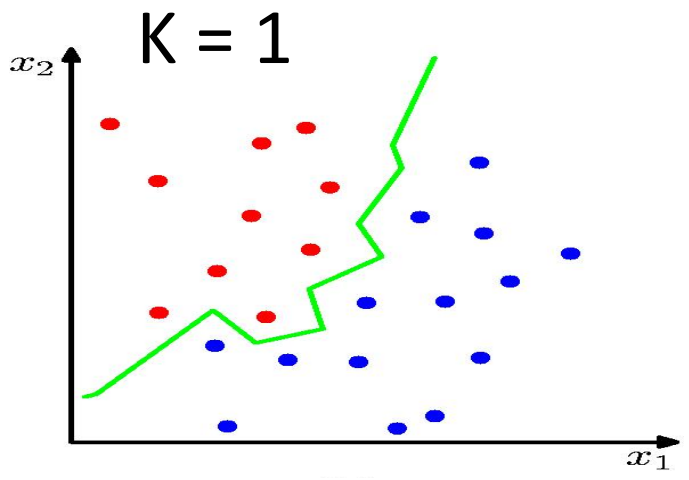
- and correspondingly

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{K_k}{N_k V}.$$

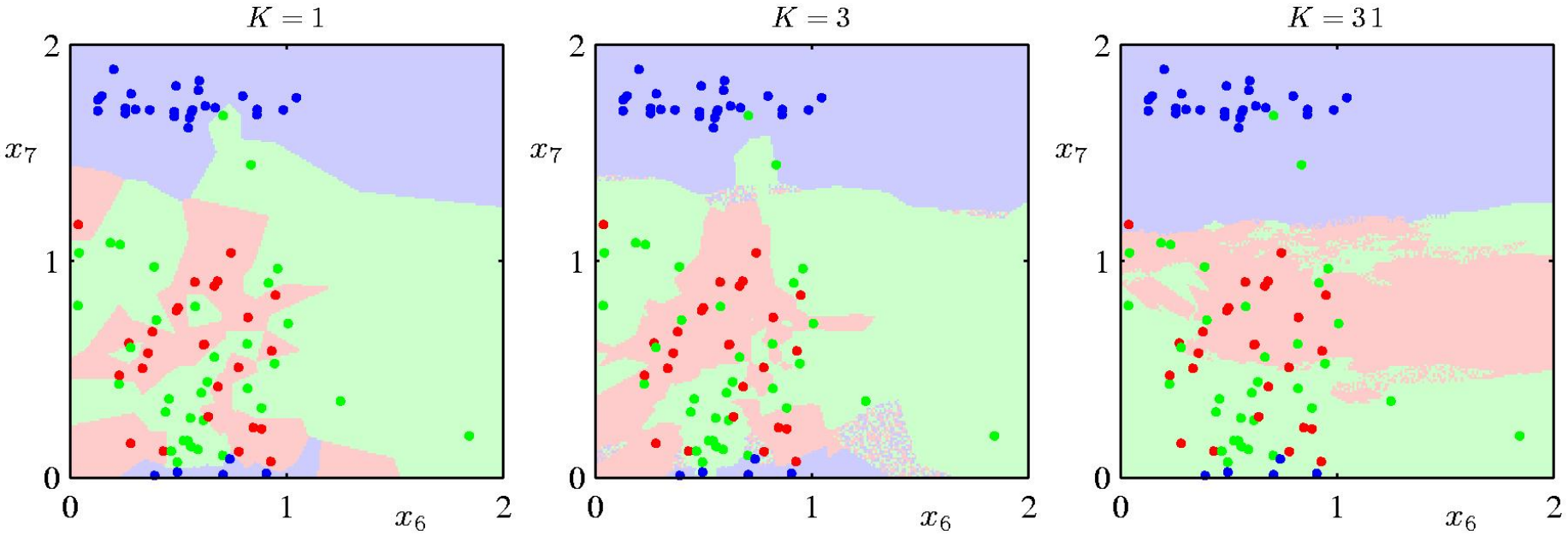
- Since $p(\mathcal{C}_k) = N_k/N$, Bayes' theorem gives

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} = \frac{K_k}{K}.$$





K-Nearest-Neighbours for Classification (3)



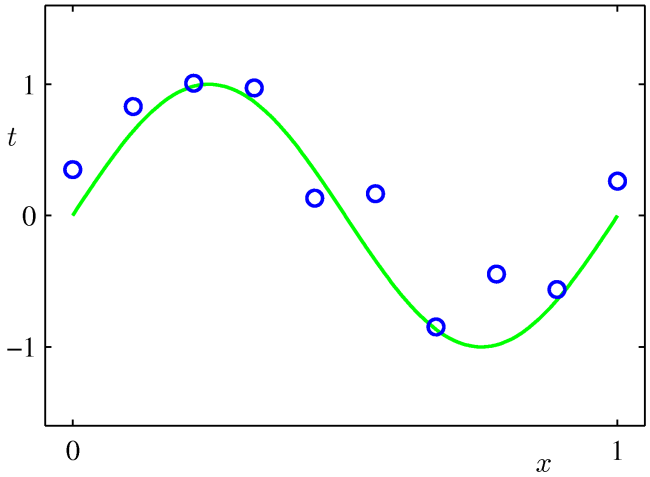
- K acts as a smother
- For $N \to \infty$, the error rate of the nearest-neighbour (K=1) classifier is never more than twice the optimal error (from the true conditional class distributions).

Linear regression: Linear Basis Function Models (1)

Generally

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

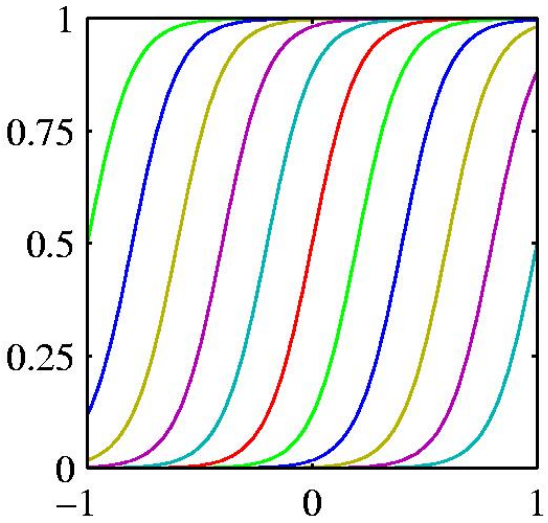
- where $\phi_j(x)$ are known as *basis functions*.
- Typically, $\phi_0(x) = 1$, so that w_0 acts as a bias.
- Simplest case is linear basis functions: $\phi_d(x) = x_d$.



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

http://playground.tensorflow.org/

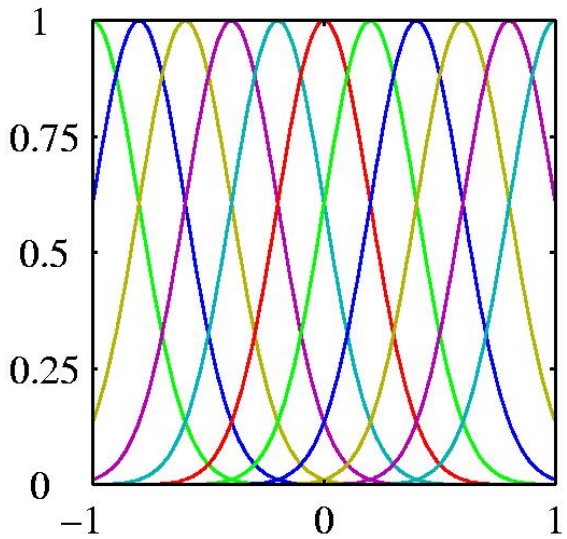
Some types of basis function in 1-D



Sigmoids

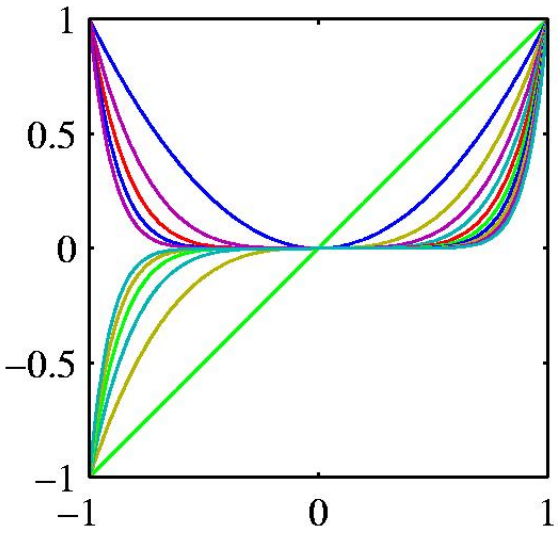
$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

$$\sigma(a) = \frac{1}{1 + \exp(-a)}.$$



Gaussians

$$\phi_j(x) = \exp\left\{-\frac{(x - \mu_j)^2}{2s^2}\right\}$$



Polynomials

$$\phi_j(x) = x^j.$$

Sigmoid and Gaussian basis functions can also be used in multilayer neural networks, but neural networks learn the parameters of the basis functions. This is more powerful but also harder and messier.

Two types of linear model that are equivalent with respect to learning

bias

$$y(\mathbf{x},\mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots = \mathbf{w}^T \mathbf{x}$$

$$y(\mathbf{x},\mathbf{w}) = w_0 + w_1 \phi_1(\mathbf{x}) + w_2 \phi_2(\mathbf{x}) + \ldots = \mathbf{w}^T \Phi(\mathbf{x})$$

- The first and second model has the same number of adaptive coefficients as the number of basis functions +1.
- Once we have replaced the data by basis functions outputs, fitting the second model is exactly the same the first model.
  - No need to clutter math with basis functions

Maximum Likelihood and Least Squares (1)

- Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \quad \text{where} \quad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

- or,

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}).$$

- Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets $\mathbf{t} = [t_1, \ldots, t_N]^{\mathrm{T}}$, we obtain the likelihood function

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

Maximum Likelihood and Least Squares (2)

Taking the logarithm, we get

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{w},\beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1}) \\ &= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w})\end{aligned}$$

Where the sum-of-squares error is

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

Maximum Likelihood and Least Squares (3)

Computing the gradient and setting it to zero yields

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

Solving for w,

where

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}$$

The Moore-Penrose pseudo-inverse, $\boldsymbol{\Phi}^{\dagger}$.

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

Maximum Likelihood and Least Squares (4)

Maximizing with respect to the bias, w_0, alone,

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \overline{\phi_j}$$

$$= \frac{1}{N}\sum_{n=1}^{N} t_n - \sum_{j=1}^{M-1} w_j \frac{1}{N}\sum_{n=1}^{N} \phi_j(\mathbf{x}_n).$$

We can also maximize with respect to β, giving

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

Geometry of Least Squares

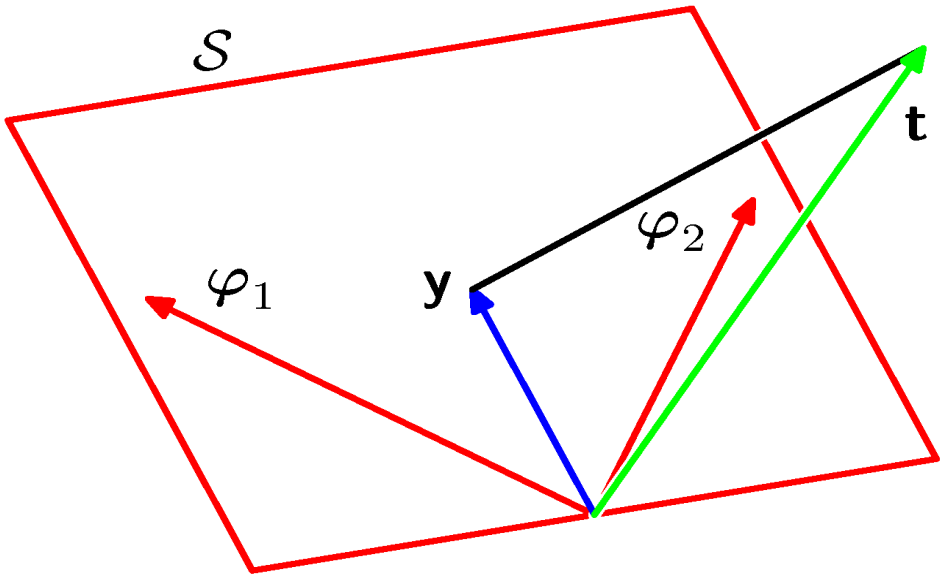
Consider

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}_{\mathrm{ML}} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M]\,\mathbf{w}_{\mathrm{ML}}.$$

$$\mathbf{y} \in \mathcal{S} \subseteq \mathcal{T} \qquad \mathbf{t} \in \mathcal{T}$$

N-dimensional

M-dimensional



S is spanned by $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M$

w_{ML} minimizes the distance between t and its orthogonal projection on S, i.e. y.