

- Homework

✓ • Gaussian, Bishop 2.3

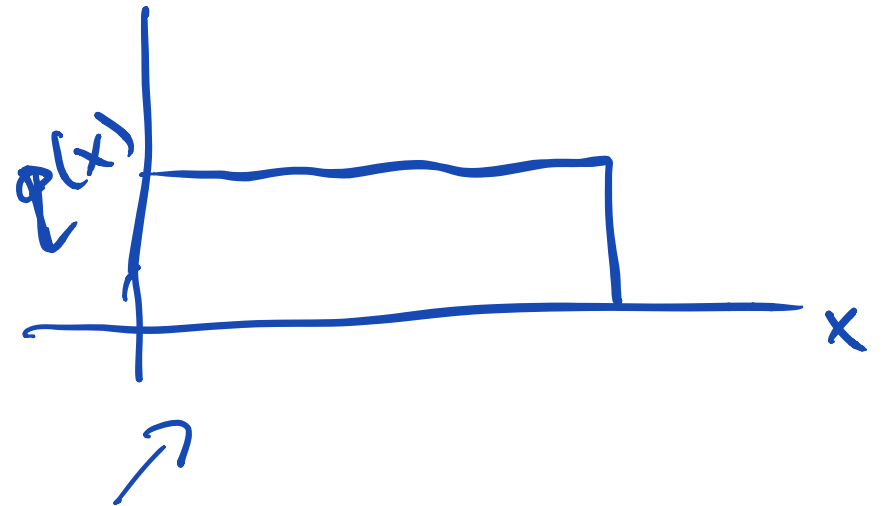
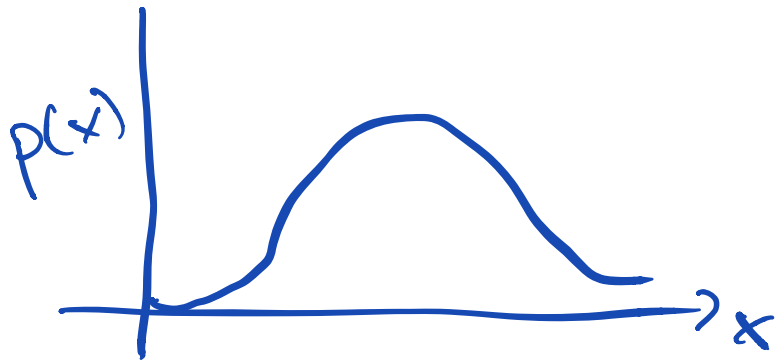
- Non-parametric, Bishop 2.5
- Linear regression 3.0-3.2

✓ • Pod-cast lecture on-line

- Next lectures:

✓ • I posted a rough plan.

– It is flexible though so please come with suggestions

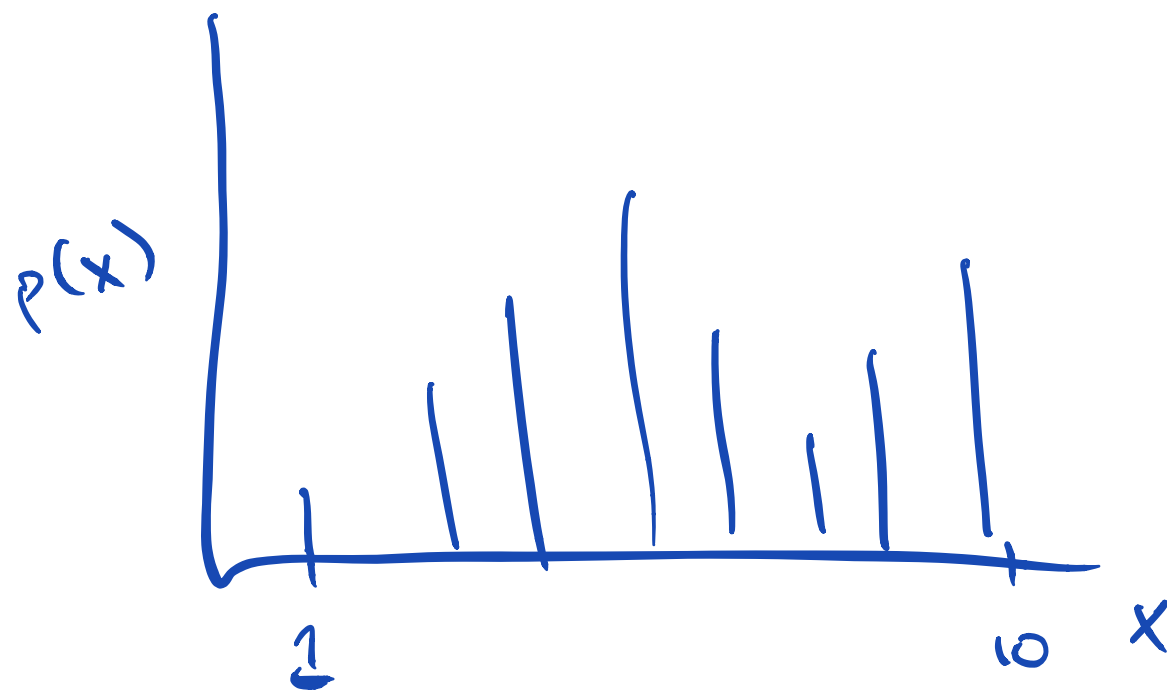


Entropy

$$H = - \sum_{i=1}^N p(x_i) \log_2 p(x_i)$$

→ discrete
↑

$$KL(p \parallel q) = - \sum_{i=1}^N p(x_i) \left(\log_2 p(x_i) - \log_2 q(x_i) \right)$$



KL div is NOT a distance metri

$$\underline{KL(p \parallel q) \neq KL(q \parallel p)} \quad \frac{p}{q} \neq \frac{q}{p}$$

Bayes for linear model

$$y = Ax + n \quad n \sim N(0, C_n) \quad y \sim N(\underline{Ax}, C_n) \quad \text{prior: } x \sim N(0, C_x)$$

$$p(x|y) \sim p(y|x)p(x) \sim N(x_p, C_p)$$

mean

$$x_p = C_p A^T C_n^{-1} y$$

$$C_p^{-1} = A^T C_n^{-1} A + C_x^{-1}$$

$$\sim e^{-\frac{1}{2}(x-x_p)^T C_p^{-1} (x-x_p)} \leftarrow$$

Covariance

$$= e^{-\frac{1}{2}(y-Ax)^T C_n^{-1} (y-Ax)} e^{-\frac{1}{2}x^T C_x^{-1} x}$$

$$= e^{-\frac{1}{2}(x^T A^T C_n^{-1} A x + x^T C_x^{-1} x)} e^{-\frac{1}{2}x^T A^T C_n^{-1} y}$$

$$\underbrace{\hspace{10em}}_{x^T C_p^{-1} x} \quad \underbrace{\hspace{10em}}_{x^T C_p^{-1} x_p}$$

$$C_p^{-1} = A^T C_n^{-1} A + C_x^{-1}$$

$$x_p = C_p A^T C_n^{-1} y$$

Bayes' Theorem for Gaussian Variables

- Given
$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \underline{\mu}, \underline{\Lambda}^{-1})$$
$$p(\mathbf{y} | \mathbf{x}) = \mathcal{N}(\mathbf{y} | \mathbf{A}\mathbf{x} + \underline{\mathbf{b}}, \mathbf{L}^{-1})$$
- we have
$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | \mathbf{A}\underline{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\underline{\Lambda}^{-1}\mathbf{A}^T)$$
$$p(\mathbf{x} | \mathbf{y}) = \mathcal{N}(\mathbf{x} | \underline{\Sigma}\{\mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b}) + \underline{\Lambda}\underline{\mu}\}, \underline{\Sigma})$$
- where $\underline{\Sigma} = (\underline{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}$

Sequential Estimation

Contribution of the N^{th} data point, \mathbf{x}_N

$$\mu_{\text{ML}}^{(N)} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \quad \checkmark$$

$$= \frac{1}{N} \mathbf{x}_N + \frac{1}{N} \sum_{n=1}^{N-1} \mathbf{x}_n$$

$$= \frac{1}{N} \mathbf{x}_N + \frac{N-1}{N} \mu_{\text{ML}}^{(N-1)}$$

$$= \mu_{\text{ML}}^{(N-1)} + \frac{1}{N} (\mathbf{x}_N - \mu_{\text{ML}}^{(N-1)})$$

correction given \mathbf{x}_N
correction weight
old estimate

Bayesian Inference for the Gaussian Bishop2.3.6

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

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

- 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** , $p(\mu) = \mathcal{N}(\mu | \mu_0, \sigma_0^2)$.
- this gives the posterior

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

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

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2} \mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2} \mu_{\text{ML}},$$

$$\mu_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N x_n$$

$$\frac{(\mu - \mu_N)^2}{2\sigma_N^2} = \frac{1}{2\sigma^2} \sum_n^N (x_n - \mu)^2 + \frac{(\mu - \mu_0)^2}{2\sigma_0^2}$$

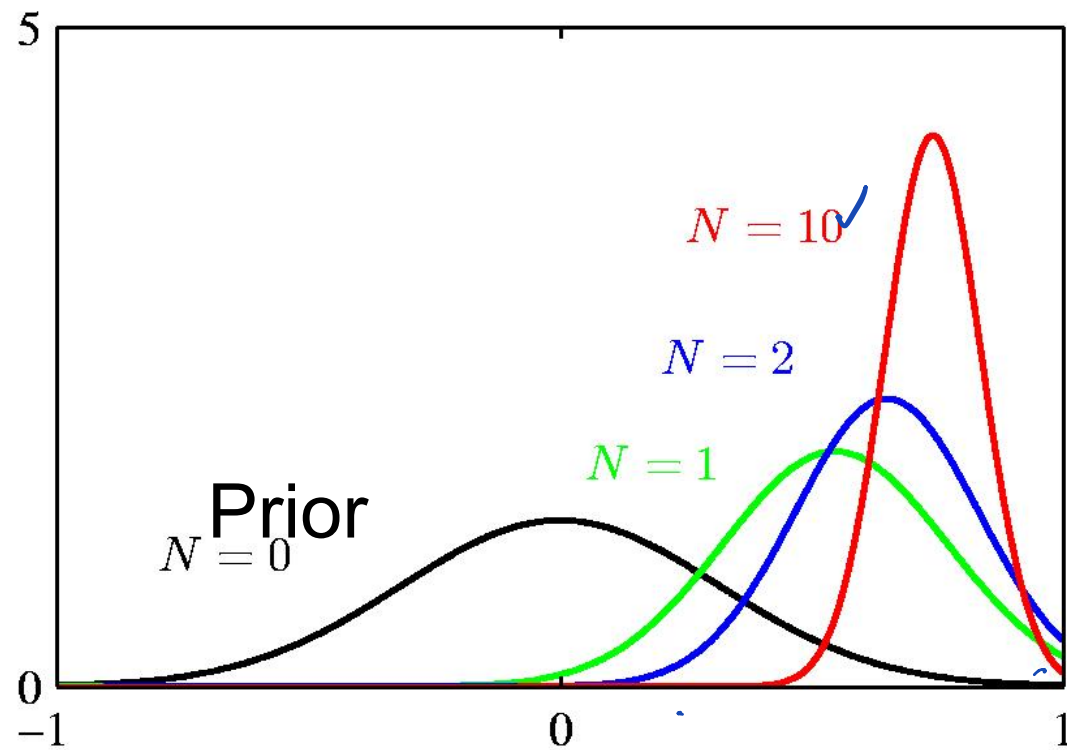
$$\frac{\mu^2}{\sigma_N^2} = \frac{1}{\sigma^2} N \mu^2 + \frac{\mu^2}{\sigma_0^2}$$

	$N = 0$	$N \rightarrow \infty$
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 .

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



Bayesian Inference for the Gaussian (4)

Sequential Estimation

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

The posterior obtained after observing N-1 data points becomes the prior when we observe the Nth 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

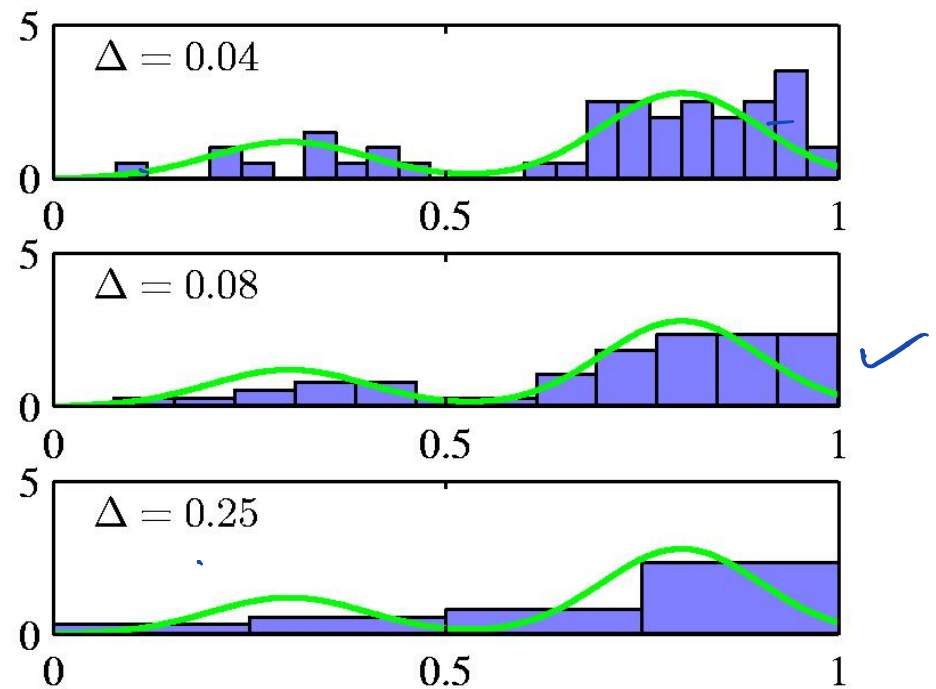
- Parametric distribution models (... Gaussian) are restricted to specific forms, which may not always be suitable; for example, consider modelling a multimodal distribution with a single, unimodal model.
- Nonparametric approaches make few assumptions about the overall shape of the distribution being 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 Δ_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 $p(\mathbf{x})$ and consider a small region R containing \mathbf{x} such that

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

If the volume of R , V , is sufficiently small, $p(\mathbf{x})$ is approximately constant over R and

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

Thus

- The probability that K out of N observations lie inside R is $\text{Bin}(K|N, P)$ and if N is large

$$K \simeq NP.$$

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

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 \mathbf{x} and define the kernel function (Parzen window)

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

- 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)

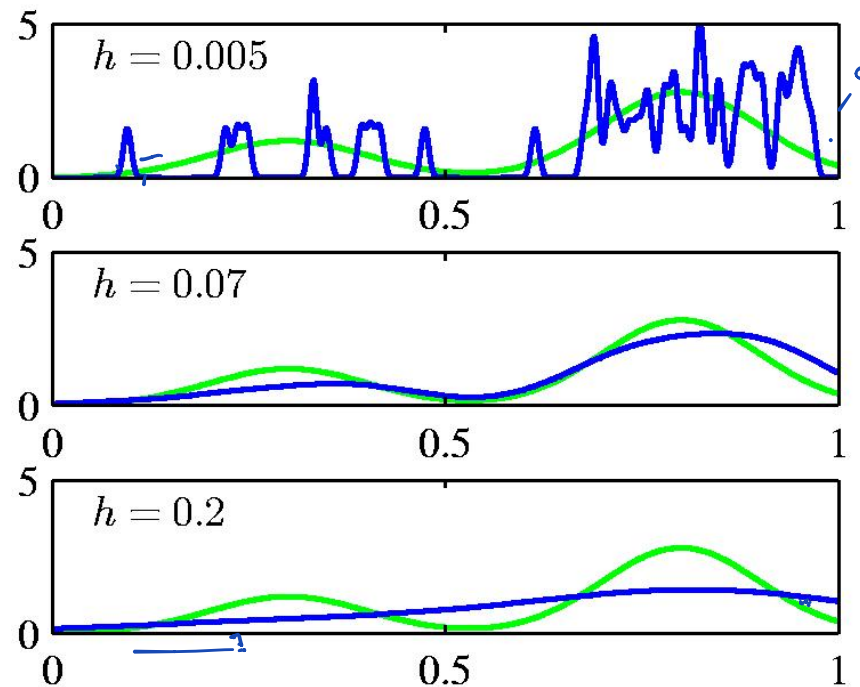
To avoid discontinuities in $p(\mathbf{x})$,
use a smooth kernel, e.g. a
Gaussian

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

Any kernel such that

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

will work.



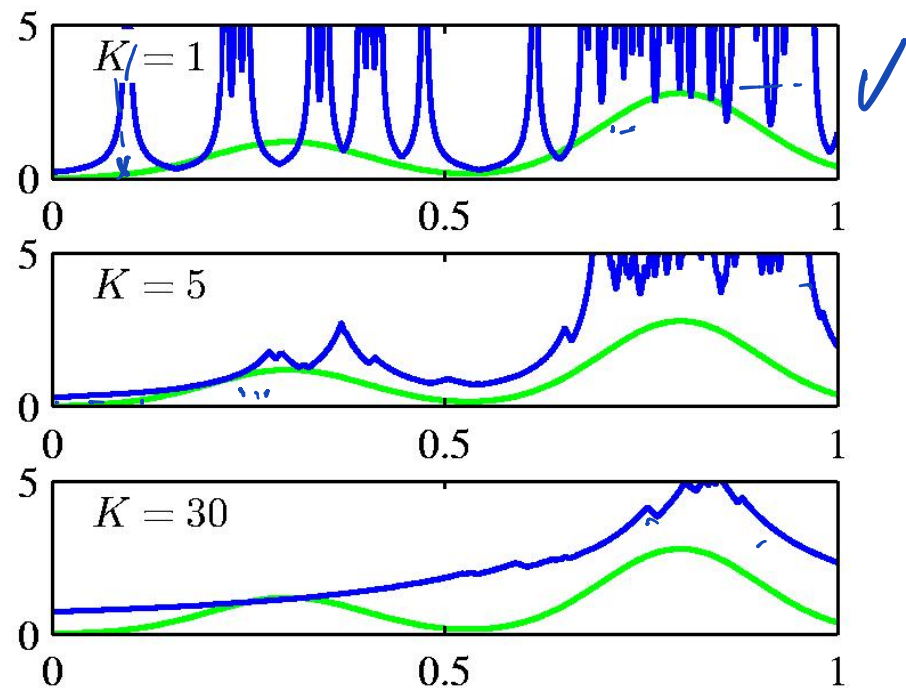
h acts as a smoother.

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

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



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

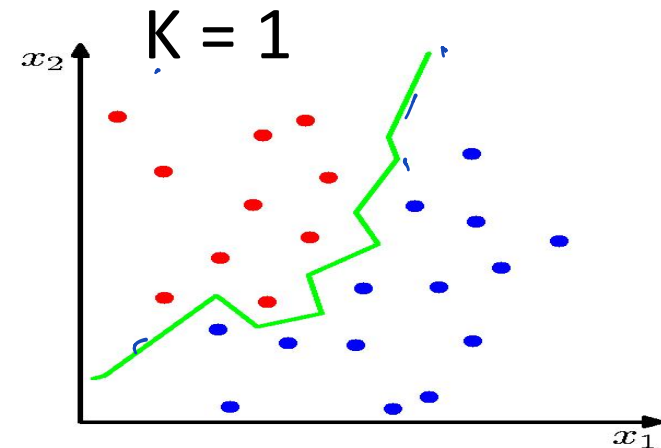
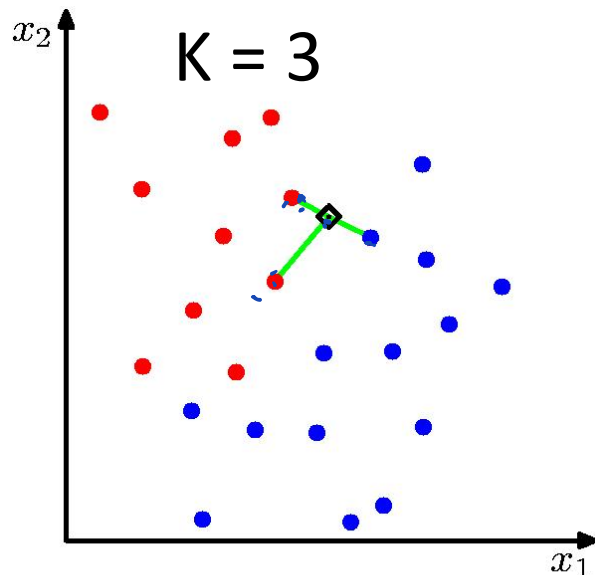
$$p(\mathbf{x}) = \frac{K}{NV}$$

- and correspondingly

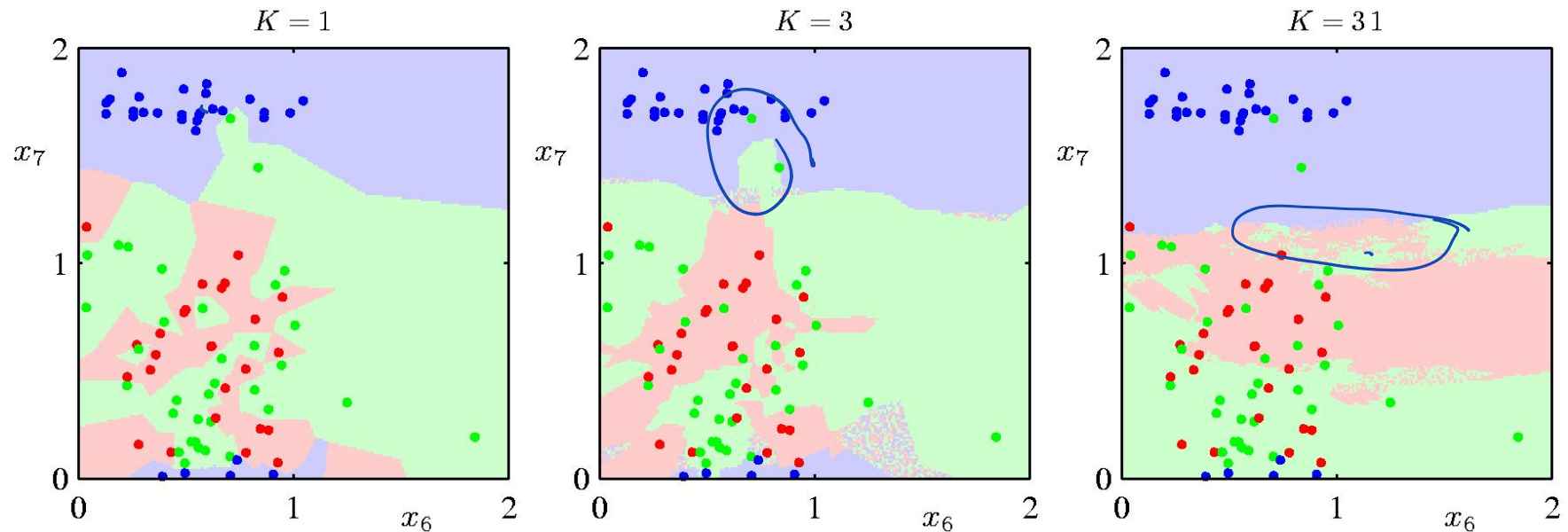
$$p(\mathbf{x}|C_k) = \frac{K_k}{N_k V}$$

- Since $p(C_k) = N_k/N$, Bayes' theorem gives

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} = \frac{K_k}{K}$$



K-Nearest-Neighbours for Classification (3)



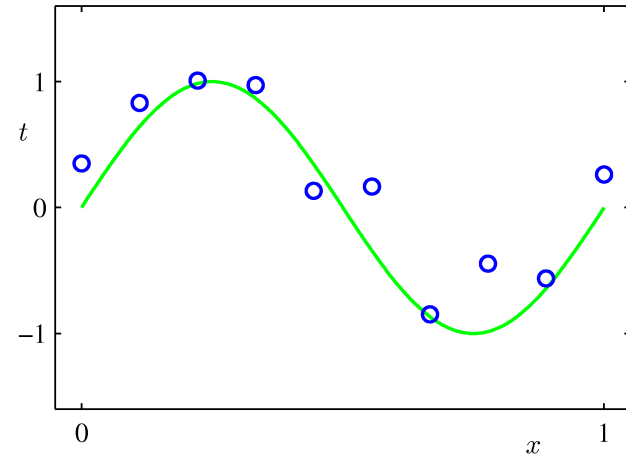
- K acts as a smoother
- For $N \rightarrow \infty$, the error rate of the nearest-neighbour ($K=1$) classifier is never more than twice the optimal error (from the true conditional class distributions).

Linear regression: Linear Basis Function Models (1)

Generally

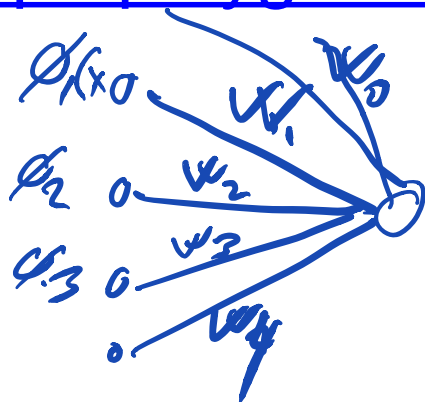
$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

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

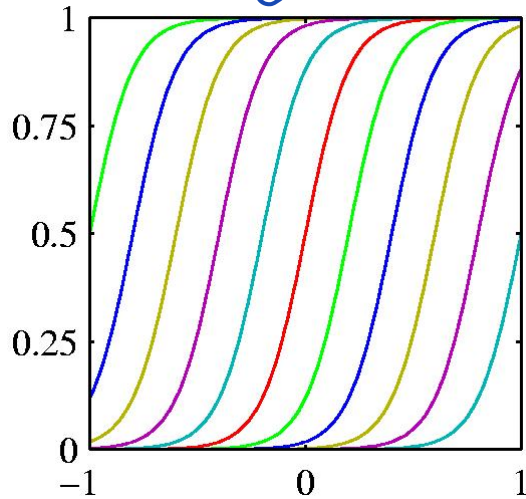


$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

<http://playground.tensorflow.org/>

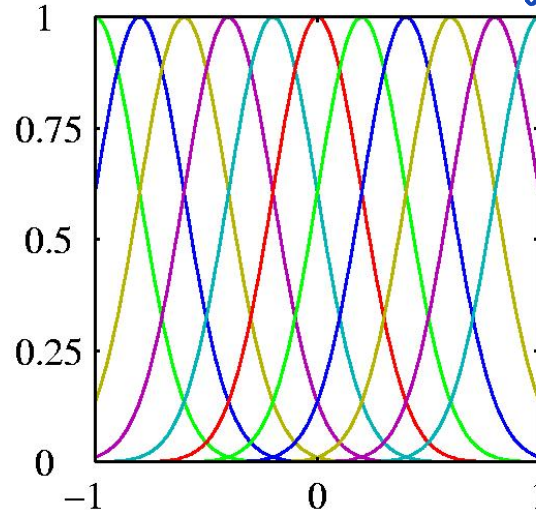


Some types of basis function in 1-D



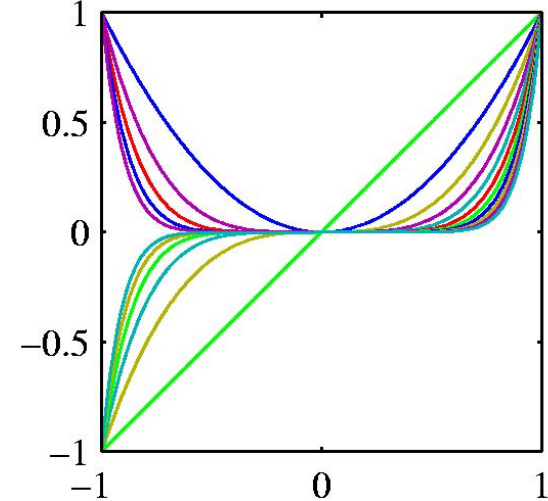
Sigmoids

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$
$$\sigma(a) = \frac{1}{1 + \exp(-a)}.$$



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

$$y(\mathbf{x}, \mathbf{w}) = \overset{\text{bias}}{\underline{w_0}} + w_1 x_1 + w_2 x_2 + \dots = \underline{\mathbf{w}}^T \mathbf{x} \quad \checkmark$$

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

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

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

Maximum Likelihood and Least Squares (2)

Taking the logarithm, we get

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

Where the sum-of-squares error is

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

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{w}} &= \sum_n \phi_n (t_n - \mathbf{w}^T \phi_n) \\ &= \sum_n \phi_n t_n - \sum_n \phi_n \mathbf{w}^T \phi_n = 0\end{aligned}$$

$$\Phi = \begin{bmatrix} -\phi_1^T \\ -\phi_2^T \\ \vdots \\ -\phi_N^T \end{bmatrix}$$

$$\underline{\Phi}^T \underline{t} - \underline{\Phi}^T \underline{\Phi} \underline{w} = 0 \Rightarrow \underline{w} = \underline{(\Phi^T \Phi)^{-1} \Phi^T t}$$

Maximum Likelihood and Least Squares (3)

Computing the gradient and setting it to zero yields

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

Solving for \mathbf{w} ,

where

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

The Moore-Penrose pseudo-inverse, Φ^\dagger .

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

Maximum Likelihood and Least Squares (4)

Maximizing with respect to the bias, w_0 , alone,

$$\begin{aligned} w_0 &= \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j \\ &= \frac{1}{N} \sum_{n=1}^N t_n - \sum_{j=1}^{M-1} w_j \frac{1}{N} \sum_{n=1}^N \phi_j(\mathbf{x}_n). \end{aligned}$$

We can also maximize with respect to β , giving

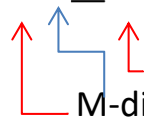
$$\frac{1}{\beta_{\text{ML}}} = \frac{1}{N} \sum_{n=1}^N \{t_n - \mathbf{w}_{\text{ML}}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

Geometry of Least Squares

Consider

$$\mathbf{y} = \Phi \mathbf{w}_{\text{ML}} = [\varphi_1, \dots, \varphi_M] \mathbf{w}_{\text{ML}}.$$

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

 M-dimensional
N-dimensional

\mathcal{S} is spanned by

$$\varphi_1, \dots, \varphi_M$$

\mathbf{w}_{ML} minimizes the distance between \mathbf{t} and its orthogonal projection on \mathcal{S} , i.e. \mathbf{y} .

