STA 4273H: Statistical Machine Learning

Russ Salakhutdinov

Department of Statistics and Computer Science
rsalakhu@utstat.toronto.edu
http://www.utstat.utoronto.ca/~rsalakhu/
Sidney Smith Hall, Room 6002

Lecture 2

Last Class

- In our last class, we looked at:
  - Statistical Decision Theory
  - Linear Regression Models
  - Linear Basis Function Models
  - Regularized Linear Regression Models
  - Bias-Variance Decomposition

- We will now look at the Bayesian framework and Bayesian Linear Regression Models.

Bayesian Approach

- We formulate our knowledge about the world probabilistically:
  - We define the model that expresses our knowledge qualitatively (e.g. independence assumptions, forms of distributions).
  - Our model will have some unknown parameters.
  - We capture our assumptions, or prior beliefs, about unknown parameters (e.g. range of plausible values) by specifying the prior distribution over those parameters before seeing the data.
- We observe the data.
- We compute the posterior probability distribution for the parameters, given observed data.
- We use this posterior distribution to:
  - Make predictions by averaging over the posterior distribution
  - Examine/Account for uncertainly in the parameter values.
  - Make decisions by minimizing expected posterior loss.

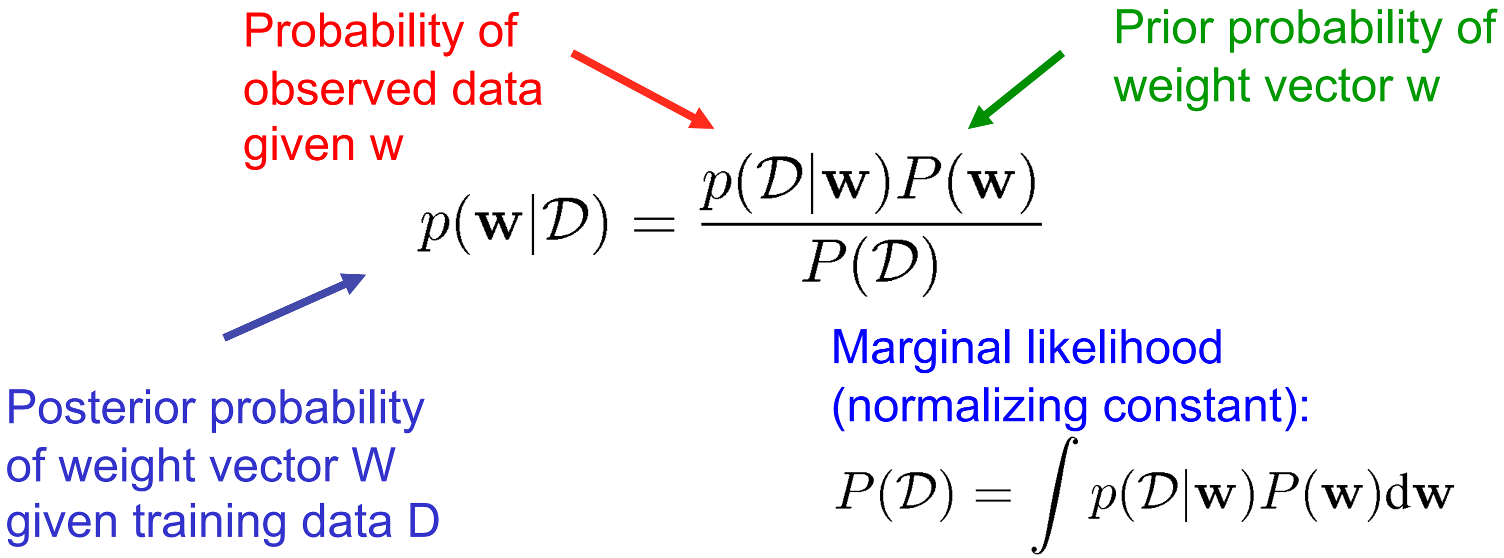
(See Radford Neal's NIPS tutorial on ``Bayesian Methods for Machine Learning'')

Posterior Distribution

- The posterior distribution for the model parameters can be found by combining the prior with the likelihood for the parameters given the data.

- This is accomplished using Bayes’ Rule:

$$P(\text{parameters} \mid \text{data}) = \frac{P(\text{data} \mid \text{parameters}) P(\text{parameters})}{P(\text{data})}$$



Probability of observed data given w

Prior probability of weight vector w

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}$$

Posterior probability of weight vector W given training data D

Marginal likelihood (normalizing constant):

$$P(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})P(\mathbf{w})\mathrm{d}\mathbf{w}$$

This integral can be high-dimensional and is often difficult to compute.

The Rules of Probability

Sum Rule:

$$p(X) = \sum_{Y} p(X, Y)$$

Product Rule:

$$p(X, Y) = p(Y|X)p(X)$$

Predictive Distribution

- We can also state Bayes' rule in words:

$$\text{posterior} \propto \text{likelihood} \times \text{prior}.$$

- We can make predictions for a new data point **x***, given the training dataset by integrating over the posterior distribution:

$$p(\mathbf{x}^*|\mathcal{D}) = \int p(\mathbf{x}^*|\mathbf{w},\mathcal{D})p(\mathbf{w}|\mathcal{D})\mathrm{d}\mathbf{w} = \mathbb{E}_{P(\mathbf{w}|\mathcal{D})}\left[p(\mathbf{x}^*|\mathbf{w},\mathcal{D})\right],$$

which is sometimes called predictive distribution.

- Note that computing predictive distribution requires knowledge of the posterior distribution:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}, \quad \text{where } P(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})P(\mathbf{w})\mathrm{d}\mathbf{w}$$

which is usually intractable.

Modeling Challenges

- The first challenge is in specifying suitable model and suitable prior distributions. This can be challenging particularly when dealing with high-dimensional problems we see in machine learning.

  - A suitable model should admit all the possibilities that are thought to be at all likely.
  - A suitable prior should avoid giving zero or very small probabilities to possible events, but should also avoid spreading out the probability over all possibilities.

- We may need to properly model dependencies between parameters in order to avoid having a prior that is too spread out.

- One strategy is to introduce latent variables into the model and hyperparameters into the prior.

- Both of these represent the ways of modeling dependencies in a tractable way.

Computational Challenges

The other big challenge is computing the posterior distribution. There are several main approaches:

- Analytical integration: If we use “conjugate” priors, the posterior distribution can be computed analytically. Only works for simple models and is usually too much to hope for.
- Gaussian (Laplace) approximation: Approximate the posterior distribution with a Gaussian. Works well when there is a lot of data compared to the model complexity (as posterior is close to Gaussian).
- Monte Carlo integration: Once we have a sample from the posterior distribution, we can do many things. The dominant current approach is Markov Chain Monte Carlo (MCMC) -- simulate a Markov chain that converges to the posterior distribution. It can be applied to a wide variety of problems.
- Variational approximation: A cleverer way to approximate the posterior. It often works much faster compared to MCMC. But often not as general as MCMC.

Bayesian Linear Regression

- Given observed inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, and corresponding target values $\mathbf{t} = [t_1, t_2, ..., t_N]^T$, we can write down the likelihood function:

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

where $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), ..., \phi_{M-1}(\mathbf{x}))^T$ represent our basis functions.

- The corresponding conjugate prior is given by a Gaussian distribution:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0).$$

- As both the likelihood and the prior terms are Gaussians, the posterior distribution will also be Gaussian.

- If the posterior distributions p(θ|x) are in the same family as the prior probability distribution p(θ), the prior and posterior are then called **conjugate distributions**, and the prior is called a **conjugate prior** for the likelihood.

Bayesian Linear Regression

- Combining the prior together with the likelihood term:

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \mathbf{w}, \beta) \propto \left[\prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \right] \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0).$$

- The posterior (with a bit of manipulation) takes the following Gaussian form:

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

where

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}. \end{aligned}$$

- The posterior mean can be expresses in terms of the least-squares estimator and the prior mean:

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi} \mathbf{w}_{ML} \right).$$

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

- As we increase our prior precision (decrease prior variance), we place greater weight on the prior mean relative the data.

Bayesian Linear Regression

- Consider a zero mean isotropic Gaussian prior, which is govern by a single precision parameter α:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

for which the posterior is Gaussian with:

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

$$\begin{aligned}\mathbf{m}_N &= \beta\mathbf{S}_N\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}.\end{aligned}$$

- If we consider an infinitely broad prior, $\alpha \to 0$, the mean $\mathbf{m}_\mathbf{N}$ of the posterior distribution reduces to maximum likelihood value $\mathbf{w}_\mathbf{ML}$.

- The log of the posterior distribution is given by the sum of the log-likelihood and the log of the prior:

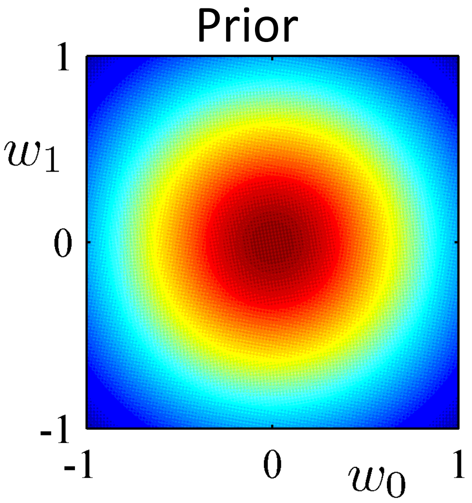
$$\ln p(\mathbf{w}|\mathcal{D}) = -\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \text{const.}$$

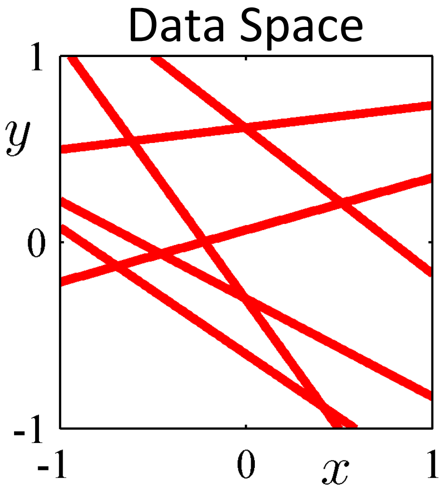
- Maximizing this posterior with respect to **w** is equivalent to minimizing the sum-of-squares error function with a quadratic regulation term λ = α / β.

Bayesian Linear Regression

- Consider a linear model of the form: $y(x, \mathbf{w}) = w_0 + w_1 x.$
- The training data is generated from the function $f(x, \mathbf{a}) = a_0 + a_1 x$ with $a_0 = 0.3; a_1 = 0.5,$ by first choosing x_n uniformly from [-1;1], evaluating $f(x, \mathbf{a})$, and adding a small Gaussian noise.
- Goal: recover the values of a_0, a_1 from such data.

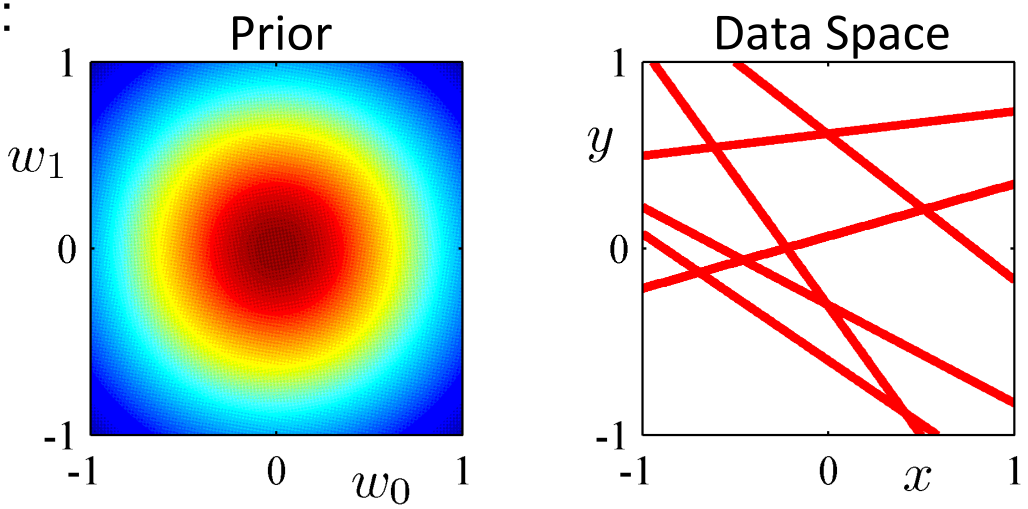
0 data points are observed:



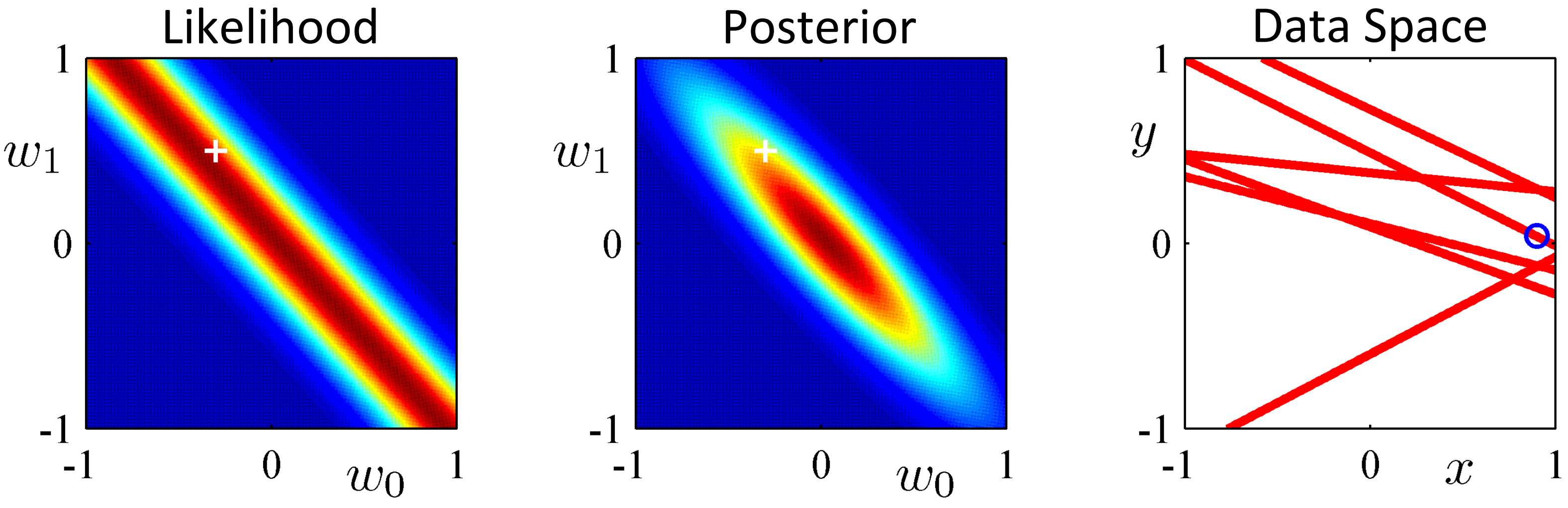


Bayesian Linear Regression

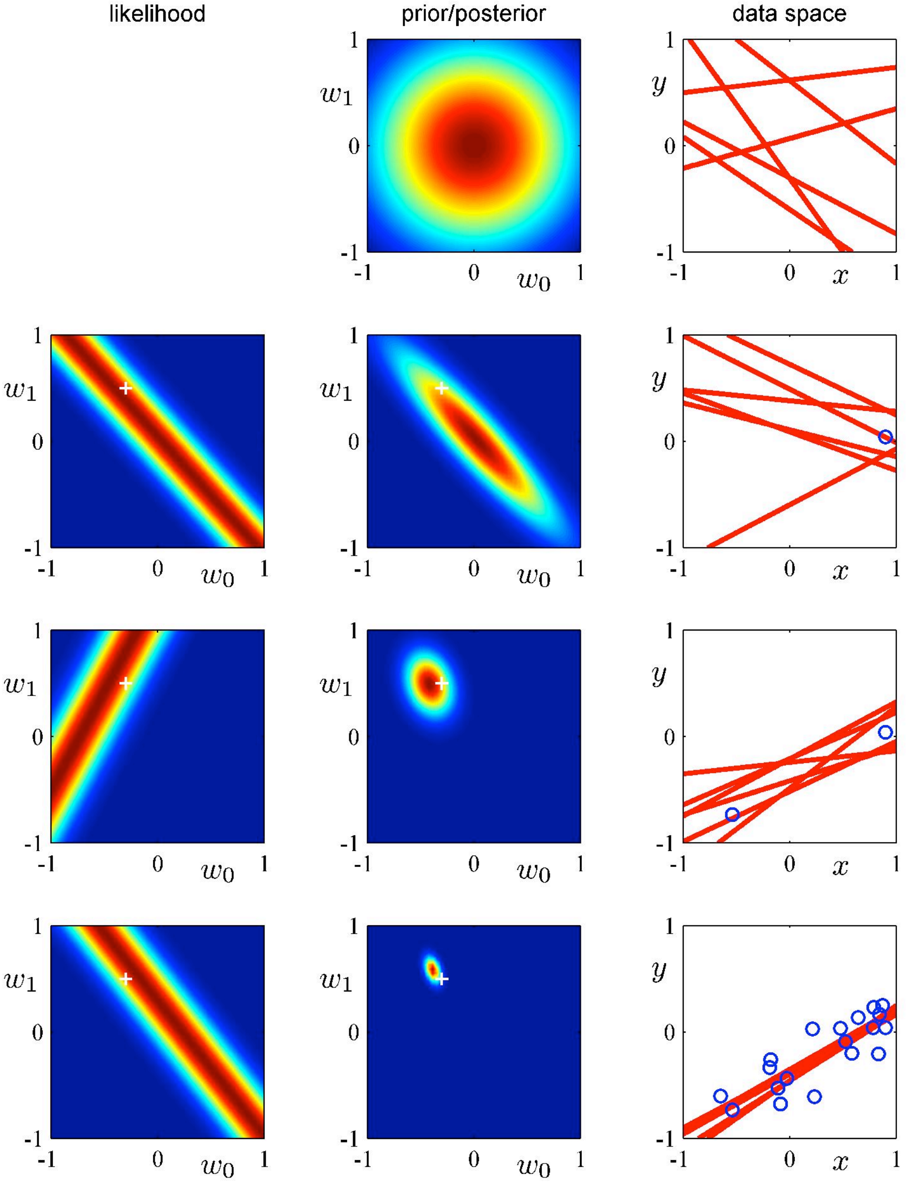
0 data points are observed:



1 data point is observed:



Bayesian Linear Regression



0 data points are observed.

1 data point is observed.

2 data points are observed.

20 data points are observed.

Predictive Distribution

- We can make predictions for a new input vector **x** by integrating over the posterior distribution:

$$\begin{aligned} p(t|\mathbf{t},\mathbf{x},\mathbf{X},\alpha,\beta) &= \int p(t|\mathbf{x},\mathbf{w},\beta)p(\mathbf{w}|\mathbf{t},\mathbf{X},\alpha,\beta)\mathrm{d}\mathbf{w} \\ &= \mathcal{N}\left(t|\mathbf{m}_N^T\boldsymbol{\phi}(\mathbf{x}),\sigma_N^2(\mathbf{x})\right), \end{aligned}$$

where

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}).$$

$$\boxed{\begin{aligned} \mathbf{m}_N &= \beta\mathbf{S}_N\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}. \end{aligned}}$$

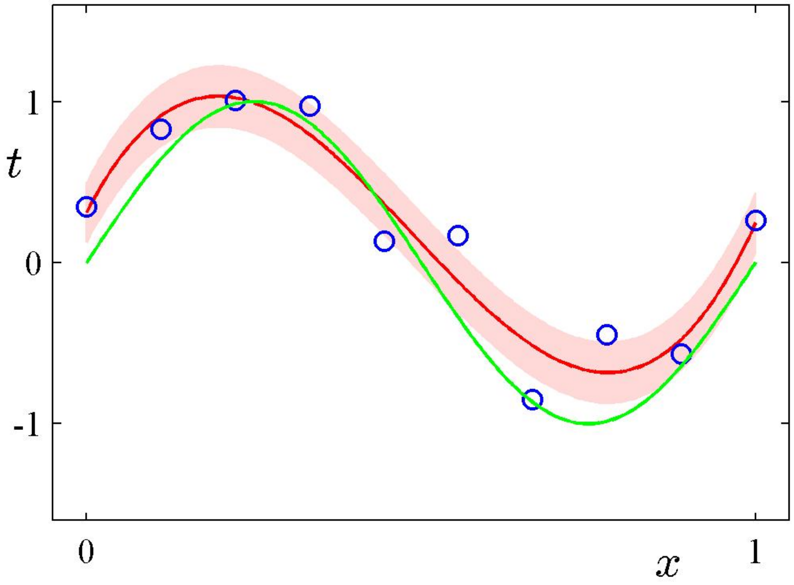
Noise in the target values

Uncertainly associated with parameter values.

- In the limit, as N $\rightarrow \infty$, the second term goes to zero.
- The variance of the predictive distribution arises only from the additive noise governed by parameter β.

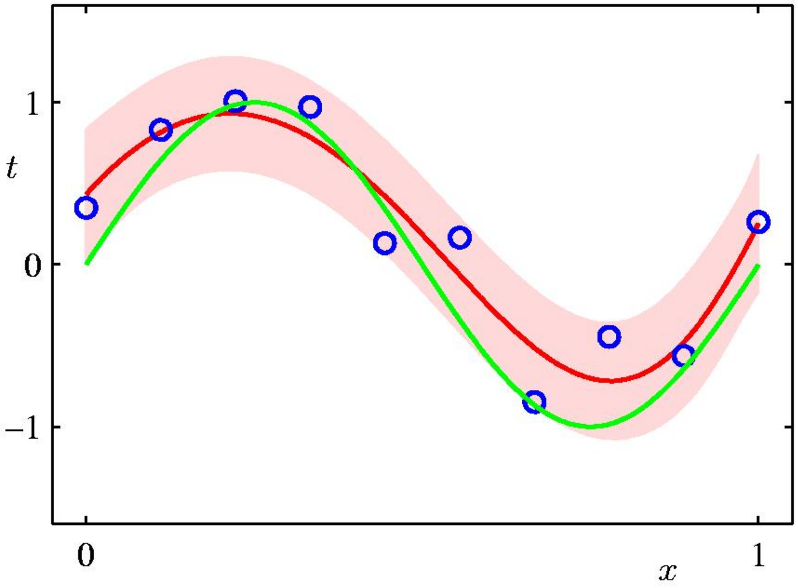
Predictive Distribution: Bayes vs. ML

Predictive distribution based on maximum likelihood estimates



$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right)$$

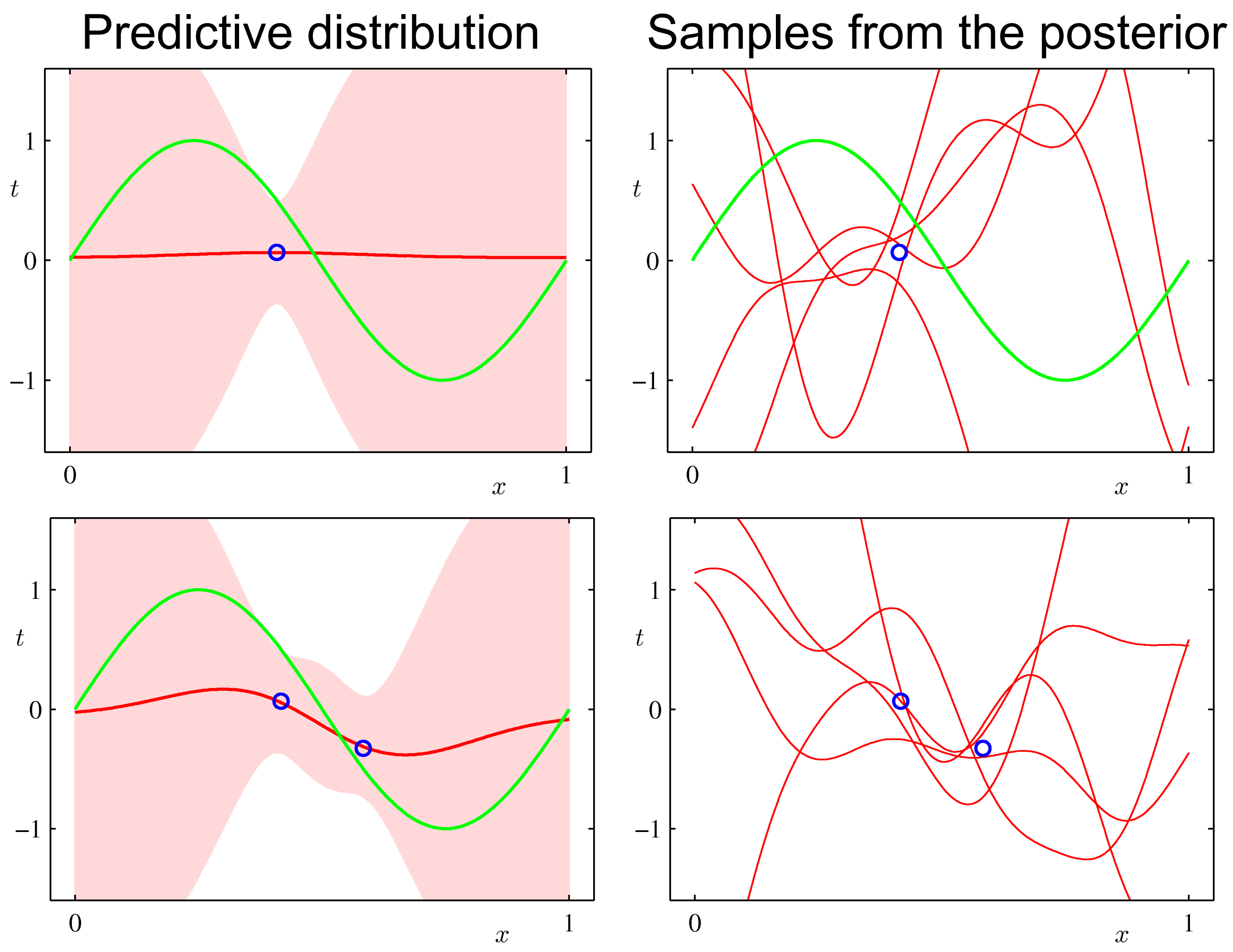
Bayesian predictive distribution



$$p(t|x, \mathbf{t}, \mathbf{X}) = \mathcal{N}\big(t|\mathbf{m}_N^T \boldsymbol{\phi}(x), \sigma_N^2(x)\big)$$

Predictive Distribution

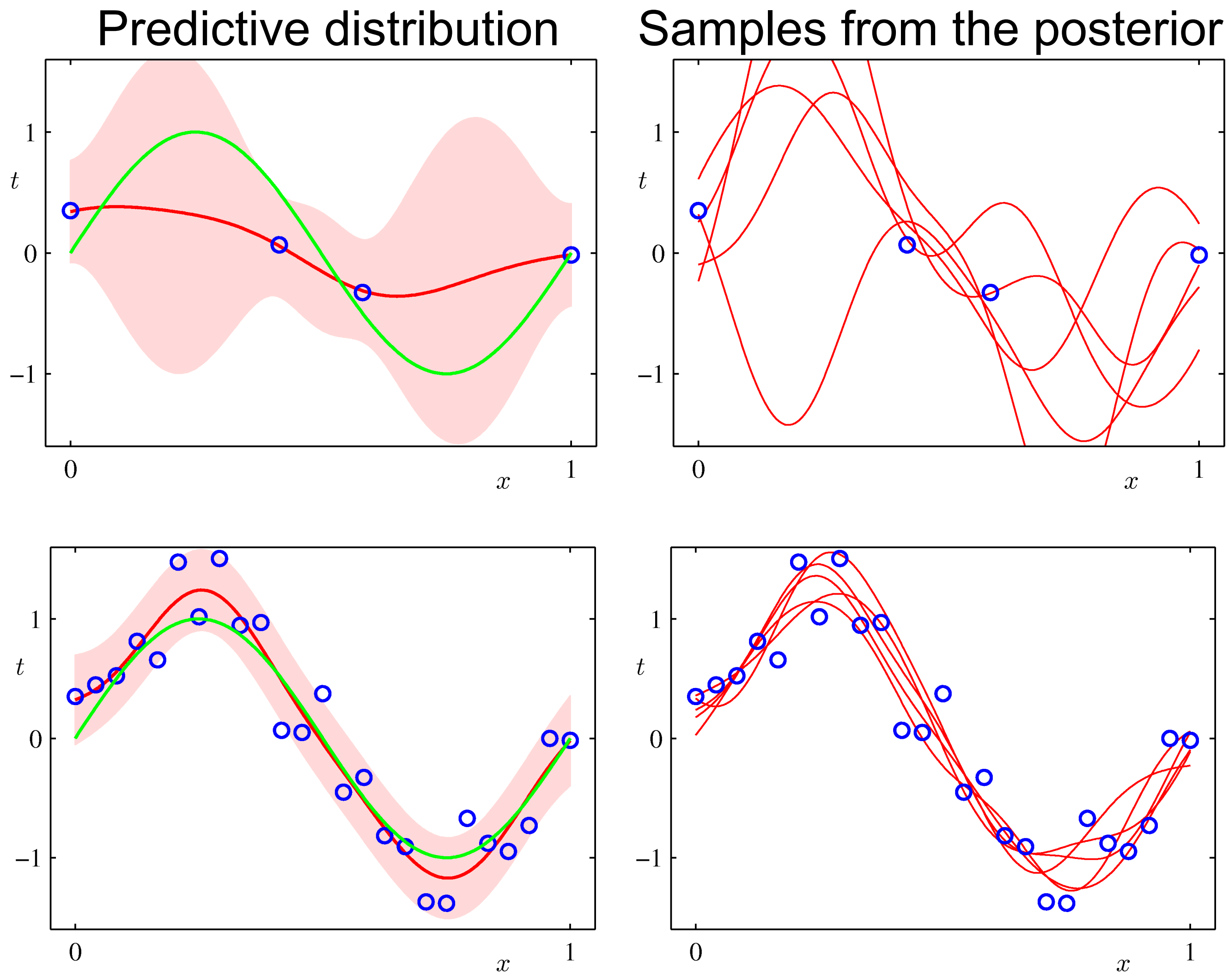
Sinusoidal dataset, 9 Gaussian basis functions.


Predictive distribution

Samples from the posterior

Predictive Distribution

Sinusoidal dataset, 9 Gaussian basis functions.



Predictive distribution

Samples from the posterior

Gamma-Gaussian Conjugate Prior

- So far we have assumed that the noise parameter β is known.
- If both **w** and β are treated as unknown, then we can introduce a conjugate prior distribution that will be given by the Gaussian-Gamma distribution:

$$p(\mathbf{w}, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \beta^{-1}\mathbf{S}_0)\mathrm{Gam}(\beta|a_0, b_0),$$

where the Gamma distribution is given by:

$$\mathrm{Gam}(\beta|a, b) = \frac{1}{\Gamma(a)} b^a \beta^{a-1} \exp(-b\beta), \qquad \Gamma(a) = \int_0^\infty u^{a-1} e^{-u} \mathrm{d}u.$$

- The posterior distribution takes the same functional form as the prior:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \beta^{-1}\mathbf{S}_N)\mathrm{Gam}(\beta|a_N, b_N).$$

Equivalent Kernel

- The predictive mean can be written as:

$$\begin{aligned} y(\mathbf{x}, \mathbf{m}_N) &= \mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \\ &= \sum_{n=1}^{N} \underbrace{\beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}_n)} t_n \\ &= \sum_{n=1}^{N} \overbrace{k(\mathbf{x}, \mathbf{x}_n)} t_n . \end{aligned}$$

$$\boxed{\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} . \end{aligned}}$$

Equivalent kernel or smoother matrix.

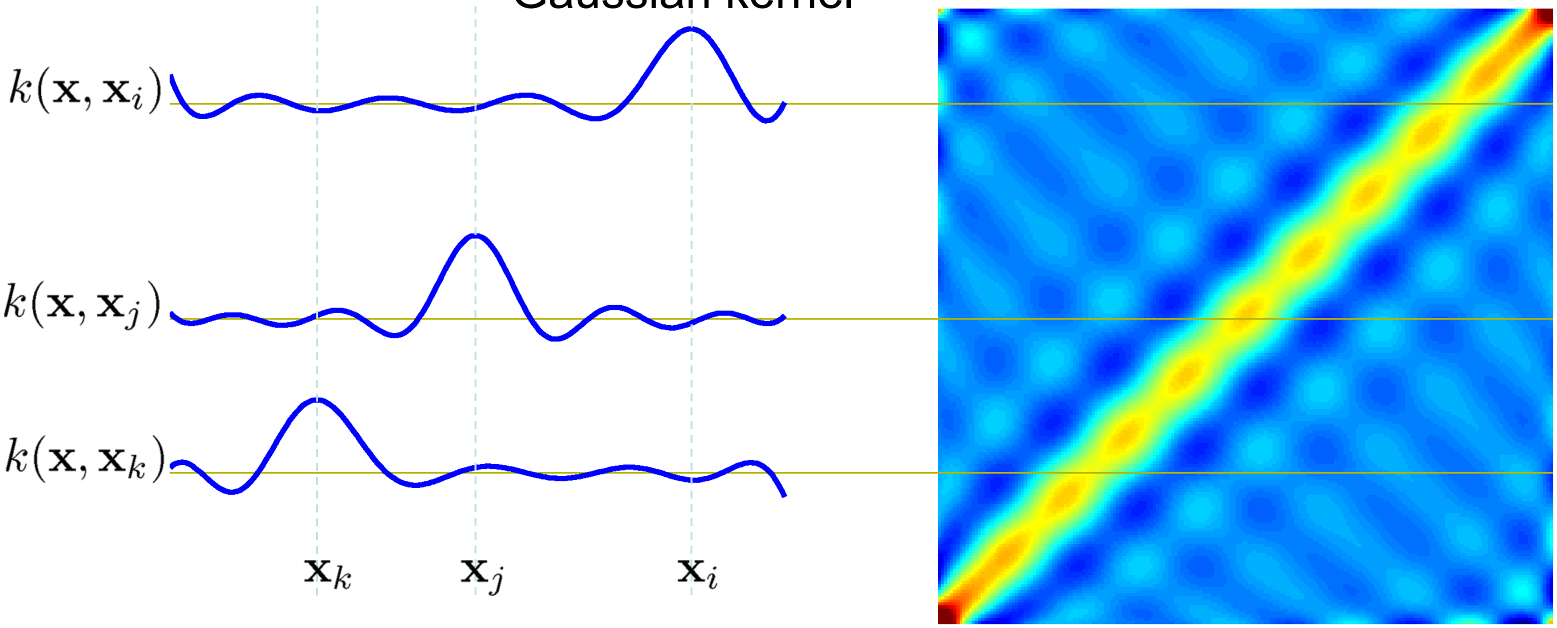
- The mean of the predictive distribution at a time **x** can be written as a linear combination of the training set target values.

- Such regression functions are called linear smoothers.

Equivalent Kernel

- The weight of t_n depends on distance between x and x_n; nearby x_n carry more weight.

Gaussian kernel



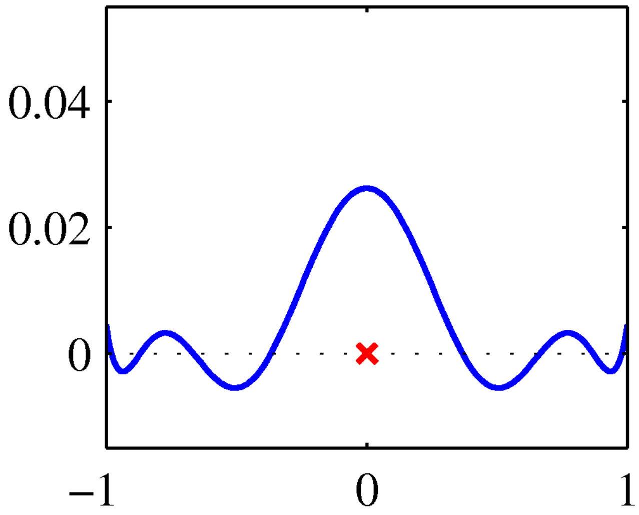
- The kernel as a covariance function:

$$\begin{aligned}
\operatorname{cov}[y(\mathbf{x}), y(\mathbf{x}')] &= \operatorname{cov}[\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{w}, \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}')] \\
&= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}') = \beta^{-1}k(\mathbf{x}, \mathbf{x}').
\end{aligned}$$

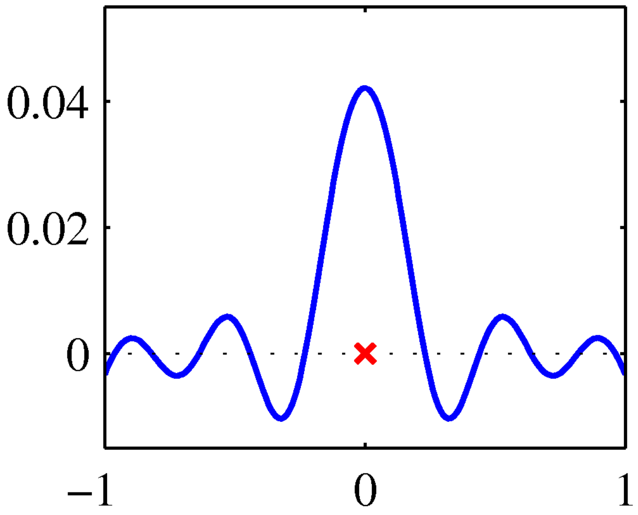
- We can avoid the use of basis functions and define the kernel function directly, leading to *Gaussian Processes.*

Other Kernels

- Examples of kernels k(x,x') for x=0, plotted as a function corresponding to x'.



Polynomial



Sigmoidal

- Note that these are localized functions of x'.

Bayesian Model Comparison

- The Bayesian view of model comparison involves the use of probabilities to represent uncertainty in the choice of the model.
- We would like to compare a set of L models $\{\mathcal{M}_i\},$ where $i = 1, 2, ..., L,$ using a training set D.
- We specify the prior distribution over the different models $p(\mathcal{M}_i).$
- Given a training set D, we evaluate the posterior:

$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i).$$

Posterior Prior *Model evidence* or *marginal likelihood*

- For simplicity, we will assume that all model are a-priori equal.
- The model evidence expresses the preference shown by the data for different models.
- The ratio of two model evidences for two models is known as Bayes factor:

$$\frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}$$

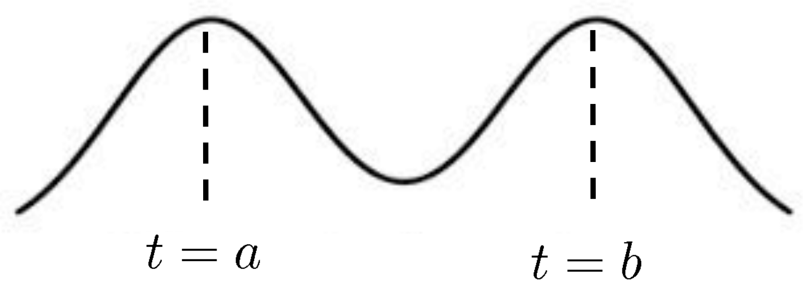
Bayesian Model Comparison

- Once we compute the posterior $p(M_i|\mathcal{D})$, we can compute the predictive (mixture) distribution:

$$p(t|\mathbf{x},\mathcal{D}) = \sum_{i=1}^{L} p(t|\mathbf{x},\mathcal{M}_i,\mathcal{D})p(\mathcal{M}_i|\mathcal{D}).$$

- The overall predictive distribution is obtained by averaging the predictive distributions of individual models, weighted by the posterior probabilities.

- For example, if we have two models, and one predicts a narrow distribution around t=a while the other predicts a narrow distribution around t=b, then the overall predictions will be bimodal:



- A simpler approximation, known as model selection, is to use the model with the highest evidence.

Bayesian Model Comparison

- Remember, the posterior is given by

$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i).$$

For a model governed by a set of parameters **w**, the model evidence can be computed as follows:

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w},\mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)\,\mathrm{d}\mathbf{w}.$$

- Observe that the evidence is the normalizing term that appears in the denominator in Bayes' rule:

$$p(\mathbf{w}|\mathcal{D},\mathcal{M}_i) = \frac{p(\mathcal{D}|\mathbf{w},\mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_i)}$$

- The model evidence is also often called marginal likelihood.

Bayesian Model Comparison

- We next get some insight into the model evidence by making simple approximations.

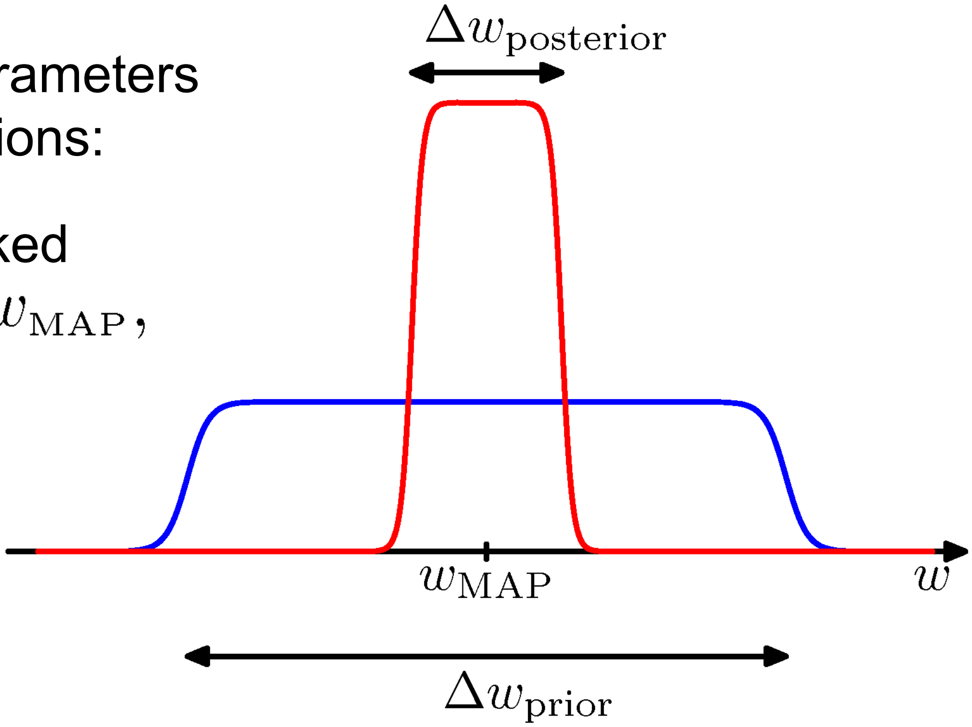
- For a give model with a single parameters parameter, w, consider approximations:

  - Assume that the posterior is picked around the most probable value w_{MAP}, with width $\Delta w_{\mathrm{posterior}}$

  - Assume that the prior is flat with width $\Delta w_{\mathrm{prior}}$

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)\,\mathrm{d}w$$

$$\simeq \quad p(\mathcal{D}|w_{\mathrm{MAP}})\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}$$



Bayesian Model Comparison

- Taking the logarithms, we obtain:

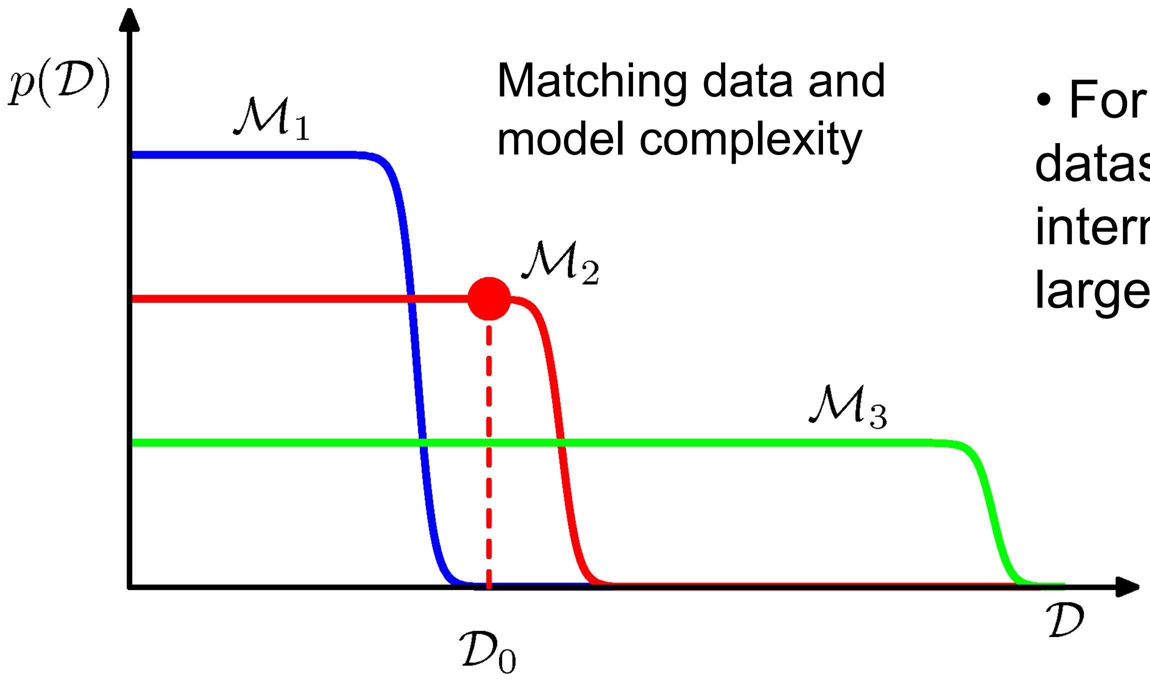
$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|w_{\text{MAP}}) + \ln \underbrace{\left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right)}_{\text{Negative}} .$$

- With M parameters, all assumed to have the same $\Delta w_{\text{posterior}}/\Delta w_{\text{prior}}$ ratio:

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}}) + M \underbrace{\ln \left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right)}_{\text{Negative and linear in M.}} .$$

- As we increase the complexity of the model (increase the number of adaptive parameters M), the first term will increase, whereas the second term will decrease due to the dependence on M.
- The optimal model complexity: trade-off between these two competing terms.

Bayesian Model Comparison



Matching data and model complexity

- For the particular observed dataset $\mathcal{D}_0$, the model $\mathcal{M}_2$ with intermediate complexity has the largest evidence.

- The simple model cannot fit the data well, whereas the more complex model spreads its predictive probability and so assigns relatively small probability to any one of them.

- The marginal likelihood is very sensitive to the prior used!

- Computing the marginal likelihood makes sense only if you are certain about the choice of the prior.

Evidence Approximation

- In the fully Bayesian approach, we would also specify a prior distribution over the hyperparameters $p(\alpha, \beta)$.

- The fully Bayesian predictive distribution is then given by marginalizing over model parameters as well as hyperparameters:

Likelihood

posterior over weights

posterior over hyperparameters

$$p(t^*|\mathbf{x}^*, \mathcal{D}) = \iiint p(t^*|\mathbf{x}^*, \mathbf{w}, \beta) p(\mathbf{w}|\mathcal{D}, \alpha, \beta) p(\alpha, \beta|\mathcal{D}) \mathrm{d}\mathbf{w}\, \mathrm{d}\alpha\, \mathrm{d}\beta.$$

target and input on test case

precision of output noise

precision of the prior

training data: inputs and targets

- However, this integral is intractable (even when everything is Gaussian). Need to approximate.

- Note: the fully Bayesian approach is to integrate over the posterior distribution for $\{\alpha, \beta, \mathbf{w}\}$. This can be done by MCMC, which we will consider later. For now, we will use evidence approximation: much faster.

Evidence Approximation

- The fully Bayesian predictive distribution is given by:

$$p(t^*|\mathbf{x}^*, \mathcal{D}) = \iiint p(t^*|\mathbf{x}^*, \mathbf{w}, \beta) p(\mathbf{w}|\mathcal{D}, \alpha, \beta) p(\alpha, \beta|\mathcal{D}) \mathrm{d}\mathbf{w}\, \mathrm{d}\alpha\, \mathrm{d}\beta.$$

- If we assume that the posterior over hyperparameters α and β is sharply picked, we can approximate:

$$p(t^*|\mathbf{x}^*, \mathcal{D}) \approx p(t^*|\mathbf{x}^* \mathcal{D}, \hat{\alpha}, \hat{\beta}) = \int p(t^*|\mathbf{x}^*, \mathcal{D}, \hat{\beta}) p(\mathbf{w}|\mathcal{D}, \hat{\alpha}, \hat{\beta}) \mathrm{d}\mathbf{w}.$$

where $\left(\widehat{\alpha}, \widehat{\beta}\right)$ is the mode of the posterior $p(\alpha, \beta|\mathcal{D})$.

- So we integrate out parameters but maximize over hyperparameters.

- This is known as empirical Bayes, Type II Maximum Likelihood, Evidence Approximation.

Evidence Approximation

- From Bayes' rule we obtain:

$$p(\alpha, \beta|\mathbf{t}, \mathbf{X}) \propto p(\mathbf{t}|\mathbf{X}, \alpha, \beta)p(\alpha, \beta).$$

- If we assume that the prior over hyperparameters $p(\alpha, \beta)$ is flat, we get:

$$p(\alpha, \beta|\mathbf{t}, \mathbf{X}) \propto p(\mathbf{t}|\mathbf{X}, \alpha, \beta).$$

- The values $\left(\widehat{\alpha}, \widehat{\beta}\right)$ are obtained by maximizing the marginal likelihood $p(\mathbf{t}|\mathbf{X}, \alpha, \beta).$

- This will allow us to determine the values of these hyperparameters from the training data.

- Recall that the ratio α/β is analogous to the regularization parameter.

Evidence Approximation

- The marginal likelihood is obtained by integrating out parameters:

$$p(\mathbf{t}|\mathbf{X}, \alpha, \beta) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w}|\alpha) \mathrm{d}\mathbf{w}.$$

$$\boxed{\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}. \end{aligned}}$$

- We can write the evidence function in the form:

$$p(\mathbf{t}|\mathbf{X}, \alpha, \beta) = \left(\frac{\beta}{2\pi}\right)^{N/2} \left(\frac{\alpha}{2\pi}\right)^{M/2} \int \exp\big(-E(\mathbf{w})\big) \mathrm{d}\mathbf{w},$$

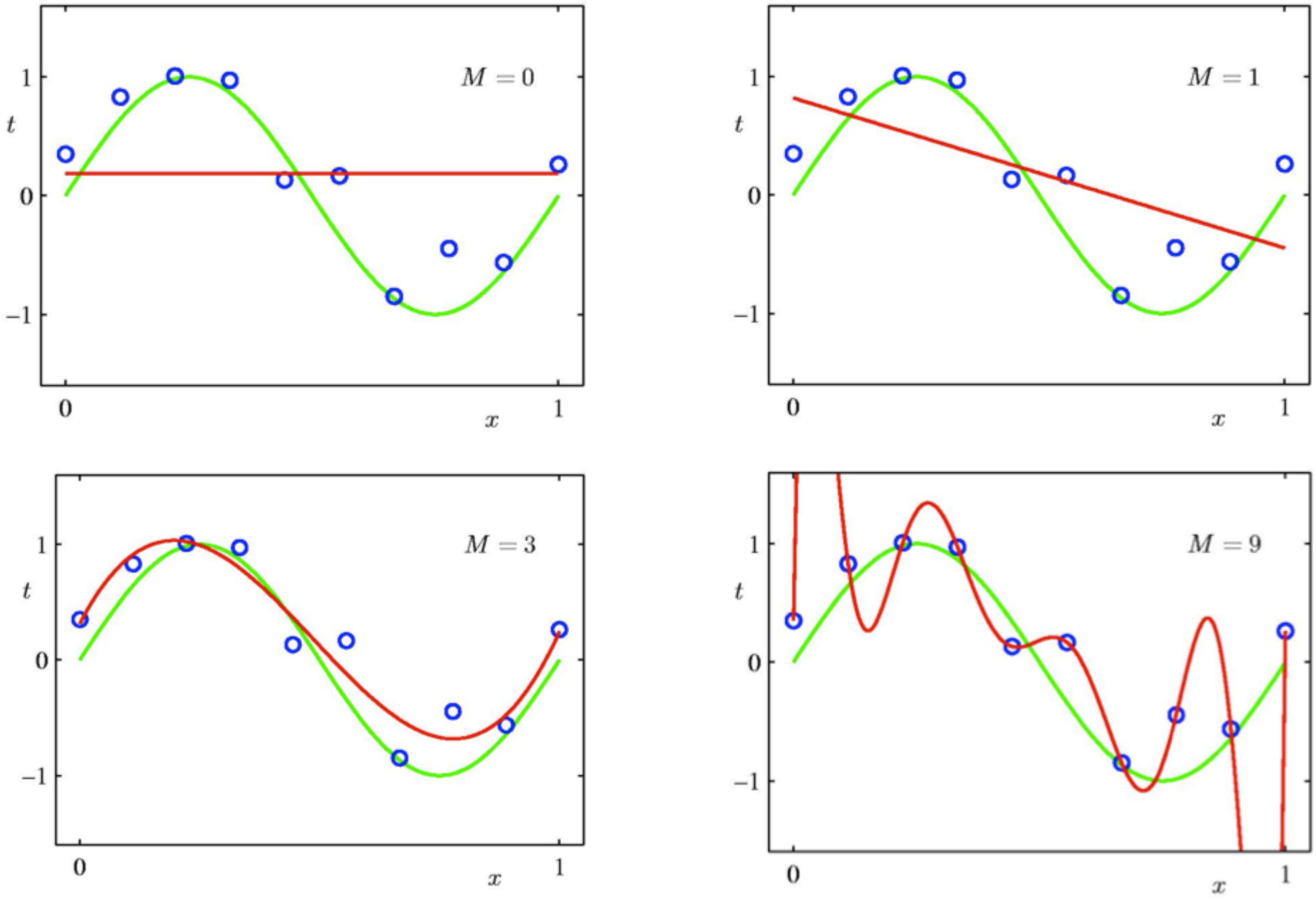
where

$$E(\mathbf{w}) = \beta E_{\mathcal{D}}(\mathbf{w}) + \alpha E_W(\mathbf{w}) = \frac{\beta}{2}||\mathbf{t} - \boldsymbol{\Phi}\mathbf{w}||^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}.$$

- Using standard results for the Gaussian distribution, we obtain:

$$\ln p(\mathbf{t}|\alpha, \beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) + \frac{1}{2}\ln|\mathbf{S}_N| - \frac{N}{2}\ln(2\pi).$$

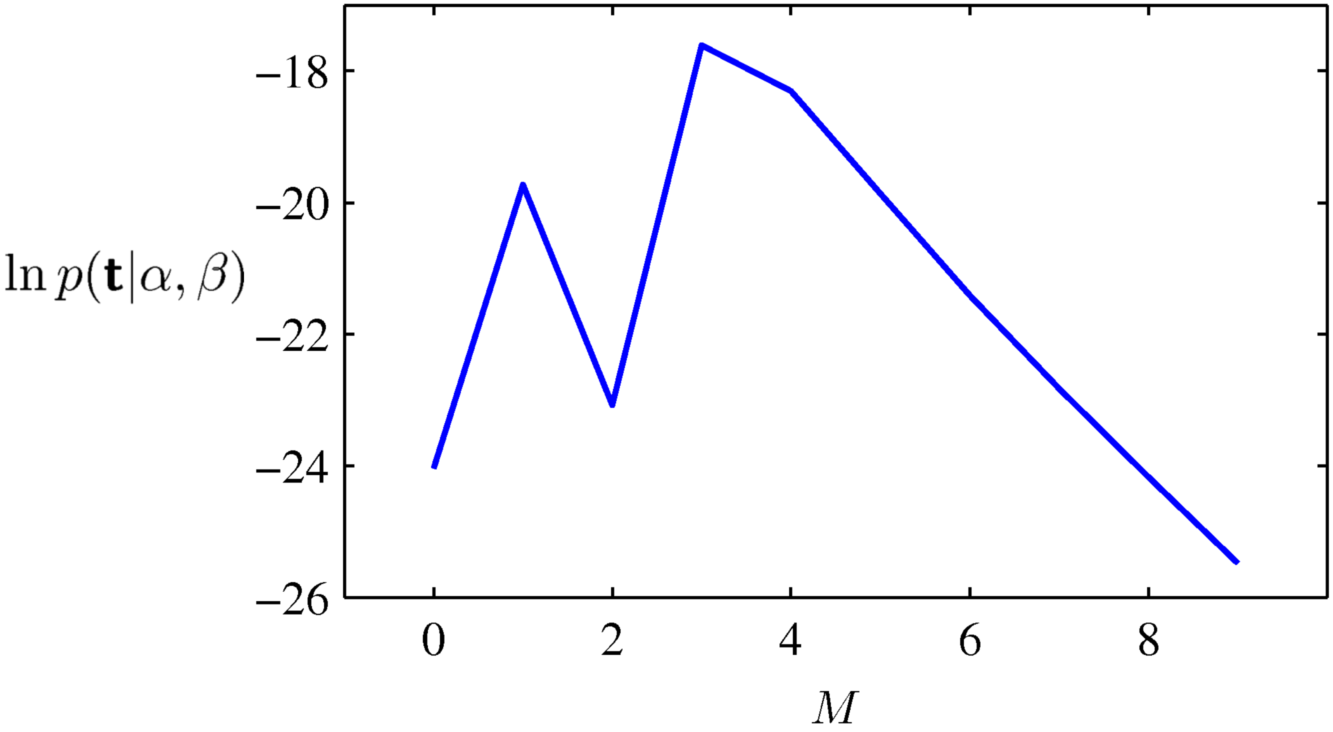
Some Fits to the Data



For M=9, we have fitted the training data perfectly.

Evidence Approximation

Using sinusoidal data, Mth degree polynomial.



The evidence favours the model with M=3.

Maximizing the Evidence

- Remember:

$$\ln p(\mathbf{t}|\alpha,\beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) + \frac{1}{2}\ln|\mathbf{S}_N| - \frac{N}{2}\ln(2\pi).$$

- To maximize the evidence $p(\mathbf{t}|\mathbf{X},\alpha,\beta)$ with respect to α and β, define the following eigenvector equation:

$$\left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\mathbf{u}_i = \lambda_i\mathbf{u}_i.$$

Precision matrix of the Gaussian posterior distribution

- Therefore the matrix:

$$\mathbf{A} = \mathbf{S}_N^{-1} = \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}$$

has eigenvalues α + λ_i.

- The derivative:

$$\frac{d}{d\alpha}\ln|\mathbf{A}| = \frac{d}{d\alpha}\ln\prod_i(\alpha+\lambda_i) = \frac{d}{d\alpha}\sum_i\ln(\alpha+\lambda_i) = \sum_i\frac{1}{\alpha+\lambda_i}.$$

Maximizing the Evidence

- Remember:

$$\ln p(\mathbf{t}|\alpha,\beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) + \frac{1}{2}\ln|\mathbf{S}_N| - \frac{N}{2}\ln(2\pi).$$

where

$$E(\mathbf{m}_N) = \frac{\beta}{2}||\mathbf{t} - \mathbf{\Phi}\mathbf{m}_N||^2 + \frac{\alpha}{2}\mathbf{m}_N^T\mathbf{m}_N.$$

- Differentiating $\ln p(\mathbf{t}|\alpha,\beta)$, the stationary points with respect to α satisfy:

$$\frac{M}{2\alpha} - \frac{1}{2}\mathbf{m}_N^T\mathbf{m}_N - \frac{1}{2}\sum_i \frac{1}{\alpha+\lambda_i} = 0.$$

$$\alpha\mathbf{m}_N^T\mathbf{m}_N = M - \alpha\sum_i \frac{1}{\alpha+\lambda_i} = \gamma,$$

where the quantity γ, effective number of parameters, can be defined as:

$$\gamma = \sum_i \frac{\lambda_i}{\lambda_i+\alpha}.$$

Maximizing the Evidence

- The stationary points with respect to α satisfy:

$$\alpha \mathbf{m}_N^T \mathbf{m}_N = M - \alpha \sum_i \frac{1}{\alpha + \lambda_i} = \gamma,$$

where the quantity γ, effective number of parameters, is defined as:

$$\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \alpha}.$$

Note that the eigenvalues need to be computed only once.

- Iterate until convergence:

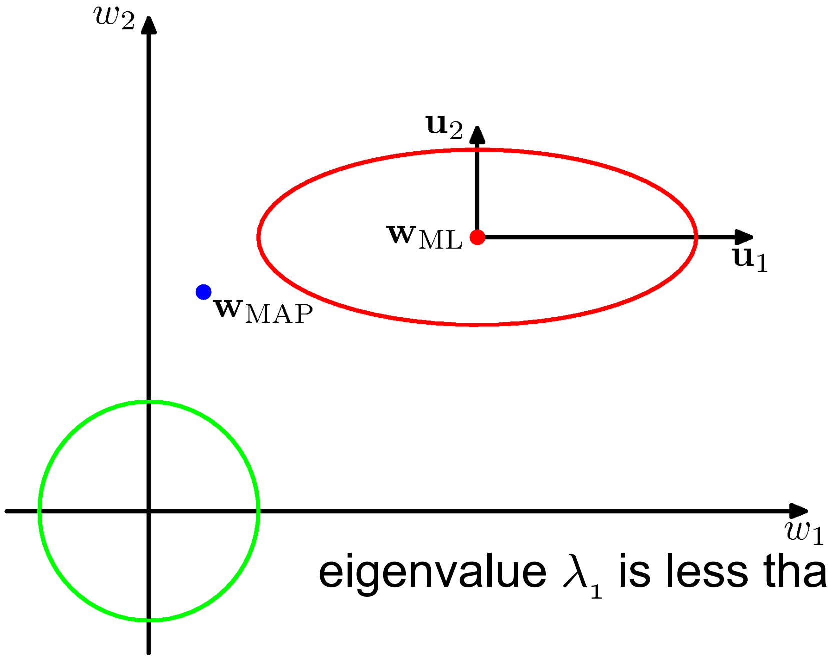
$$\alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}}; \qquad \gamma = \sum_i \frac{\lambda_i}{\lambda_i + \alpha}; \qquad \begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}. \end{aligned}$$

- Similarly:

$$\frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^{N} \left\{ t_n - \mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2$$

Effective Number of Parameters

- Consider the contours of the likelihood function and the prior.



eigenvalue λ_1 is less than λ_2.

- The eigenvalue λ_i measures the curvature of the log-likelihood function.
- The quantity γ will lie $0 \leq \gamma \leq$ M.
- For $\lambda_i \gg \alpha$, the corresponding parameter **w**$_{\mathbf{i}}$ will be close to its maximum likelihood. The ratio:

$$\frac{\lambda_i}{\lambda_i + \alpha}$$ will be close to one.

- Such parameters are called well determined, as their values are highly constrained by the data.
- For $\lambda_i \ll \alpha$, the corresponding parameters will be close to zero (pulled by the prior), as will the ratio $\lambda_i/(\lambda_i + \alpha)$.
- We see that γ measures the effective total number of well determined parameters.

Quick Approximation

- In the limit $N \gg M$, γ = M, and we consider to use the easy to compute approximations:

$$\begin{aligned}\alpha &= \frac{M}{\mathbf{m}_N^{\mathrm{T}}\mathbf{m}_N} \\ \frac{1}{\beta} &= \frac{1}{N}\sum_{n=1}^{N}\left\{t_n - \mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2 .\end{aligned}$$

Limitations

- M basis function along each dimension of a D-dimensional input space requires M^D basis functions: the curse of dimensionality.

- Fortunately, we can get away with fewer basis functions, by choosing these using the training data (e.g. adaptive basis functions), which we will see later.

- Second, the data vectors typically lie close to a nonlinear low-dimensional manifold, whose intrinsic dimensionality is smaller than that of the input space.

Linear Models for Classification

- So far, we have looked at the linear models for regression that have particularly simple analytical and computational properties.

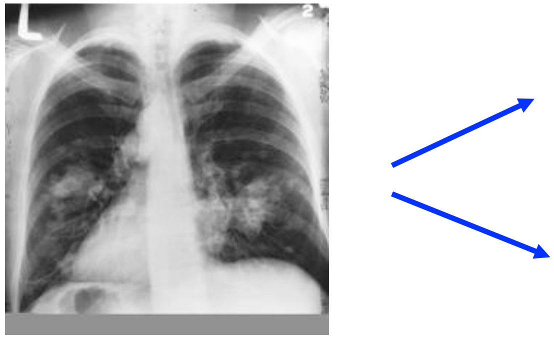
- We will now look at analogous class of models for solving classification problems.

- We will also look at the Bayesian treatment of linear models for classification.

Classification

- The goal of classification is to assign an input **x** into one of K discrete classes C_k, where k=1,..,K.
- Typically, each input is assigned only to one class.

- Example: The input vector **x** is the set of pixel intensities, and the output variable t will represent the presence of cancer, class C_1, or absence of cancer, class C_2.



C_1: Cancer present

C_2: Cancer absent

x -- set of pixel intensities

Linear Classification

- The goal of classification is to assign an input **x** into one of K discrete classes C_k, where k=1,..,K.

- The input space is divided into decision regions whose boundaries are called decision boundaries or decision surfaces.

- We will consider linear models for classification. Remember, in the simplest linear regression case, the model is linear in parameters:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T \mathbf{w} + w_0.$$

adaptive parameters

$$y(\mathbf{x}, \mathbf{w}) = f(\mathbf{x}^T \mathbf{w} + w_0).$$

fixed nonlinear function:
activation function

- For classification, we need to predict discrete class labels, or posterior probabilities that lie in the range of (0,1), so we use a nonlinear function.

Linear Classification

$$y(\mathbf{x}, \mathbf{w}) = f(\mathbf{x}^T\mathbf{w} + w_0).$$

- The decision surfaces correspond to $y(\mathbf{x}, \mathbf{w}) = \mathrm{const},$ so that $\mathbf{x}^T\mathbf{w} + w_0 = \mathrm{const},$ and hence the decision surfaces are linear functions of **x**, even if the activation function is nonlinear.

- These class of models are called generalized linear models.

- Note that these models are no longer linear in parameters, due to the presence of nonlinear activation function.

- This leads to more complex analytical and computational properties, compared to linear regression.

- Note that we can make a fixed nonlinear transformation of the input variables using a vector of basis functions $\phi(\mathbf{x}),$ as we did for regression models.

Notation

- In the case of two-class problems, we can use the binary representation for the target value $t \in \{0, 1\}$, such that t=1 represents the positive class and t=0 represents the negative class.
  - We can interpret the value of t as the probability of the positive class, and the output of the model can be represented as the probability that the model assigns to the positive class.

- If there are K classes, we use a 1-of-K encoding scheme, in which **t** is a vector of length K containing a single 1 for the correct class and 0 elsewhere.

- For example, if we have K=5 classes, then an input that belongs to class 2 would be given a target vector:

$$t = (0, 1, 0, 0, 0)^T.$$

  - We can interpret a vector **t** as a vector of class probabilities.

Three Approaches to Classification

- First approach: Construct a discriminant function that directly maps each input vector to a specific class.

- Model the conditional probability distribution $p(\mathcal{C}_k|\mathbf{x})$, and then use this distribution to make optimal decisions.

- There are two alternative approaches:
  - Discriminative Approach: Model $p(\mathcal{C}_k|\mathbf{x})$, directly, for example by representing them as parametric models, and optimize for parameters using the training set (e.g. logistic regression).
  - Generative Approach: Model class conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ together with the prior probabilities $p(\mathcal{C}_k)$ for the classes. Infer posterior probability using Bayes' rule:

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

- For example, we could fit multivariate Gaussians to the input vectors of each class. Given a test vector, we see under which Gaussian the test vector is most probable.

Discriminant Functions

- Consider: $y(\mathbf{x}) = \mathbf{x}^T \mathbf{w} + w_0.$

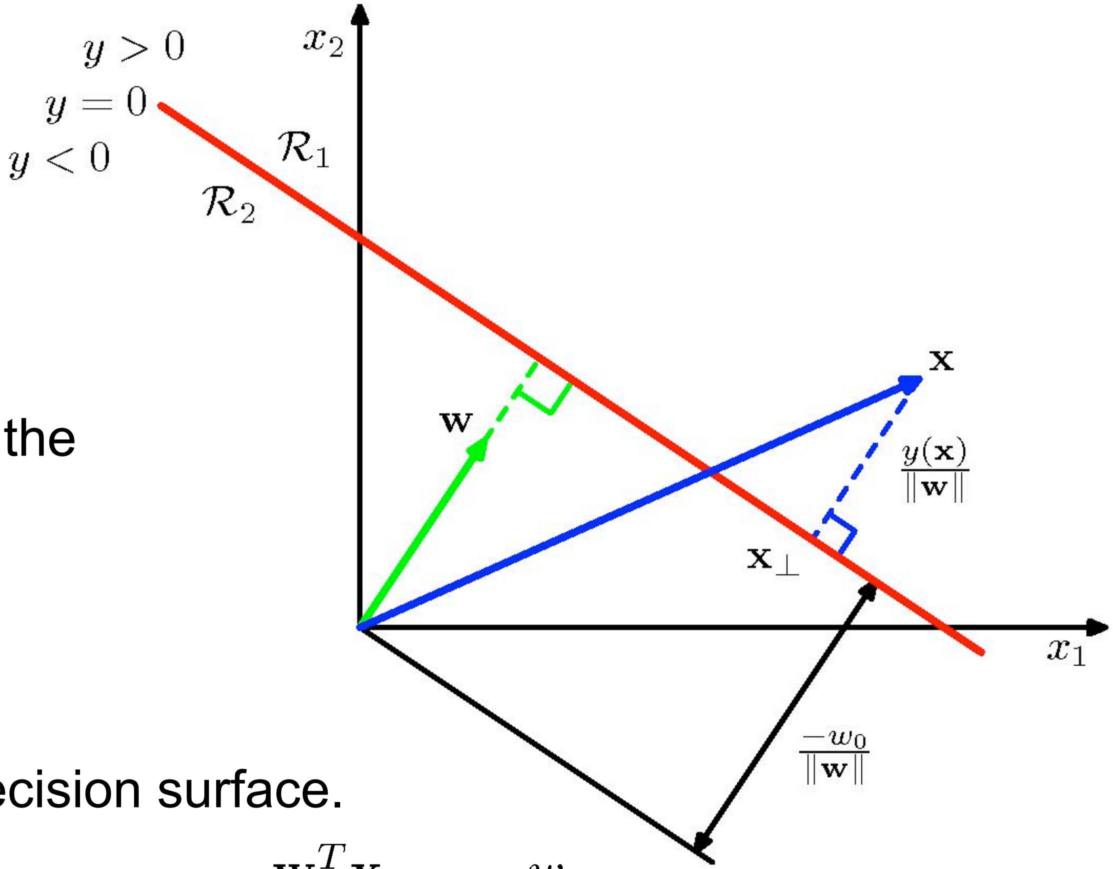
- Assign **x** to C_1 if $y(\mathbf{x}) \geq 0,$ and class C_2 otherwise.

- Decision boundary:

$$y(\mathbf{x}) = 0.$$

- If two points $\mathbf{x_A}$ and $\mathbf{x_B}$ lie on the decision surface, then:

$$y(\mathbf{x}_A) = y(\mathbf{x}_B) = 0,$$
$$\mathbf{w}^T(\mathbf{x}_A - \mathbf{x}_B) = 0.$$



- The **w** is orthogonal to the decision surface.

- If **x** is a point on decision surface, then: $\frac{\mathbf{w}^T\mathbf{x}}{||\mathbf{w}||} = -\frac{w_0}{||\mathbf{w}||}.$

- Hence w_0 determines the location of the decision surface.

Multiple Classes

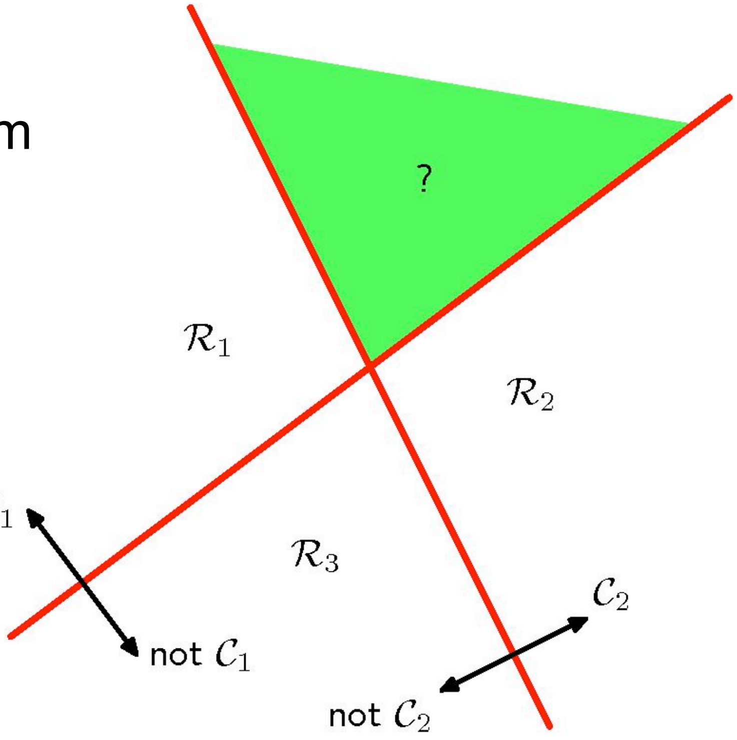
- Consider the extension of linear discriminants to K>2 classes.

- One option is to use K-1 classifiers, each of which solves a two class problem:

    - Separate points in class C_k from points not in that class.

- There are regions in input space that are ambiguously classified.

One-versus-the-rest



Multiple Classes

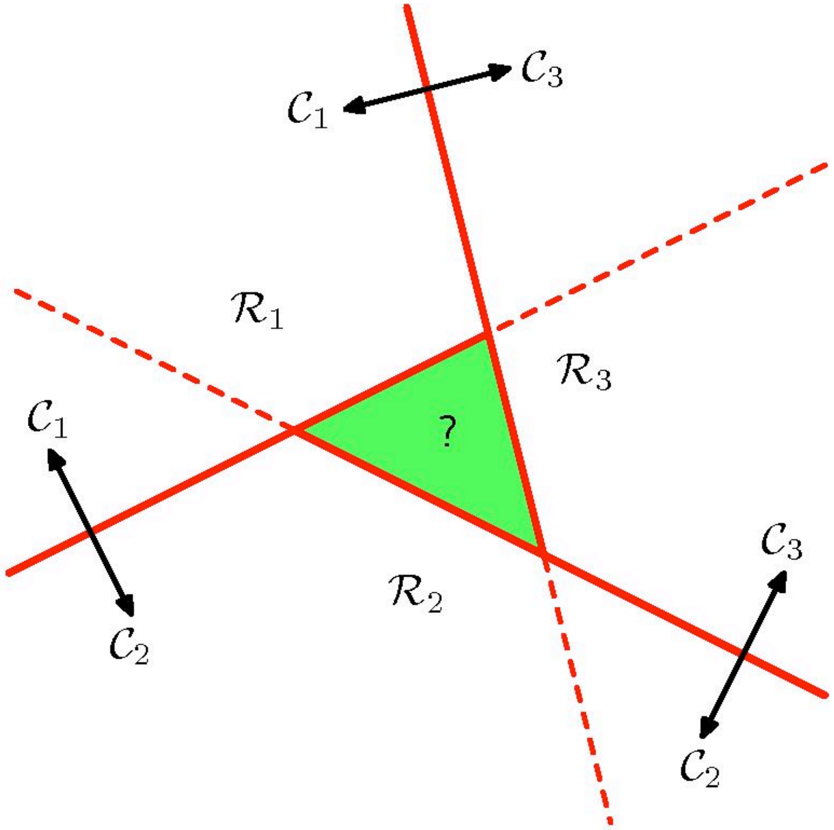
- Consider the extension of linear discriminants to K>2 classes.

- An alternative is to use K(K-1)/2 binary discriminant functions.

  - Each function discriminates between two particular classes.

- Similar problem of ambiguous regions.

One-versus-one



Simple Solution

- Use K linear discriminant functions of the form:

$$y_k(\mathbf{x}) = \mathbf{x}^T\mathbf{w}_k + w_{k0}, \text{ where } k = 1, ..., K.$$

- Assign **x** to class C$_k$, if $y_k(\mathbf{x}) > y_j(\mathbf{x}) \ \forall j \neq k$ (pick the max).

- This is guaranteed to give decision boundaries that are singly connected and convex.

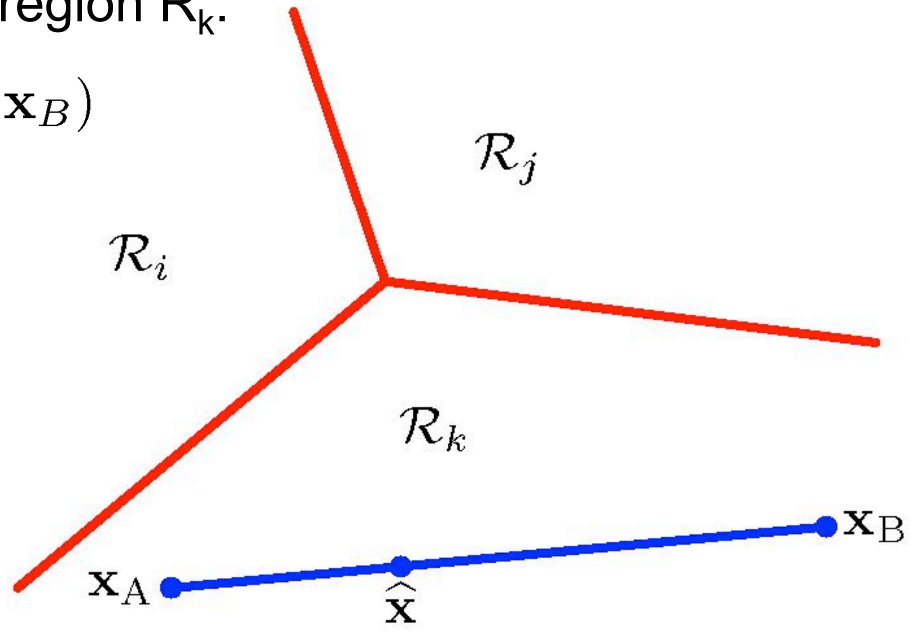
- For any two points that lie inside the region R$_k$:

$$y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A) \text{ and } y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$$

implies that for positive α

$$y_k(\alpha\mathbf{x}_A + (1-\alpha)\mathbf{x}_B) > y_j(\alpha\mathbf{x}_A + (1-\alpha)\mathbf{x}_B)$$

due to linearity of the discriminant functions.



Least Squares for Classification

- Consider a general classification problem with K classes using 1-of-K encoding scheme for the target vector **t**.

- Remember: Least Squares approximates the conditional expectation $\mathbb{E}[\mathbf{t}|\mathbf{x}]$.

- Each class is described by its own linear model:

$$y_k(\mathbf{x}) = \mathbf{x}^T\mathbf{w}_k + w_{k0}, \text{ where } k = 1, ..., K.$$

- Using vector notation, we can write:

$$\mathbf{y}(\mathbf{x}) = \tilde{\mathbf{W}}^T\tilde{\mathbf{x}}$$

(D+1) $\times$ K matrix whose kth column comprises of D+1 dimensional vector:

$$\tilde{\mathbf{w}}_k = (w_{k0}, \mathbf{w}_k^T)^T.$$

corresponding augmented input vector:

$$\tilde{\mathbf{x}} = (1, \mathbf{x}^T)^T.$$

Least Squares for Classification

- Consider observing a dataset {$\mathbf{x}_n$,t_n}, where n=1,…,N.
- We have already seen how to do least squares. Using some matrix algebra, we obtain the optimal weights:

$$\tilde{\mathbf{W}} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{T}$$

Optimal weights

N × (D+1) input matrix whose nth row is $\tilde{\mathbf{x}}_n^T$.

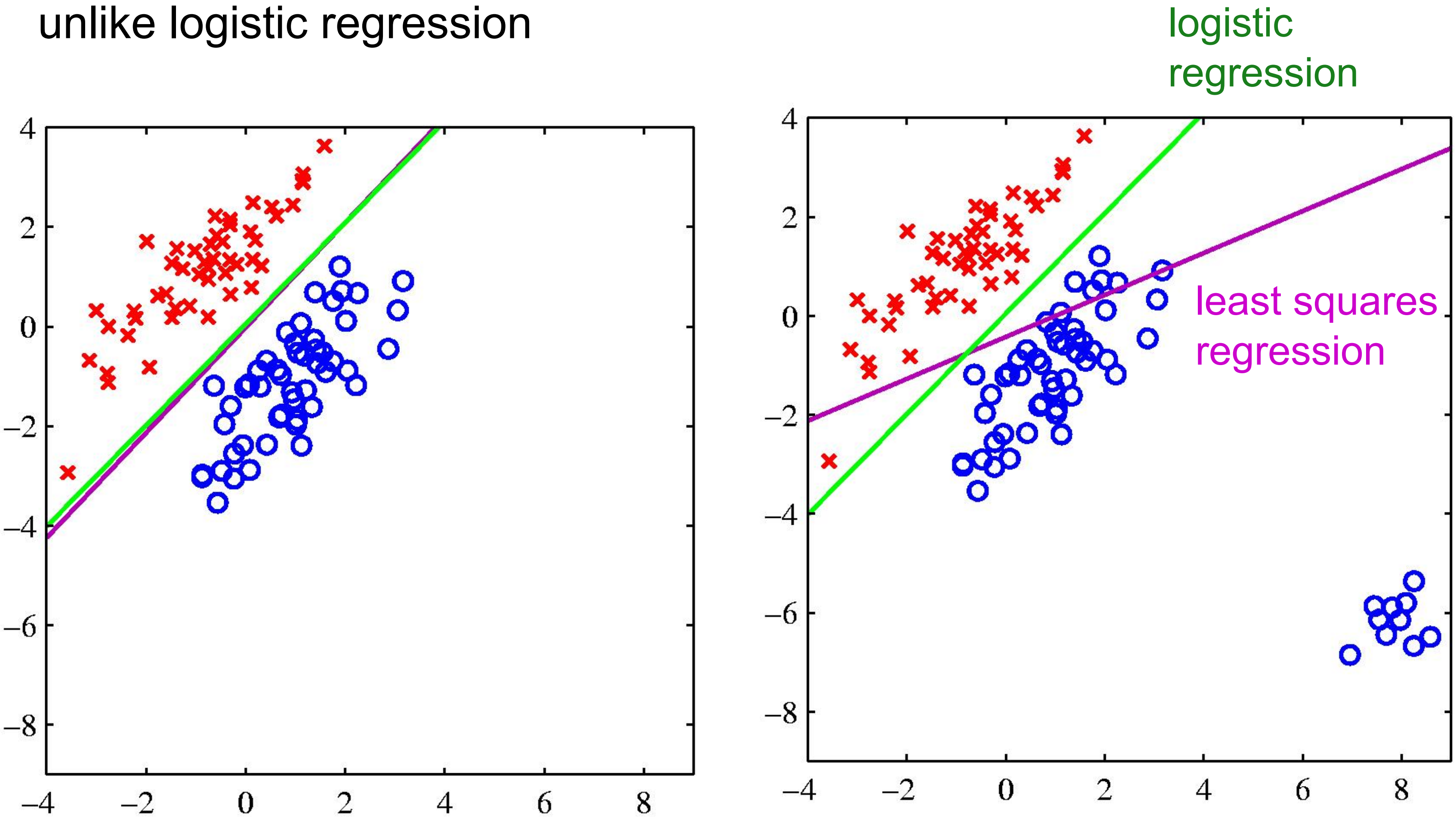
N × K target matrix whose nth row is $\mathbf{t}_n^T$.

- A new input x is assigned to a class for which $y_k = \tilde{\mathbf{x}}^T \tilde{\mathbf{w}}_k$ is largest.

- There are however several problems when using least squares for classification.

Problems using Least Squares

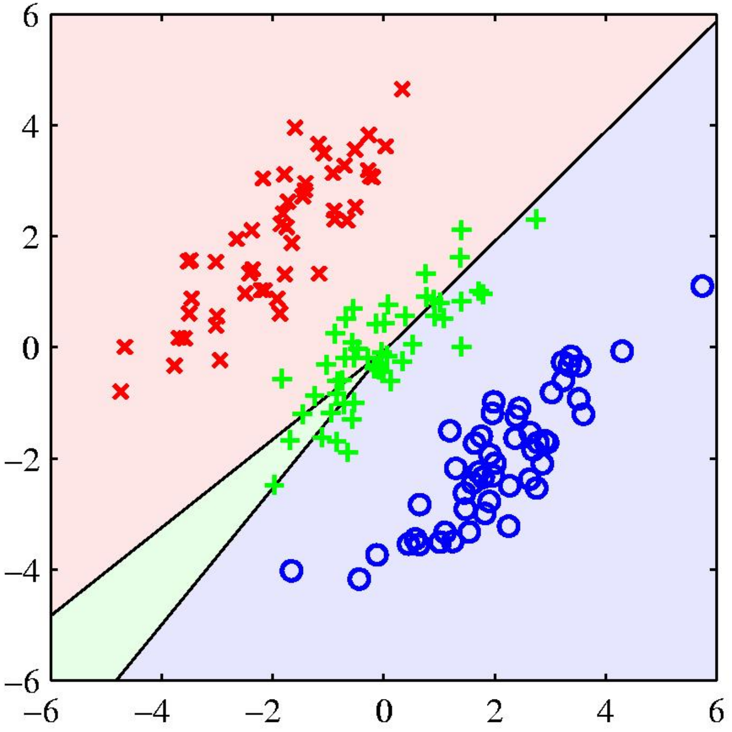
Least squares is highly sensitive to outliers, unlike logistic regression



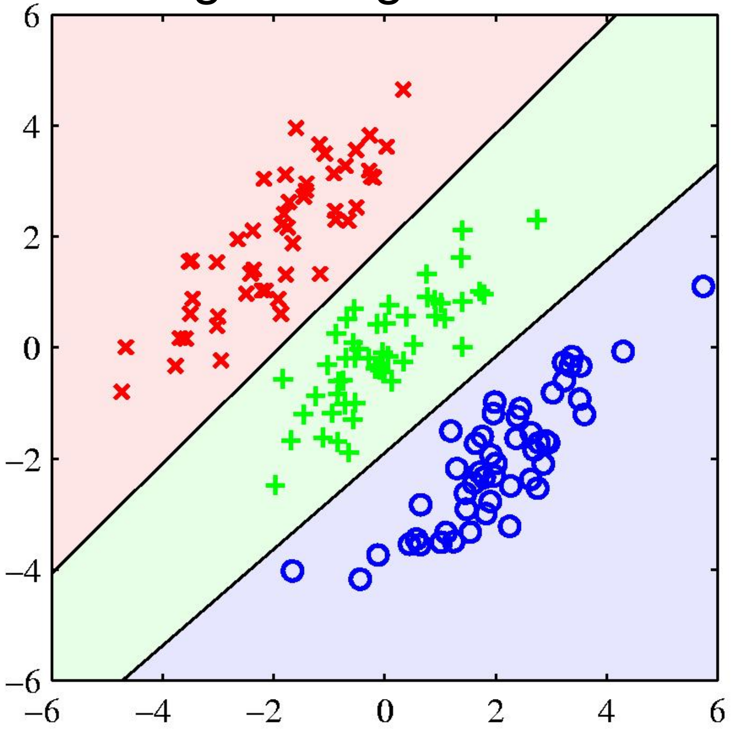
Problems using Least Squares

Example of synthetic dataset containing 3 classes, where lines denote decision boundaries.

Least squares discriminant



Logistic regression



Many green points are misclassified.

Fisher's Linear Discriminant

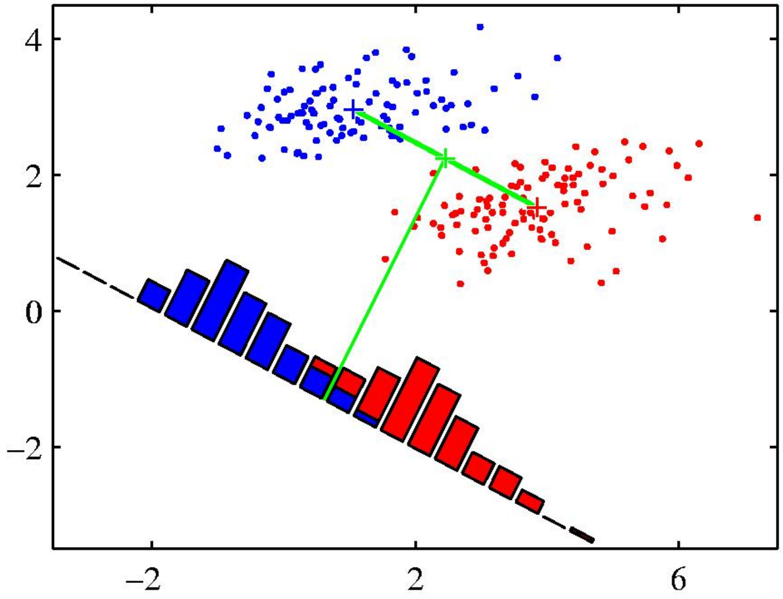
- Dimensionality reduction: Suppose we take a D-dim input vector and project it down to one dimension using:

$$y = \mathbf{w}^T \mathbf{x}.$$

- Idea: Find the projection that maximizes the class separation.
- The simplest measure of separation is the separation of the projected class means. So we project onto the line joining the two means.

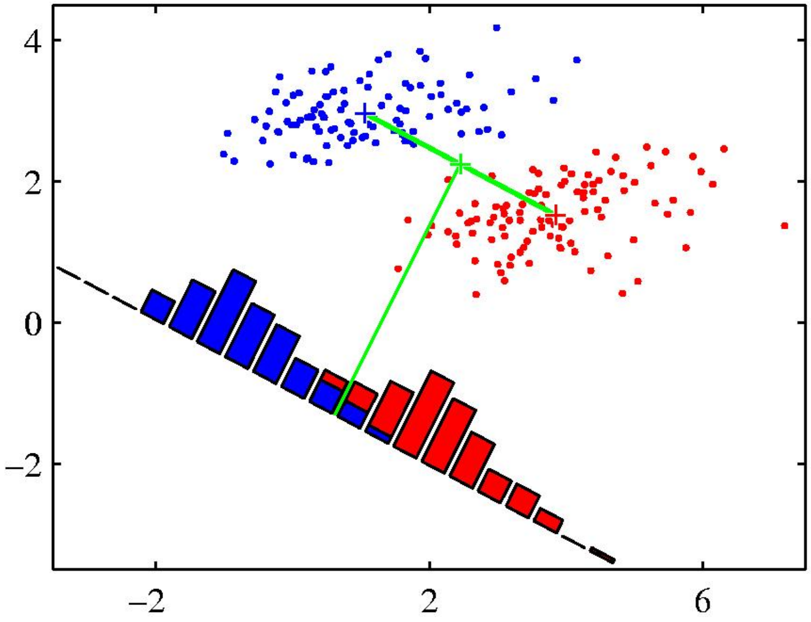
- The problem arises from strongly non-diagonal covariance of the class distributions.

- Fisher's idea: Maximize a function that
  - gives the largest separation between the projected class means,
  - but also gives a small variance within each class, minimizing class overlap.

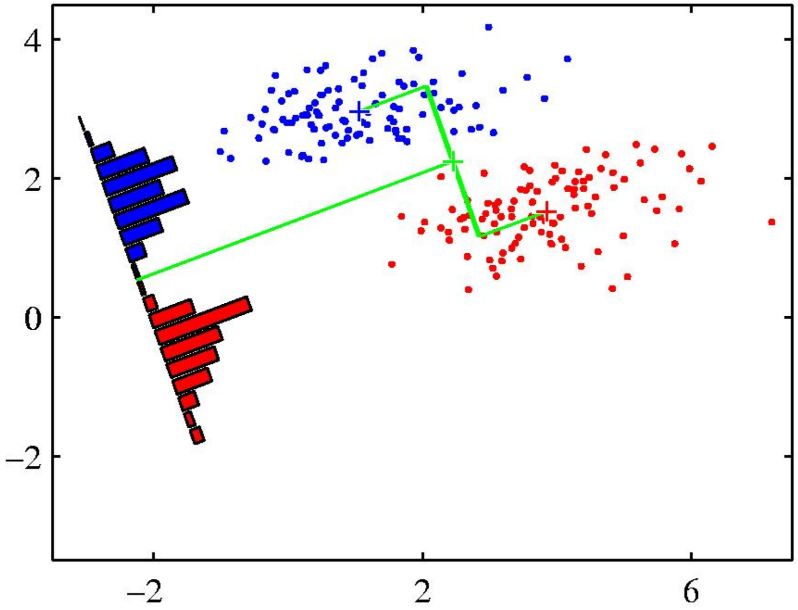


When projected onto the line joining the class means, the classes are not well separated.

Pictorial Illustration



When projected onto the line joining the class means, the classes are not well separated.



Corresponding projection based on the Fisher's linear discriminant.

Fisher's Linear Discriminant

- Let the mean of two classes be given by:

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n,$$

- Projecting onto the vector separating the two classes is reasonable:

$$\mathbf{w} \propto \mathbf{m}_1 - \mathbf{m}_2.$$

- But we also want to minimize the within-class variance:

$$s_1^2 = \sum_{n \in \mathcal{C}_1} (y_n - m_1)^2, \quad s_2^2 = \sum_{n \in \mathcal{C}_2} (y_n - m_2)^2,$$

where $m_k = \mathbf{w}^T \mathbf{m}_k.$

$y_n = \mathbf{w}^T \mathbf{x}_n.$

- We can define the total within-class variance be $s_1^2 + s_2^2$.

- Fisher's criterion: maximize ratio of the between-class variance to within-class variance:

between

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}.$$

within

Fisher's Linear Discriminant

- We can make dependence on **w** explicit:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}},$$

where the between-class and within-class covariance matrices are given by:

$$S_b = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T,$$

$$S_w = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T.$$

- Intuition: differentiating with respect to **w**:

$$(\mathbf{w}^T S_b \mathbf{w}) S_w \mathbf{w} = (\mathbf{w}^T S_w \mathbf{w}) S_b \mathbf{w}.$$

scalar factors

is always in the direction of $(\mathbf{m}_2 - \mathbf{m}_1)$.

- Multiplying by S_w^{-1}, the optimal solution is:

$$\mathbf{w} \propto S_w^{-1}(\mathbf{m}_2 - \mathbf{m}_1).$$

Fisher’s Linear Discriminant

- Notice that the objective $J(\mathbf{w})$ is invariant with respect to rescaling of the vector $\mathbf{w} \to \alpha\mathbf{w}$.
- Maximizing

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$$

is equivalent to the following constraint optimization problem, known as the generalized eigenvalue problem:

$$\min_{\mathbf{w}} -\mathbf{w}^T S_b \mathbf{w}, \quad \text{subject to } \mathbf{w}^T S_w \mathbf{w} = 1.$$

- Forming the Lagrangian:

$$L = -\mathbf{w}^T S_b \mathbf{w} + \lambda(\mathbf{w}^T S_w \mathbf{w} - 1).$$

- The following equation needs to hold at the solution:

$$2S_b\mathbf{w} = 2\lambda S_w \mathbf{w}.$$

- The solution is given by the eigenvector of $S_w^{-1}S_b$ that correspond to the largest eigenvalue.

Three Approaches to Classification

- Construct a discriminant function that directly maps each input vector to a specific class.
- Model the conditional probability distribution $p(\mathcal{C}_k|\mathbf{x})$, and then use this distribution to make optimal decisions.
- There are two alternative approaches:
  - Discriminative Approach: Model $p(\mathcal{C}_k|\mathbf{x})$, directly, for example by representing them as parametric models, and optimize for parameters using the training set (e.g. logistic regression).
  - Generative Approach: Model class conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ together with the prior probabilities $p(\mathcal{C}_k)$ for the classes. Infer posterior probability using Bayes' rule:

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

We will consider next.