

STA 4273H: Statistical Machine Learning

Russ Salakhutdinov

Department of Statistics

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 uncertainty** 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

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

Posterior probability
of weight vector $\mathbf{w}$
given training data $\mathcal{D}$

Marginal likelihood
(normalizing constant):

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

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 $\mathbf{x}^*$, given the training dataset by **integrating over the posterior distribution**:

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

which is sometimes called **predictive distribution**.

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

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

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

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

where $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \dots, \phi_{M-1}(\mathbf{x}))^T$ 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 $p(\theta|\mathbf{x})$ are in the same family as the prior probability distribution $p(\theta)$, 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:

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

- 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

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

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

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

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

Bayesian Linear Regression

- Consider a zero mean isotropic Gaussian prior, which is governed 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:

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

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

- If we consider an infinitely broad prior, $\alpha \rightarrow 0$, the mean $\mathbf{m}_N$ of the posterior distribution reduces to **maximum likelihood value** $\mathbf{w}_{ML}$.
- The log of the posterior distribution is given by the sum of the log-likelihood and the log of the prior:

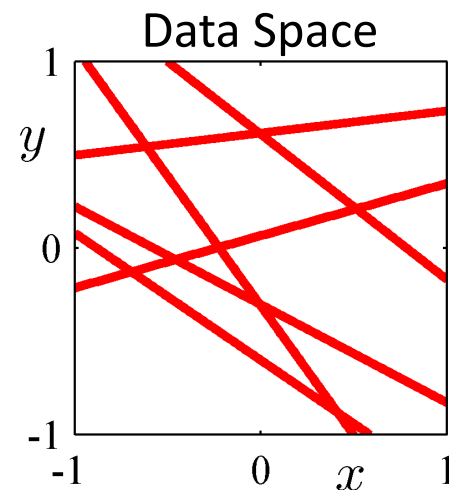
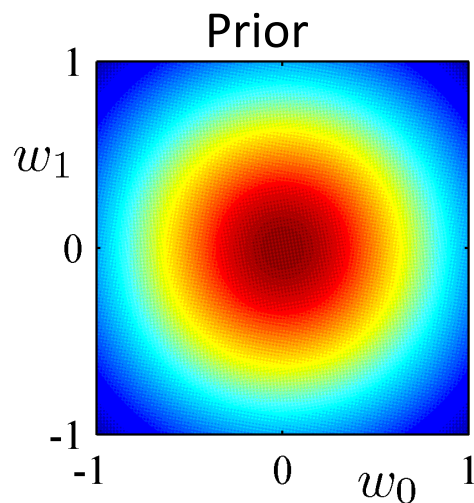
$$\ln p(\mathbf{w} | \mathcal{D}) = -\frac{\beta}{2} \sum_{n=1}^N (t_n - \mathbf{w}^T \phi(\mathbf{x}_n))^2 - \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{const.}$$

- Maximizing this posterior with respect to $\mathbf{w}$ is equivalent to minimizing the sum-of-squares error function with a quadratic regulation term $\lambda = \alpha / \beta$.

Bayesian Linear Regression

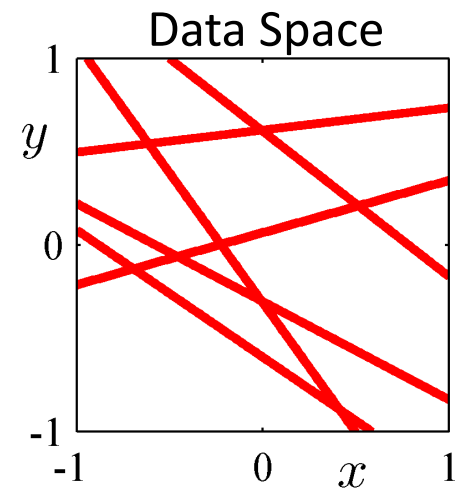
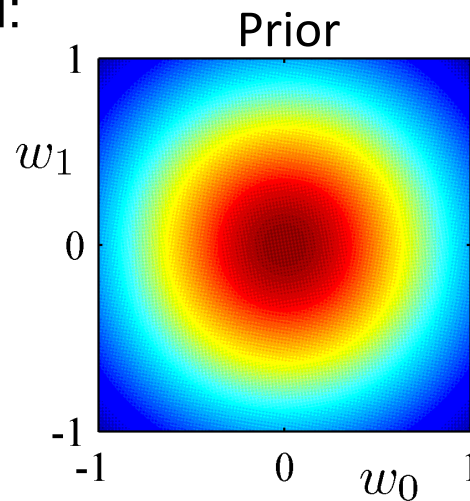
- Consider a linear model of the form: $y(x, \mathbf{w}) = w_0 + w_1 x$.
- The training data is generated from the function $f(x, \mathbf{a}) = a_0 + a_1 x$ with $a_0 = 0.3$; $a_1 = 0.5$, by first choosing x_n uniformly from $[-1; 1]$, evaluating $f(x, \mathbf{a})$, and adding a small Gaussian noise.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

0 data points are observed:

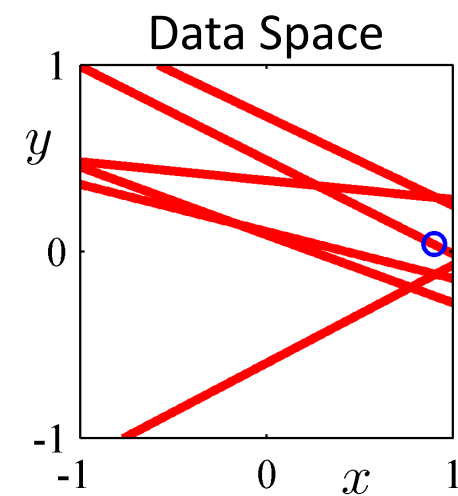
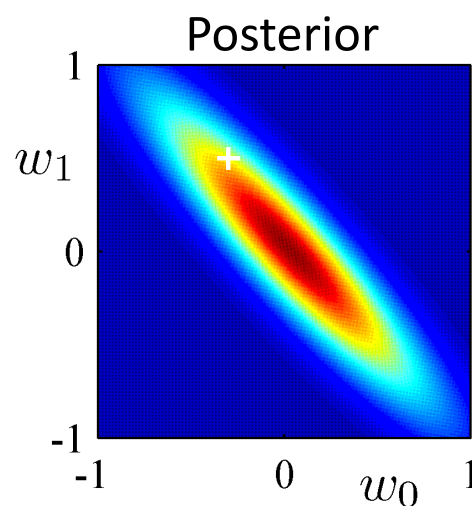
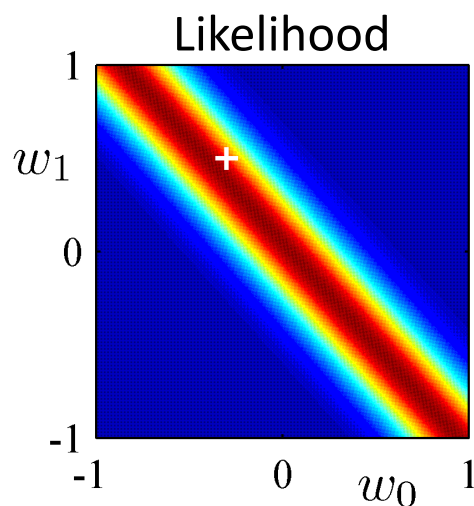


Bayesian Linear Regression

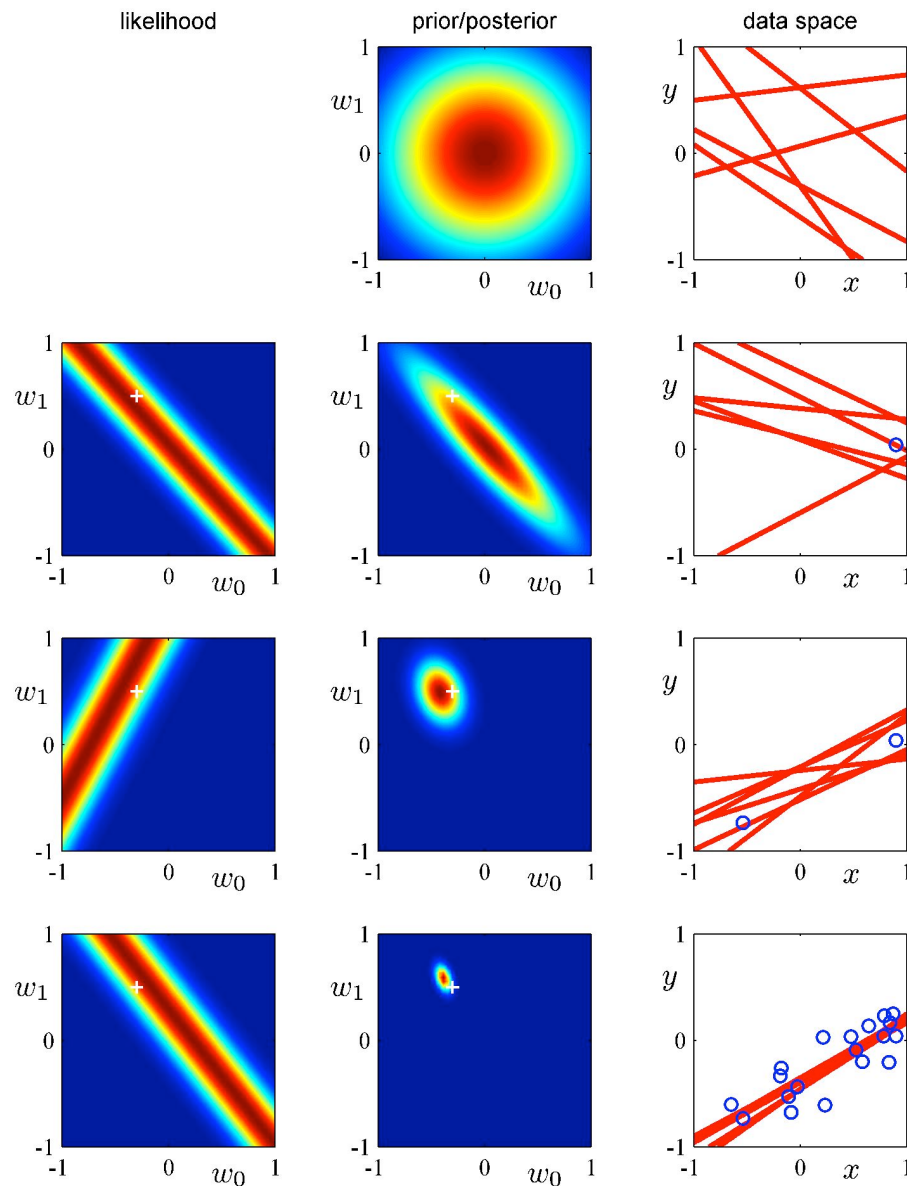
0 data points are observed:



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 $\mathbf{x}$ by **integrating over the posterior distribution**:

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

where

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

Noise in the
target values

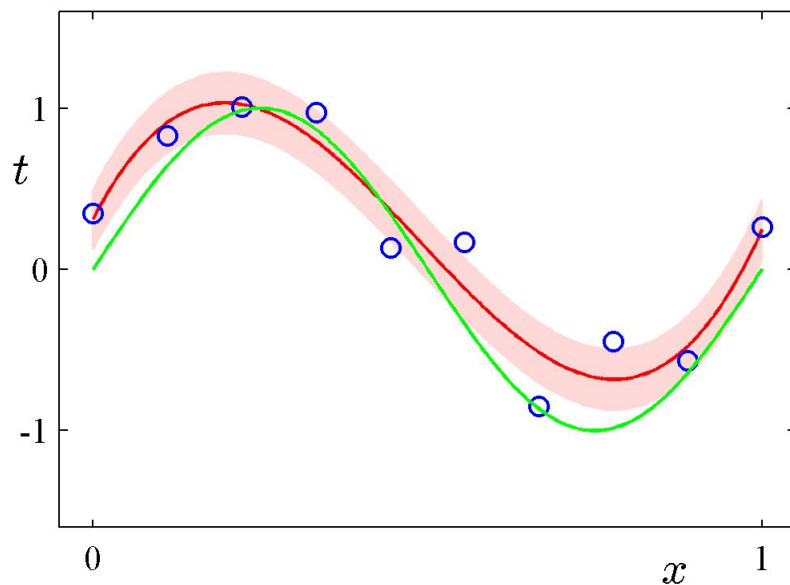
Uncertainty
associated with
parameter values.

$\mathbf{m}_N$	$=$	$\beta \mathbf{S}_N \Phi^T \mathbf{t}$
$\mathbf{S}_N^{-1}$	$=$	$\alpha \mathbf{I} + \beta \Phi^T \Phi.$

- 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 .

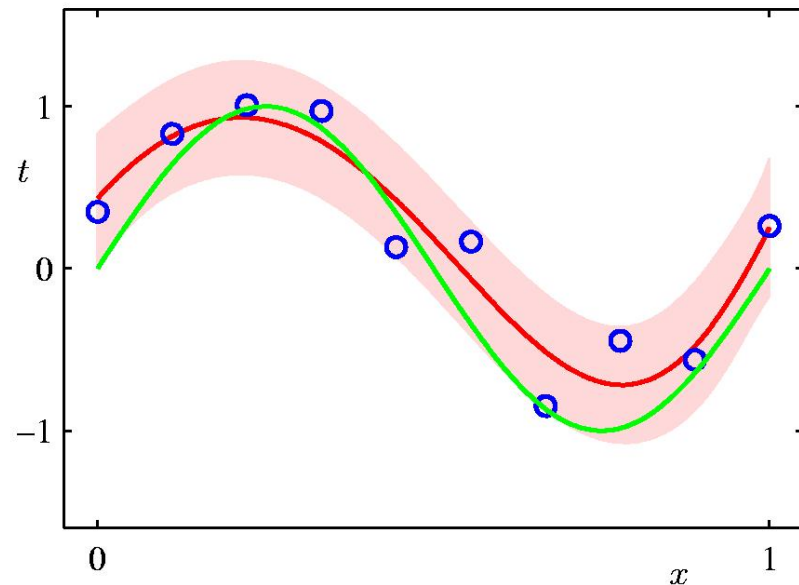
Predictive Distribution: Bayes vs. ML

Predictive distribution based on maximum likelihood estimates



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

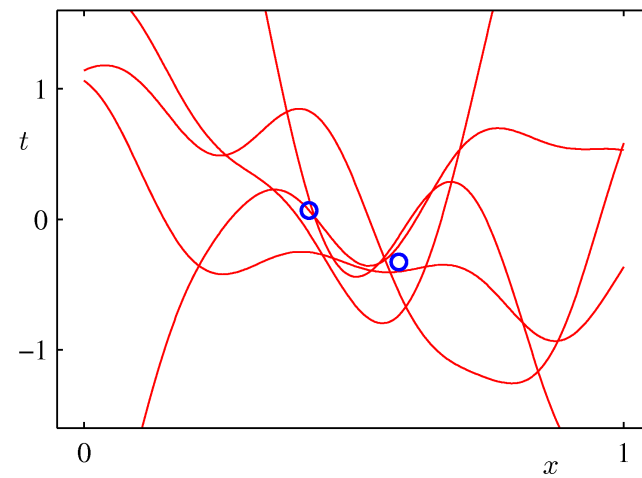
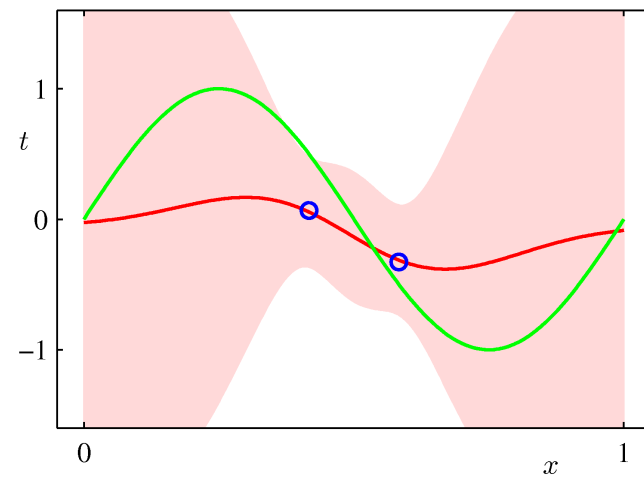
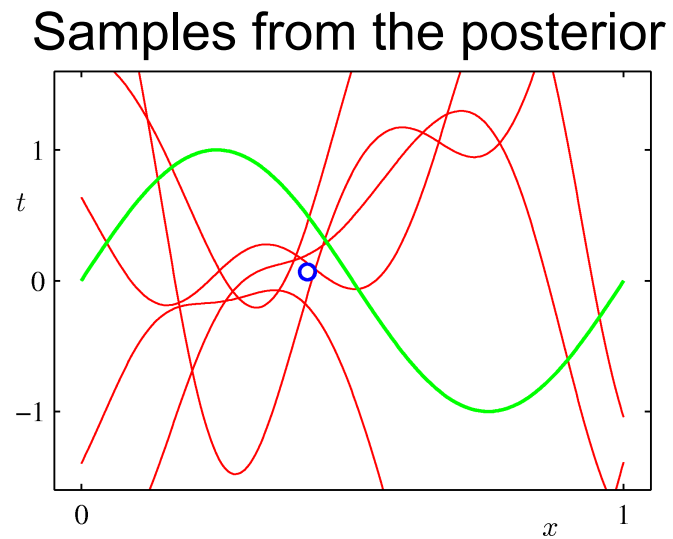
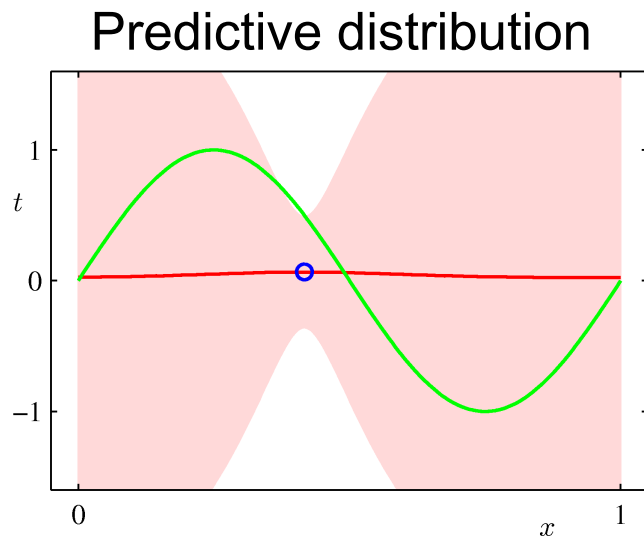
Bayesian predictive distribution



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

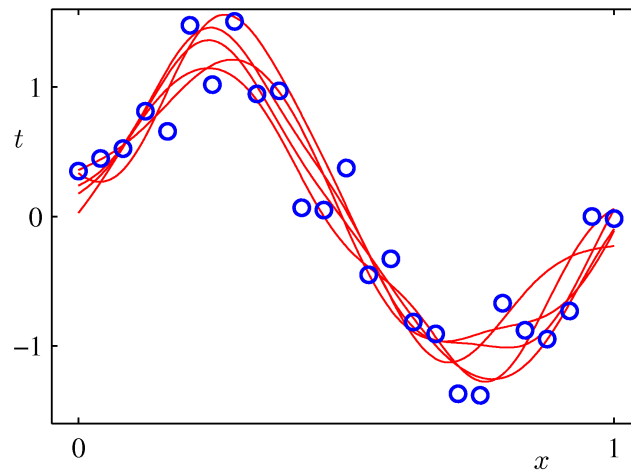
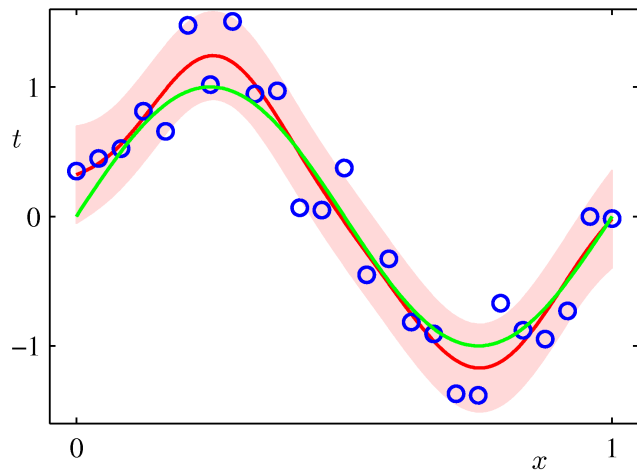
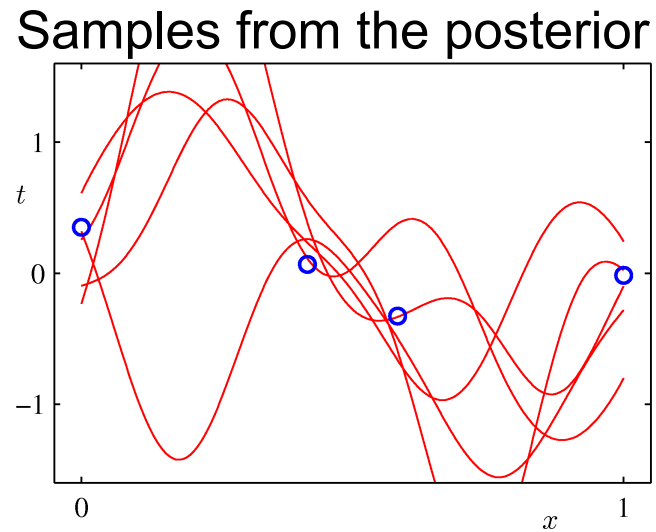
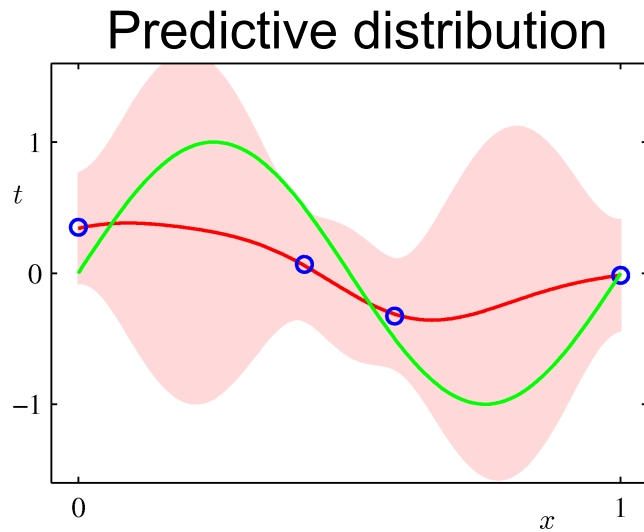
Predictive Distribution

Sinusoidal dataset, 9 Gaussian basis functions.



Predictive Distribution

Sinusoidal dataset, 9 Gaussian basis functions.



Gamma-Gaussian Conjugate Prior

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

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

where the Gamma distribution is given by:

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

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

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

Equivalent Kernel

- The predictive mean can be written as:

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

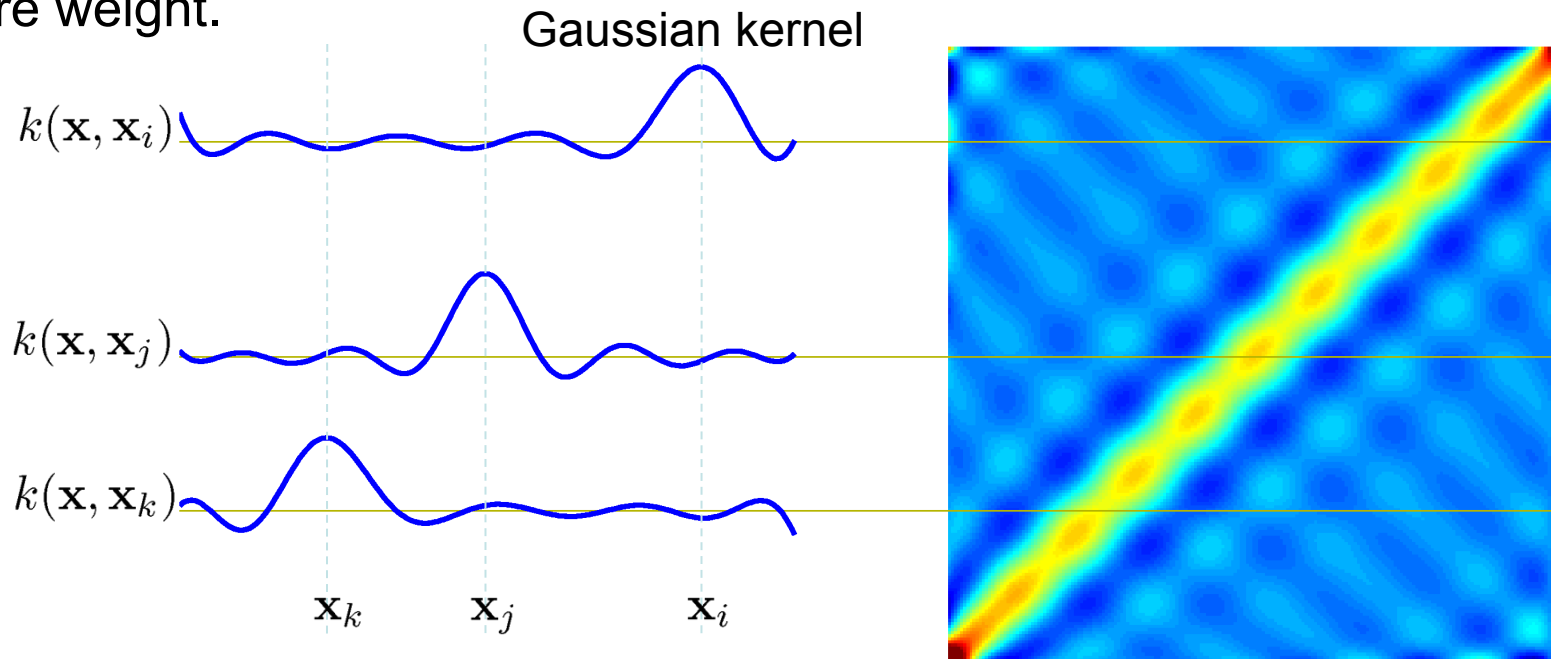
$$\begin{aligned}\mathbf{m}_N &= \beta \mathbf{S}_N \Phi^T \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \Phi^T \Phi.\end{aligned}$$

Equivalent kernel
or smoother
matrix.

- The mean of the predictive distribution at a time $\mathbf{x}$ 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 $\mathbf{x}$ and $\mathbf{x}_n$; nearby $\mathbf{x}_n$ carry more weight.



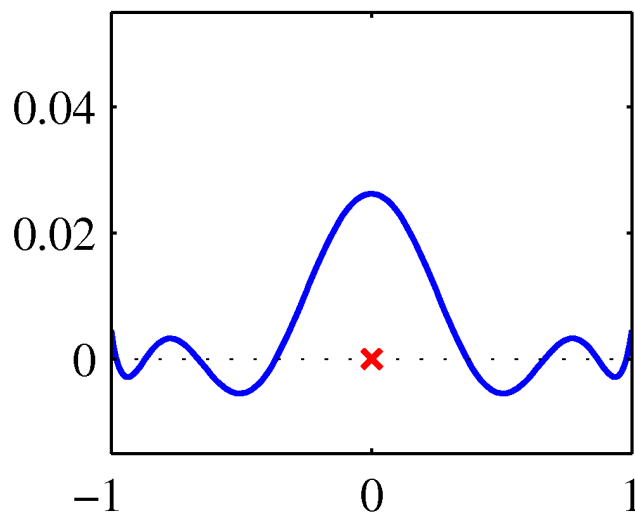
- The kernel as a covariance function:

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

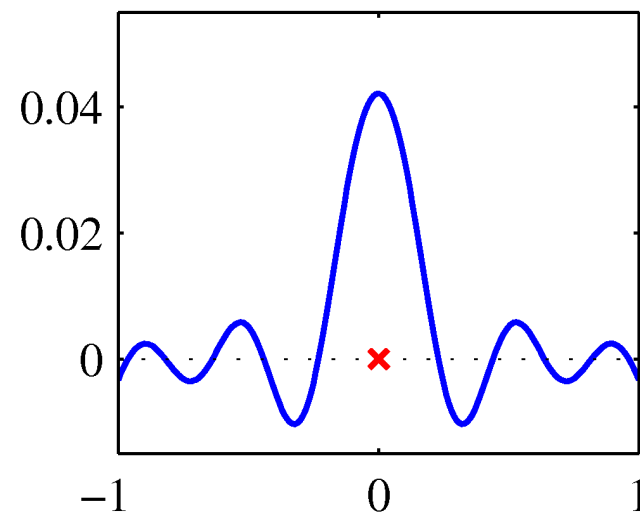
- We can avoid the use of basis functions and define the kernel function directly, leading to *Gaussian Processes*.

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 $i = 1, 2, \dots, L$, using a training set $\mathcal{D}$.
- We **specify the prior distribution** over the different models $p(\mathcal{M}_i)$.
- Given a training set $\mathcal{D}$, 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**:

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

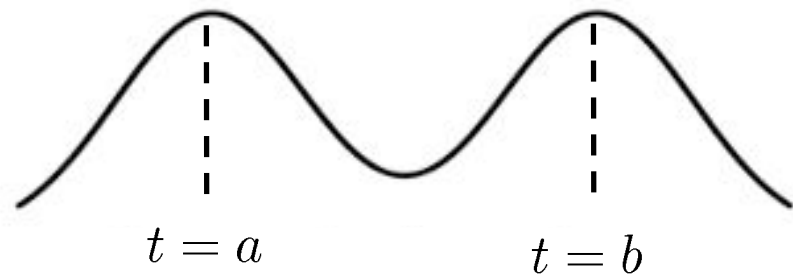
Bayesian Model Comparison

- 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 $\mathbf{w}$, the **model evidence** can be computed as follows:

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

- 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

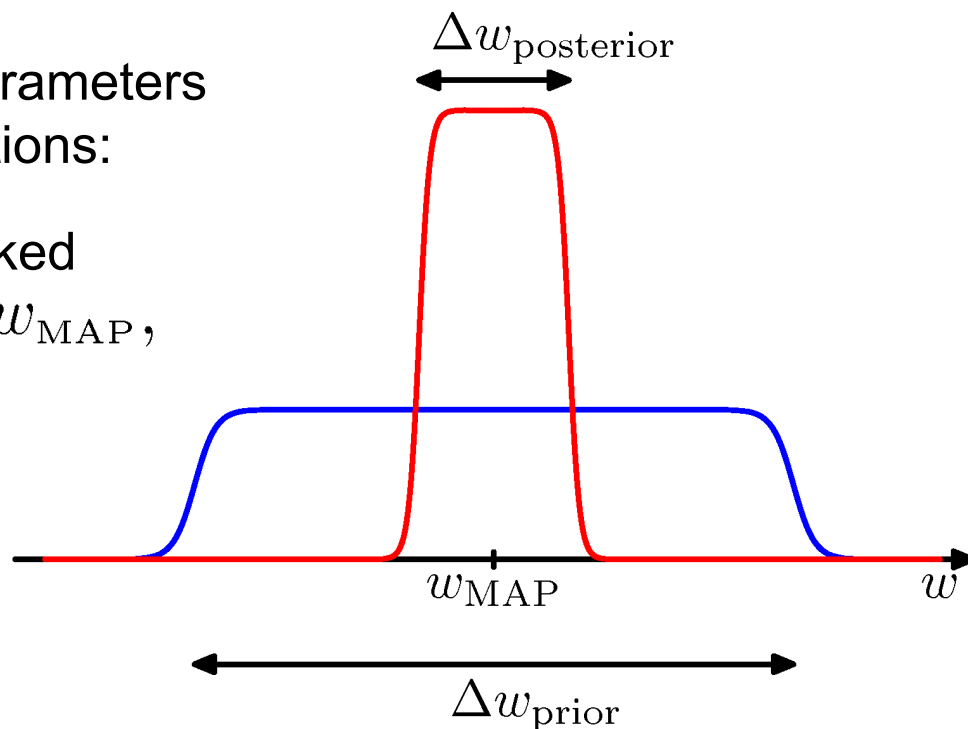
- We next get some insight into the model evidence by making simple approximations.

- 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 $\Delta w_{\text{posterior}}$

- Assume that **the prior is flat** with width Δw_{prior}

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w) dw$$
$$\simeq p(\mathcal{D}|w_{\text{MAP}}) \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}$$



Bayesian Model Comparison

- Taking the logarithms, we obtain:

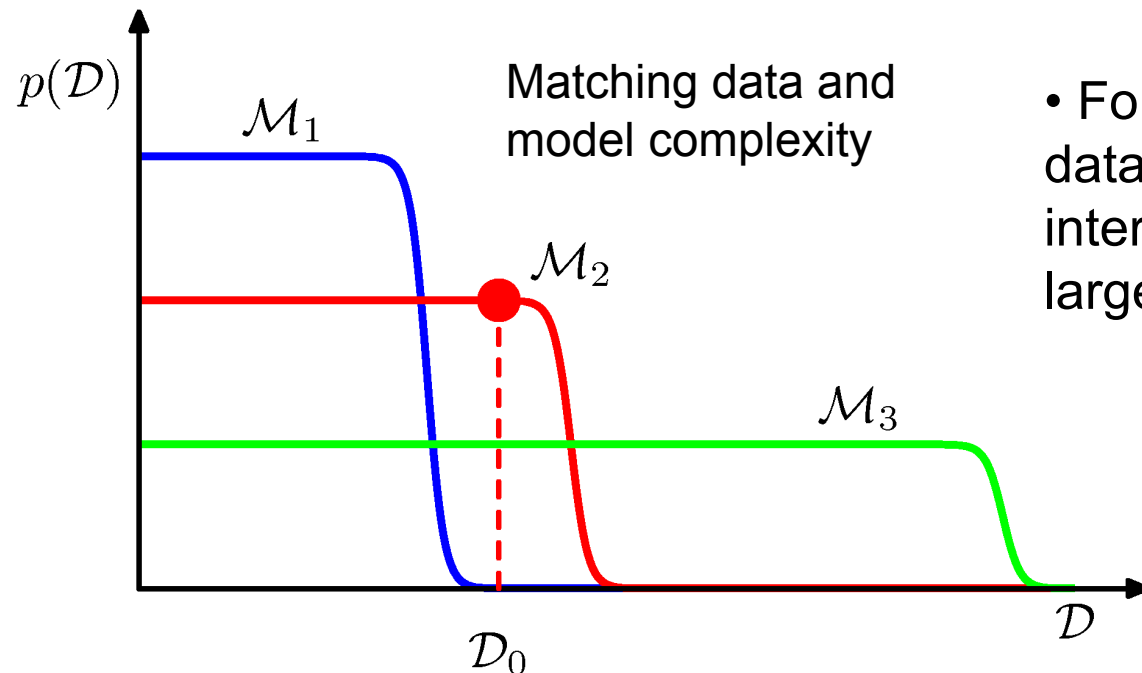
$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|w_{\text{MAP}}) + \underbrace{\ln \left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right)}_{\text{Negative}}.$$

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

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

- 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



- 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:

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

Diagram illustrating the components of the fully Bayesian predictive distribution integral:

- Likelihood** (red text): $p(t^* | \mathbf{x}^*, \mathbf{w}, \beta)$.
 - target and input on test case (blue text, with arrows pointing to t^* and $\mathbf{x}^*$)
- posterior over weights** (red text): $p(\mathbf{w} | \mathcal{D}, \alpha, \beta)$.
 - precision of output noise (blue text, with arrow pointing to β)
- posterior over hyperparameters** (red text): $p(\alpha, \beta | \mathcal{D})$.
 - precision of the prior (blue text, with arrow pointing to α)
 - training data: inputs and targets (blue text, with arrow pointing to $\mathcal{D}$)

- 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:

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

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

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

where $(\hat{\alpha}, \hat{\beta})$ 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 $(\hat{\alpha}, \hat{\beta})$ 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:

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

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

- We can write the evidence function in the form:

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

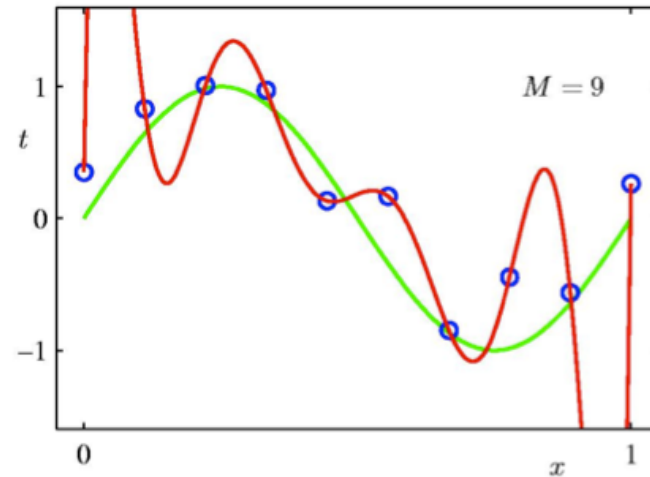
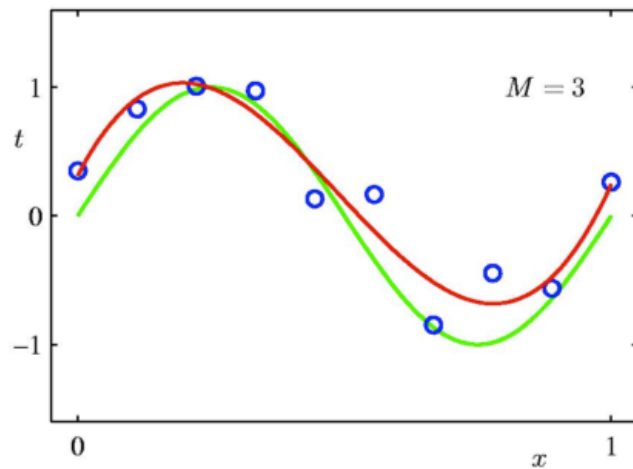
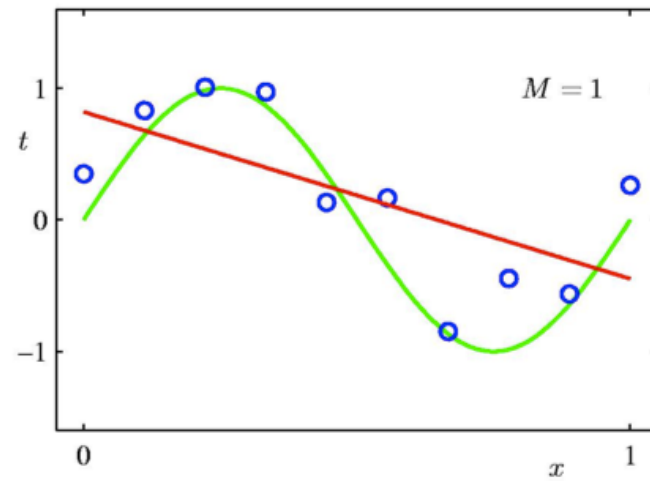
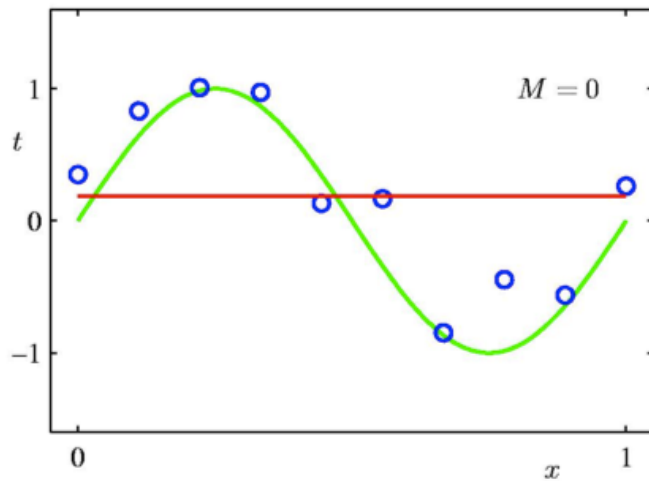
where

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

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

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

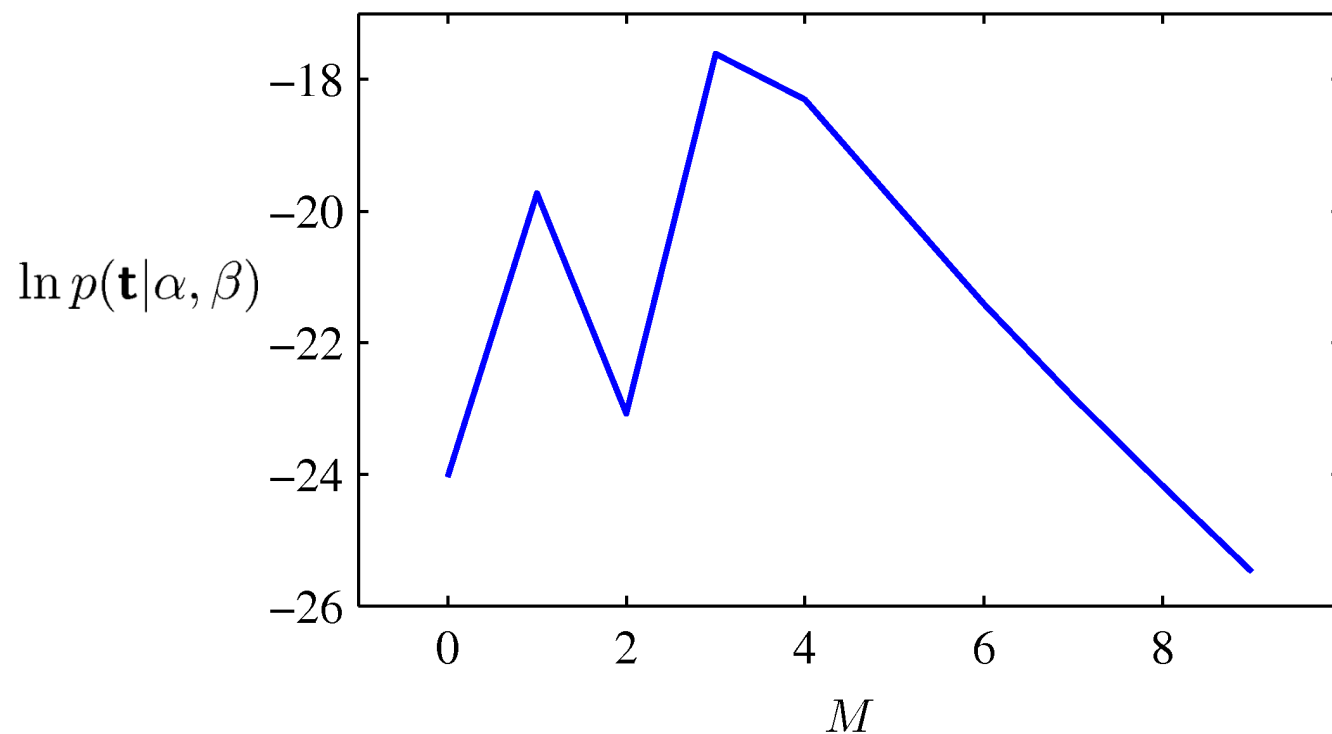
Some Fits to the Data



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

Evidence Approximation

Using sinusoidal data, M^{th} 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:

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

Precision matrix of the
Gaussian posterior
distribution



- Therefore the matrix:

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

has eigenvalues $\alpha + \lambda_i$.

- 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

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

- 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:

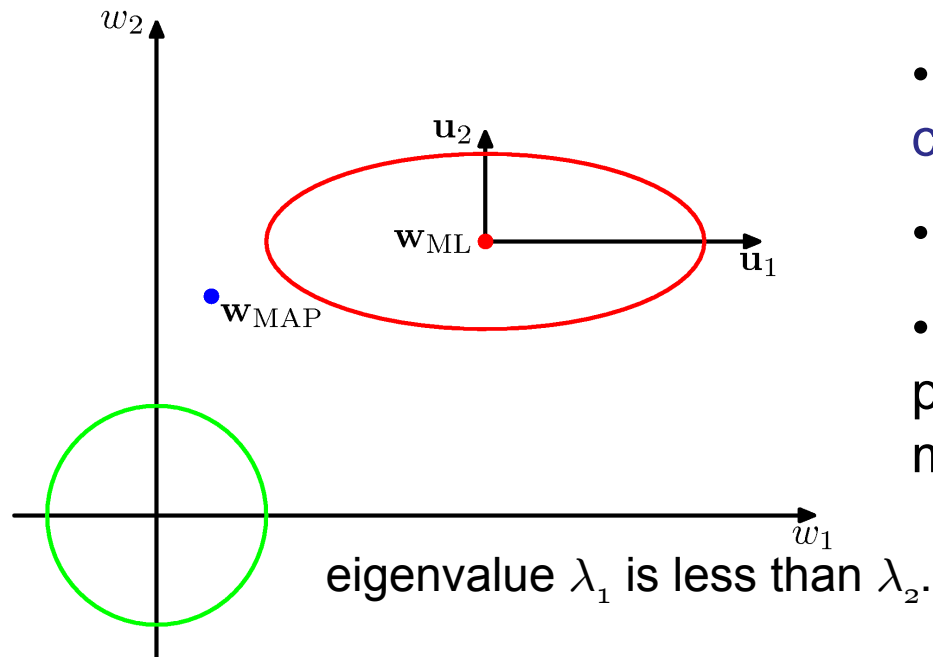
$$\alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}_N}; \quad \gamma = \sum_i \frac{\lambda_i}{\lambda_i + \alpha}; \quad \begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \Phi^T \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \Phi^T \Phi. \end{aligned}$$

- Similarly:

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

Effective Number of Parameters

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



- 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_i will be close to its maximum likelihood. The ratio:

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

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

$$\alpha = \frac{M}{\mathbf{m}_N^T \mathbf{m}_N}$$

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

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

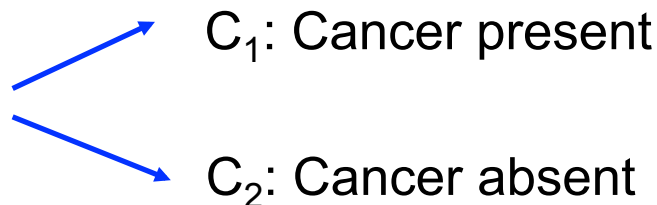
- So far, we have looked at the linear models for regression that have particularly simple analytical and computational properties.
- We will now look at analogous class of models for solving classification problems.
- We will also look at the Bayesian treatment of linear models for classification.

Classification

- The goal of classification is to assign an input $\mathbf{x}$ into one of K discrete classes C_k , where $k=1,\dots,K$.
- Typically, each input is assigned only to one class.
- **Example:** The input vector $\mathbf{x}$ 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 .



$\mathbf{x}$ -- set of pixel intensities



Linear Classification

- The goal of classification is to assign an input $\mathbf{x}$ into one of K discrete classes C_k , where $k=1,\dots,K$.
- 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 $y(\mathbf{x}, \mathbf{w}) = \text{const}$, so that $\mathbf{x}^T \mathbf{w} + w_0 = \text{const}$, and hence **the decision surfaces are linear functions of $\mathbf{x}$, 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.

Next Lecture

Linear Models for Classification:

- Fisher's Linear Discriminant
- Probabilistic Generative Models
- Probabilistic Discriminative Models, Logistic Regression
- Bayesian Logistic Regression
- Laplace Approximation