STA414/2104

Statistical Methods for Machine Learning II

Murat A. Erdogdu

Department of Computer Science
Department of Statistical Sciences

Lecture 3

UNIVERSITY OF TORONTO

Announcements

- Homework 1 - v0 is released!
  - Due on Feb 8, 13:59.
  - You will submit through Crowdmark.
  - Come to office hours (TA OHs will be announced on course webpage).

Last Time

- Maximum likelihood estimation
- Exponential families
- Important distributions

Today

- Regularization
- Bayesian linear regression
- Posterior calculations

Linear Basis Function Models

- Remember, the simplest linear model for regression:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_d x_d = w_0 + \sum_{j=1}^{d} w_j x_j,$$

where $\mathbf{x} = (x_1, x_2, ..., x_d)^T$ a d-dimensional input vector (covariates).

Key property: linear function of the parameters $w_0, w_1, ..., w_d$

- However, it is also a linear function of input variables.
- Instead consider:

$$y(\mathbf{x}, \mathbf{w}) = w_0 \phi_0(\mathbf{x}) + w_1 \phi_1(\mathbf{x}) + ... + w_{M-1} \phi_{M-1}(\mathbf{x}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}),$$

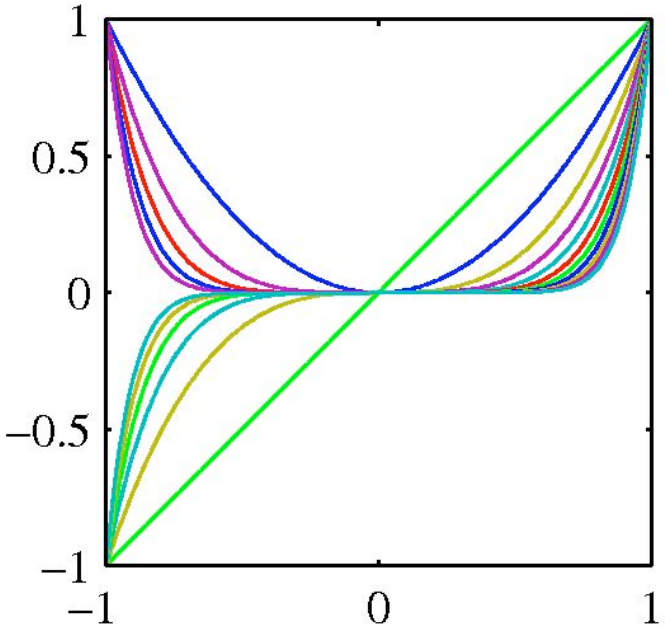
where $\phi_j(\mathbf{x})$ are known as basis functions.

- Typically $\phi_0(\mathbf{x}) = 1$ so that w_0 acts as a bias (or intercept).
- In the simplest case, we use linear bases functions: $\phi_j(\mathbf{x}) = x_j$.
- Using nonlinear basis allows the functions $y(\mathbf{x}, \mathbf{w})$ to be nonlinear functions of the input space.

Linear Basis Function Models

Polynomial basis functions:

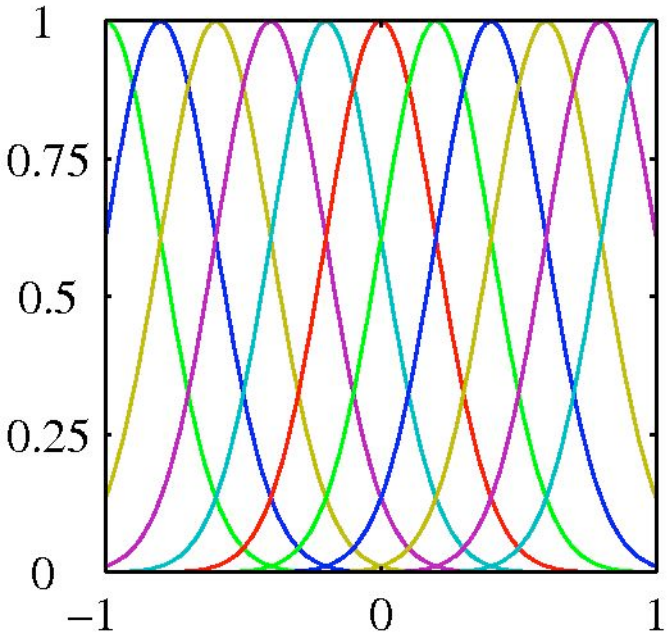
$$\phi_j(x) = x^j.$$



Basis functions are global: small changes in x affect all basis functions.

Gaussian basis functions:

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

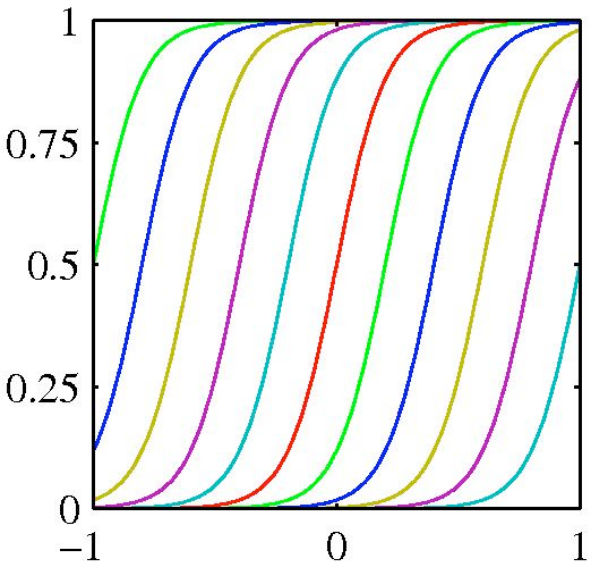


Basis functions are local: small changes in x only affect nearby basis functions.
μ_j and s control location and scale (width).

Linear Basis Function Models

Sigmoidal basis functions

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



Basis functions are local: small changes in x only affect nearby basis functions.
μ_j and s control location and scale.

• Decision boundaries will be linear in the feature space $\boldsymbol{\phi}$, but would correspond to nonlinear boundaries in the original input space x.

• Classes that are linearly separable in the feature space $\boldsymbol{\phi}(\mathbf{x})$ need not be linearly separable in the original input space.

Maximum Likelihood Estimation

• As before, assume observations arise from a deterministic function with an additive Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon,$$

which we can write as:

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

• Given observed inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, and corresponding target values $\mathbf{t} = [t_1, t_2, ..., t_N]^T$ under independence assumption, 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 $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), ..., \phi_{M-1}(\mathbf{x}))^T$.

Maximum Likelihood Estimation

Taking the logarithm, we obtain:

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

$$= -\frac{\beta}{2}\underbrace{\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{E_D(\mathbf{w})} + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi).$$

$E_D(\mathbf{w})$ sum-of-squares error function

Differentiating and setting to zero yields:

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

Maximum Likelihood Estimation

Differentiating and setting to zero yields:

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

Solving for w, we get:

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

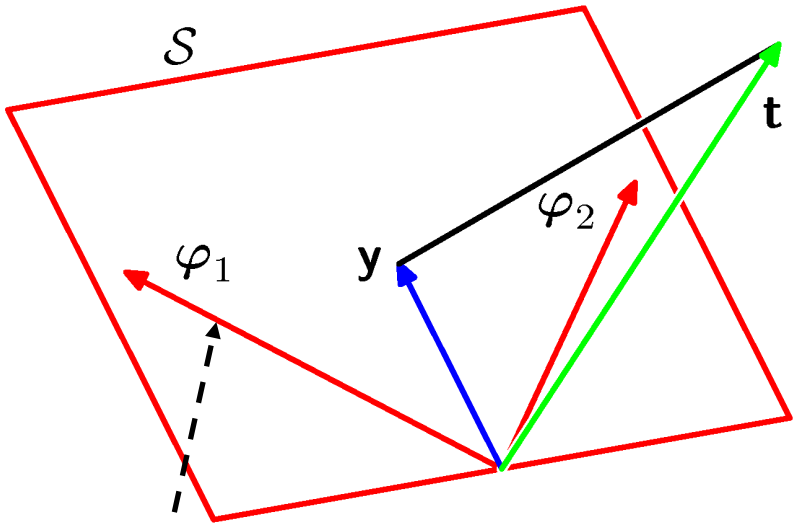
The Moore-Penrose pseudo-inverse, $\boldsymbol{\Phi}^{\dagger}$.

where $\boldsymbol{\Phi}$ is known as the design matrix:

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

Geometry of Least Squares

- Consider an N-dimensional space, so that $\mathbf{t} = [t_1, t_2, ..., t_N]^T$ is a vector in that space.
- Each basis function $\phi_j(\mathbf{x}_n)$, evaluated at the N data points, can be represented as a vector φ_j in the same space.
- If M is less than N, then the M basis function $\phi_j(\mathbf{x}_n)$, will span a linear subspace S of dimensionality M.
- Define: $\mathbf{y} = \mathbf{\Phi w}$
- The sum-of-squares error is equal to the squared Euclidean distance between y and t (up to a factor of 1/2).



$$\min \ \|\mathbf{t} - \mathbf{y}\|^2$$

$$\mathbf{y} = \mathbf{\Phi w} = \sum_j \varphi_j w_j$$

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

The solution corresponds to the orthogonal projection of t onto the subspace S.

$$\mathbf{y} = \mathbf{\Phi w_{ML}}.$$

Regularized Least Squares

- Let us consider the following error function:

$$\min \; E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Training error + Regularization term

λ is called the regularization parameter (or penalty level).

- Using sum-of-squares error function with a quadratic penalization term, we obtain:

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

which is minimized by setting:

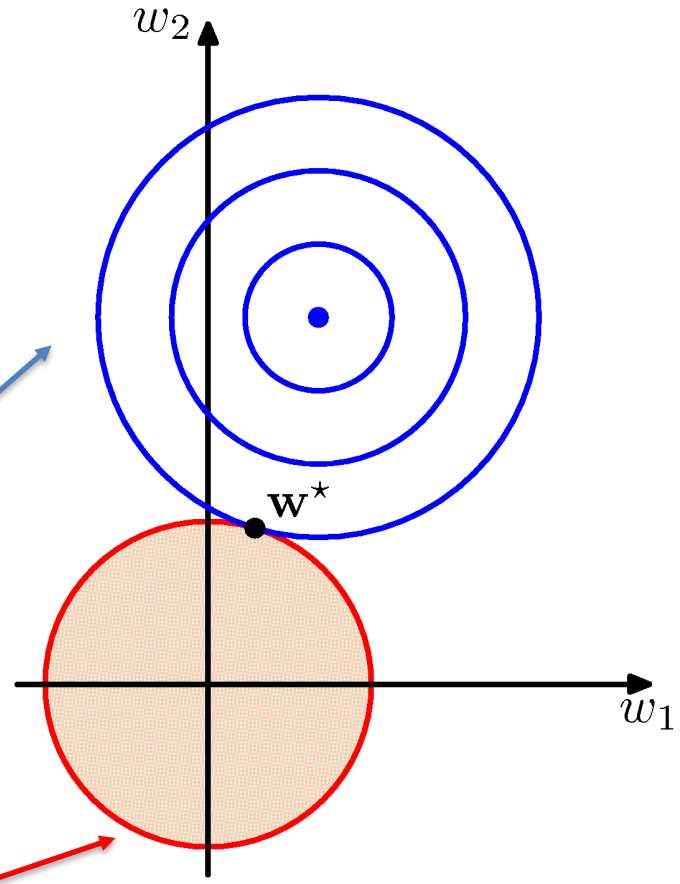
Ridge regression

$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

The solution adds a positive constant to the diagonal of $\boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Phi}$. This makes the problem nonsingular, even if $\boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Phi}$ is not of full rank (e.g. when the number of training examples is less than the number of basis functions).

Effect of Regularization

- The blue curve is the contours of the error function.
- The red curve is the feasible set for the parameters.
- The regularizer shrinks model parameters to zero.



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

$$\text{subject to: } \|\mathbf{w}\|^2 \leq \gamma(\lambda)$$

Equivalent formulations

- We can write the problem $\min\ E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$

  Training error + Regularization term

- equivalently as $\min\ E_D(\mathbf{w})$

  $$\text{subject to: } E_W(\mathbf{w}) \leq \gamma(\lambda)$$

- $\gamma(\lambda)$ depends on λ, but different than λ.

For example, we can write ridge regression,

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

$$\|w\|_2^2 = \|w\|^2$$

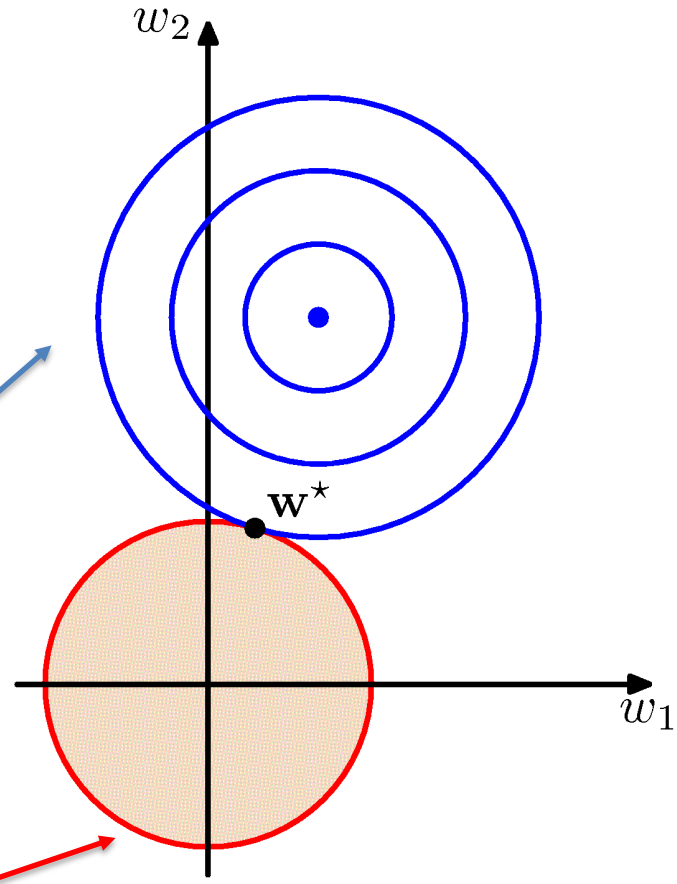
as

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

$$\text{subject to: } \|\mathbf{w}\|^2 \leq \gamma(\lambda)$$

Effect of Regularization

- The blue curve is the contours of the error function.
- The red curve is the feasible set for the parameters.
- The regularizer shrinks model parameters to zero.



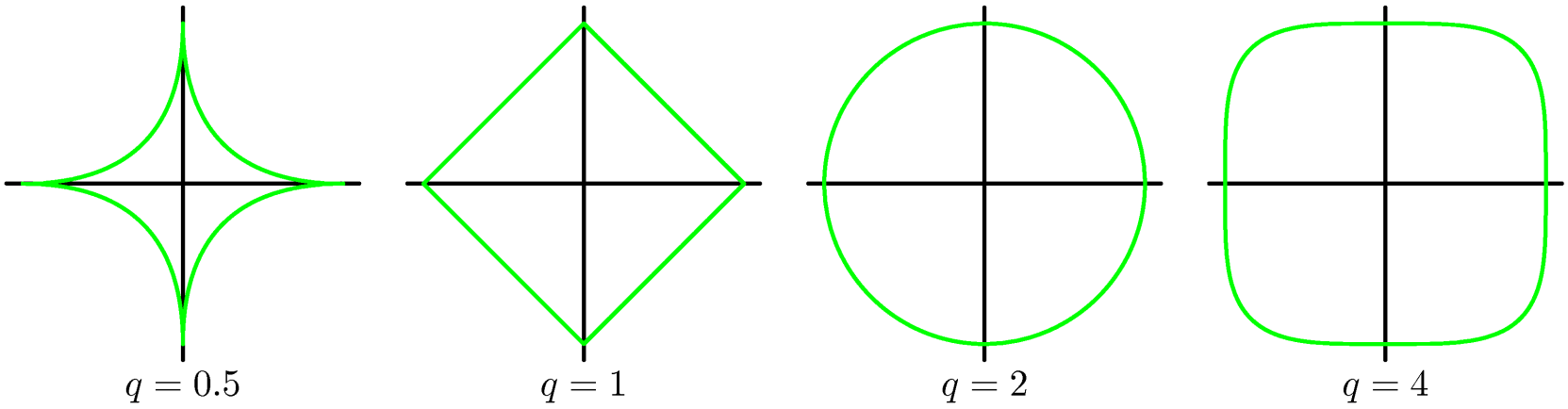
$$\min \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

$$\text{subject to: } \|\mathbf{w}\|^2 \leq \gamma(\lambda)$$

Other Regularizers

Using a more general regularizer, we get:

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Lasso Quadratic

$$\text{subject to: } E_W(\mathbf{w}) \leq \gamma(\lambda)$$

The Lasso

- Penalize the absolute value of the weights:

$$\mathbf{w}^{lasso} = \underset{\mathbf{w}}{\operatorname{argmin}} \left[\frac{1}{2} \sum_{n=1}^{N} \left(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{M-1} |w_j| \right].$$

- For sufficiently large λ, some of the coefficients will be driven to exactly zero, leading to a sparse model.

- The above formulation is equivalent to:

$$\mathbf{w}^{lasso} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2} \underbrace{\sum_{n=1}^{N} \left(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right)^2}_{\text{sum-of-squares error}}, \text{ subject to } \sum_{j=1}^{M-1} |w_j| \leq \tau.$$

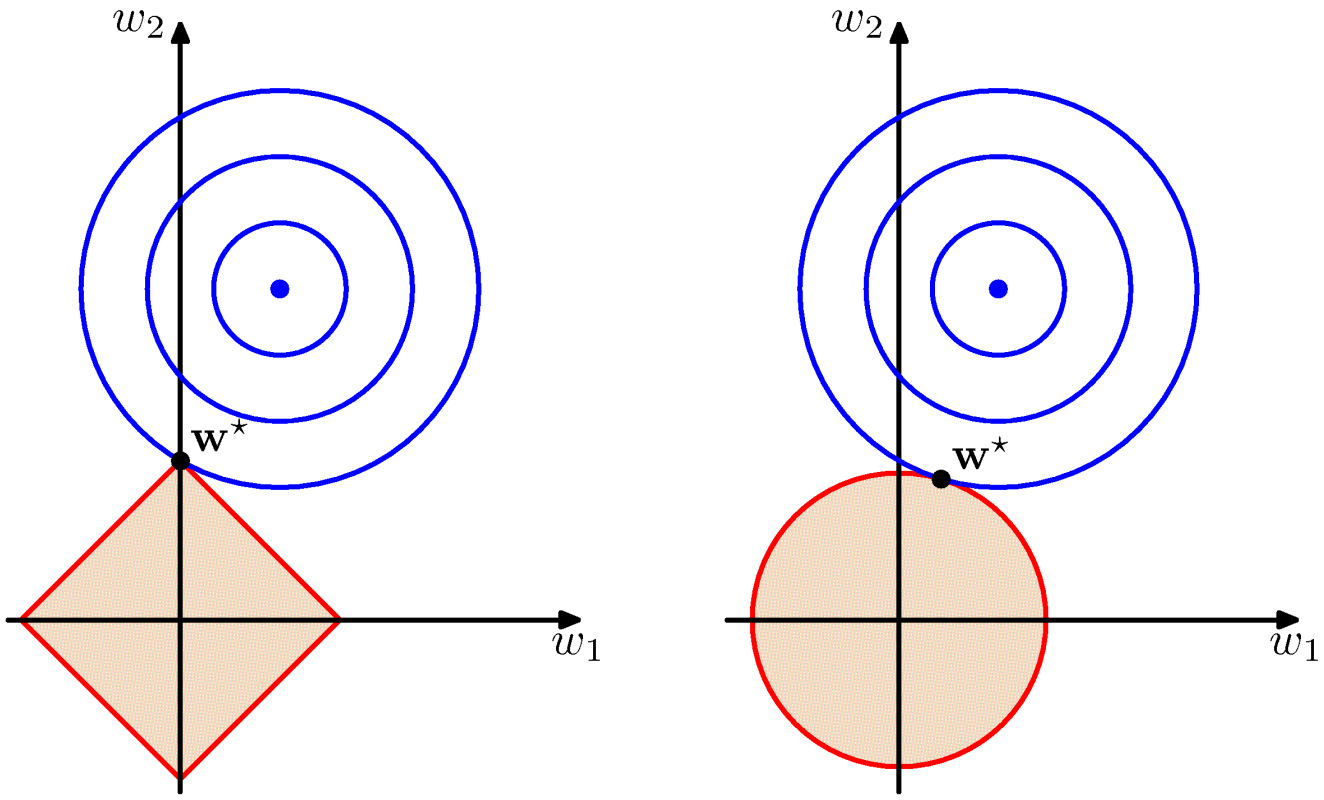
$\|w\|_1$

- The two approaches are related using Lagrange multiplies.

- The Lasso solution is a convex problem and can be solved efficiently.

Lasso vs. Quadratic Penalty

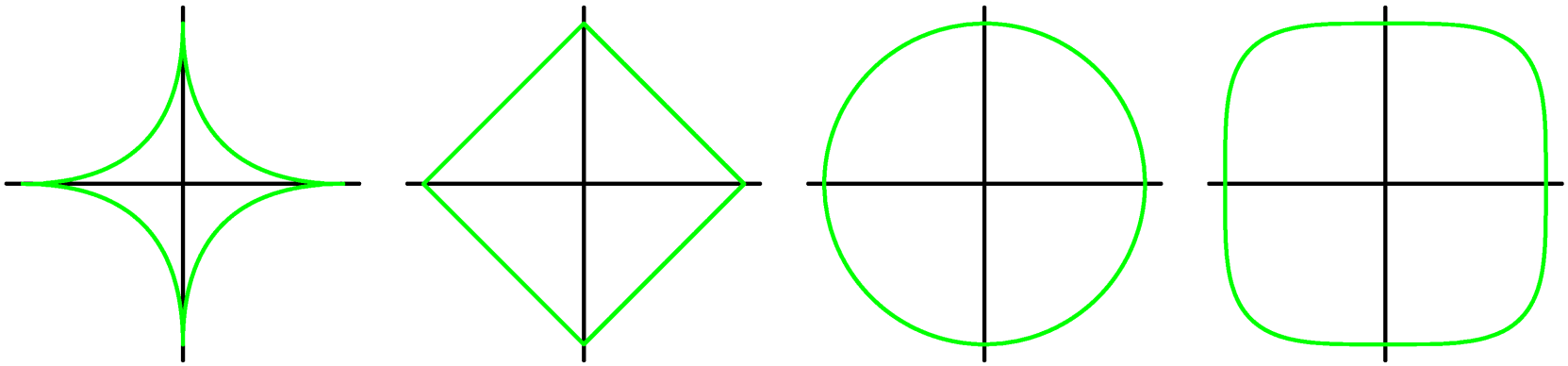
Lasso tends to generate sparser solutions compared to a quadratic regularizer (often referred to as L_1 and L_2 regularizers).



Sparsity induced by regularizers

Using a more general regularizer, we get:

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Lasso Quadratic

$$\text{subject to: } E_W(\mathbf{w}) \leq \gamma(\lambda)$$

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:

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

MAP: Maximum A posteriori Probability

- In Maximum A posteriori Probability (MAP) estimation, we maximize the posterior

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

$$\mathbf{w}^{\mathrm{MAP}} = \arg\max_{\mathbf{w}} \frac{p(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})},$$
$$= \arg\max_{\mathbf{w}} p(\mathcal{D}|\mathbf{w})P(\mathbf{w})$$

MLE vs MAP

- In Maximum Likelihood Estimation, we maximize the likelihood

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

Note that sometimes D and w is swapped to denote likelihood! But notation for likelihood is typically L(w|D), not p(D|w).

- In Maximum A posteriori Probability (MAP) estimation, we maximize the posterior

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

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

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 (coming up soon).

• 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 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).

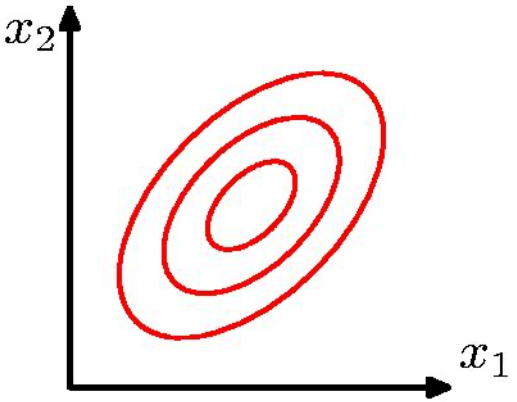
- Monte Carlo integration: Once we have a sample from the posterior distribution, we can do many things. The dominant current approach is Markov Chain Monte Carlo (MCMC) -- simulate a Markov chain that converges to the posterior distribution. It can be applied to a wide variety of problems.

- Variational approximation: A cleverer way to approximate the posterior. It often works much faster compared to MCMC. But often not as general as MCMC.

Multivariate Gaussian Distribution

- For a D-dimensional vector x, the Gaussian distribution takes form:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$



which is governed by two parameters:

- μ is a D-dimensional mean vector.
- $\boldsymbol{\Sigma}$ is a D by D covariance matrix.

and $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$.

- Note that the covariance is a symmetric **positive definite matrix.**
- Positive definite matrix: $\forall u \in \mathbb{R}^D \quad u^T\boldsymbol{\Sigma}u > 0$
- Positive semidefinite matrix: $\forall u \in \mathbb{R}^D \quad u^T\boldsymbol{\Sigma}u \geq 0$

Means "every" vector in $\mathbb{R}^D$

Moments of the Gaussian Distribution

- The expectation of x under the Gaussian distribution:

$$\begin{aligned}\mathbb{E}[\mathbf{x}] &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}\mathbf{x}\,\mathrm{d}\mathbf{x} \\ &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}\mathbf{z}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}\right\}(\mathbf{z}+\boldsymbol{\mu})\,\mathrm{d}\mathbf{z}\end{aligned}$$

Change of variable:

$$\boldsymbol{x} = \boldsymbol{z} + \boldsymbol{\mu}$$

The term in z in the factor **(z + μ)** will vanish by symmetry, or simply by noticing it is the expectation of a centered Gaussian.

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

Moments of the Gaussian Distribution

- The second order moments of the Gaussian distribution:

$$\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{\Sigma}$$

- The covariance is given by:

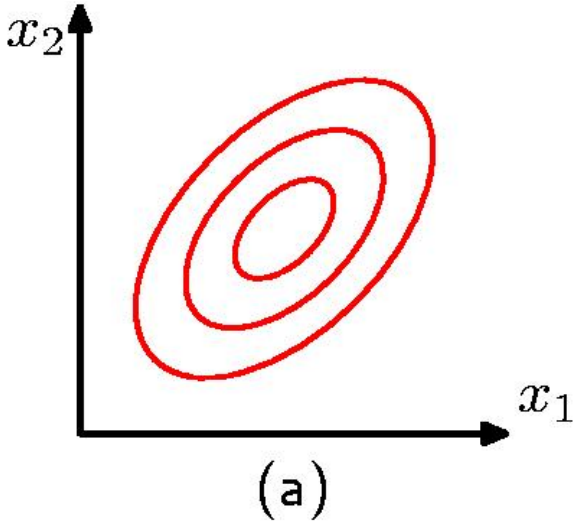
$$\mathrm{cov}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\mathrm{T}}\right] = \boldsymbol{\Sigma}$$

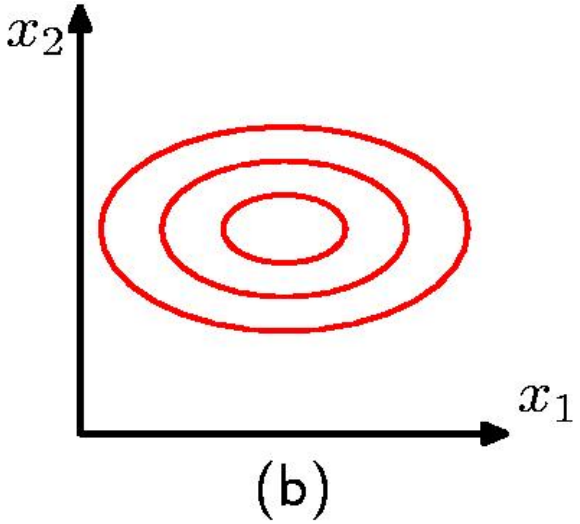
$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

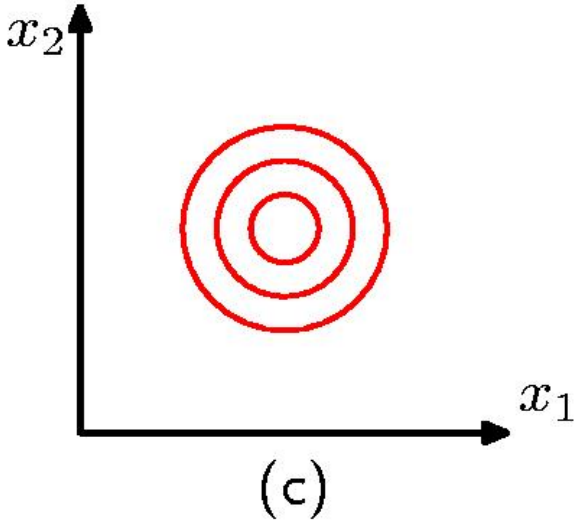
$$\mathbb{E}[\|\mathbf{x}\|^2] = \sum_i \mathbb{E}[\mathbf{x}_i^2] = \mathrm{Tr}(\mathbb{E}[\mathbf{x}\mathbf{x}^T]) = \|\mu\|^2 + \mathrm{Tr}(\boldsymbol{\Sigma})$$

Moments of the Gaussian Distribution

- Contours of constant probability density:







Covariance matrix is of general form.

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

Diagonal, axis-aligned covariance matrix.

$$\begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix}$$

Spherical (proportional to identity) covariance matrix.

$$\begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{11} \end{bmatrix}$$

Geometry of the Gaussian Distribution

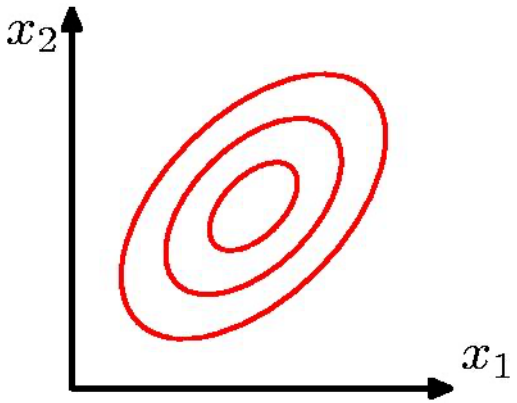
• For a D-dimensional vector x, the Gaussian distribution takes form:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

• Let us analyze the functional dependence of the Gaussian on x through the quadratic form:

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

• Here Δ is known as Mahalanobis distance between $\mathbf{x}$ and μ



• The Gaussian distribution will be constant on surfaces in x-space for which Δ is constant.

Propto for the Gaussian Distribution

- For a D-dimensional vector x, the Gaussian distribution takes form:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mu\right\}$$

- This is a density over the variable **x**. Everything else acts as normalizing constant. So we can drop terms that don't depend on **x**, and use the propto symbol.

- If for a vector **w**, we know that its density satisfies

$$p(\mathbf{w}|\mathbf{A},\mathbf{b}) \propto \exp\left\{-\frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b}\right\}$$

- then,

$$p(\mathbf{w}|\mathbf{A},\mathbf{b}) = \mathcal{N}(\mathbf{x}|\mathbf{A}^{-1}\mathbf{b},\mathbf{A}^{-1})$$

Bayesian Linear Regression

• Given observed inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, 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(w|x,t) are in the same family as the prior probability distribution p(w), 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:

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \beta) \propto \left[\prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1})\right]\mathcal{N}(\mathbf{w}|\mathbf{m_0}, \mathbf{S_0}).$$

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

- 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:

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

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

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

Bayesian Linear Regression

- Combining the prior together with the likelihood term:

$$p(\mathbf{w}|\mathbf{t},\mathbf{X},\beta) \propto \left[\prod_{n=1}^{N}\mathcal{N}(t_n|\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1})\right]\mathcal{N}(\mathbf{w}|\mathbf{m_0},\mathbf{S_0}).$$

$$\propto \exp\left\{-\frac{\beta}{2}\|\mathbf{t}-\boldsymbol{\Phi}\mathbf{w}\|^2\right\}\exp\left\{-\frac{1}{2}(\mathbf{w}-\mathbf{m}_0)^T\mathbf{S}_0^{-1}(\mathbf{w}-\mathbf{m}_0)\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\mathbf{w}^T\underbrace{(\beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\mathbf{S}_0^{-1})}_{=\mathbf{S}_N^{-1}}\mathbf{w}+\mathbf{w}^T\underbrace{(\beta\boldsymbol{\Phi}^T\mathbf{t}+\mathbf{S}_0^{-1}\mathbf{m}_0)}_{=\mathbf{S}_N^{-1}\mathbf{m}_N}\right\}$$

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

where

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

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1}+\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}.$$

Bayesian Linear Regression

• Consider a zero mean isotropic Gaussian prior, which is governed by a single precision parameter:

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

means diagonal covariance

for which the posterior is Gaussian with:

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

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

• If we consider an infinitely broad prior, α = 0, the mean m_N of the posterior distribution reduces to maximum likelihood value w_{ML}.

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

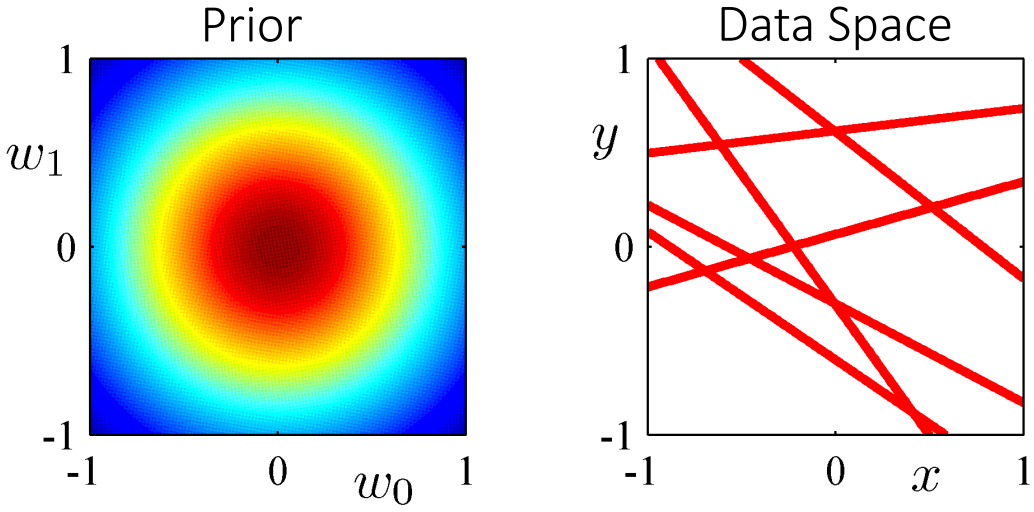
$$\ln p(\mathbf{w}|\mathcal{D}) = -\frac{\beta}{2} \sum_{n=1}^{N} \left(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right)^2 - \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{const.}$$

• Maximizing this posterior with respect to w is equivalent to minimizing the sum-of-squares error function with a quadratic regulation term (ridge regression).

Bayesian Linear Regression

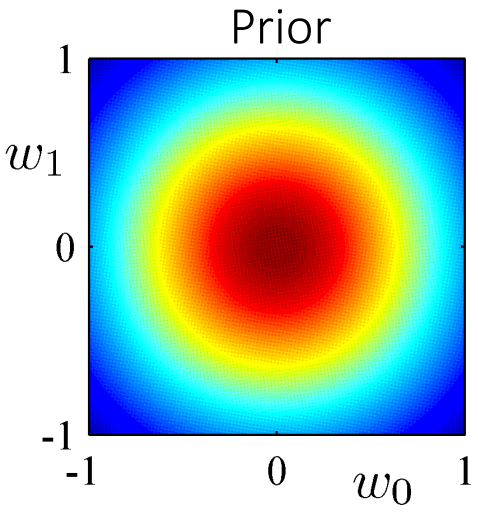
- Consider a linear model of the form: $y(x, \mathbf{w}) = w_0 + w_1 x.$
- The training data is generated from the function $f(x, \mathbf{a}) = a_0 + a_1 x$ with $a_0 = 0.3; a_1 = 0.5$, by first choosing x_n uniformly from [-1;1], evaluating $f(x, \mathbf{a}),$ and adding a small Gaussian noise.
- Goal: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

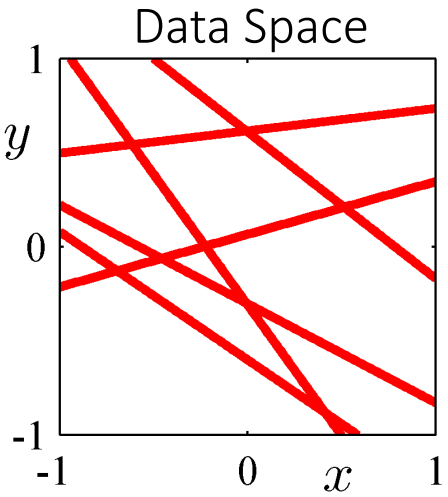
0 data points are observed:



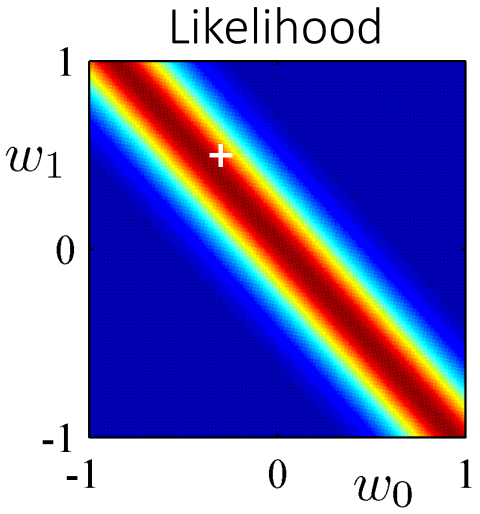
Bayesian Linear Regression

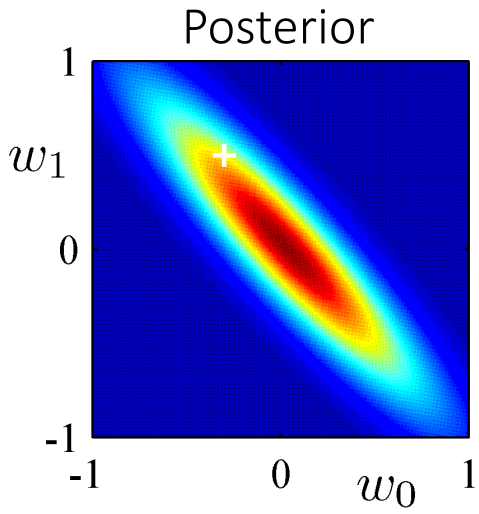
0 data points are observed:

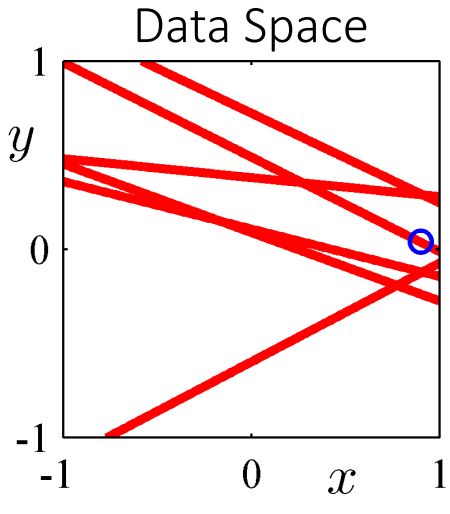




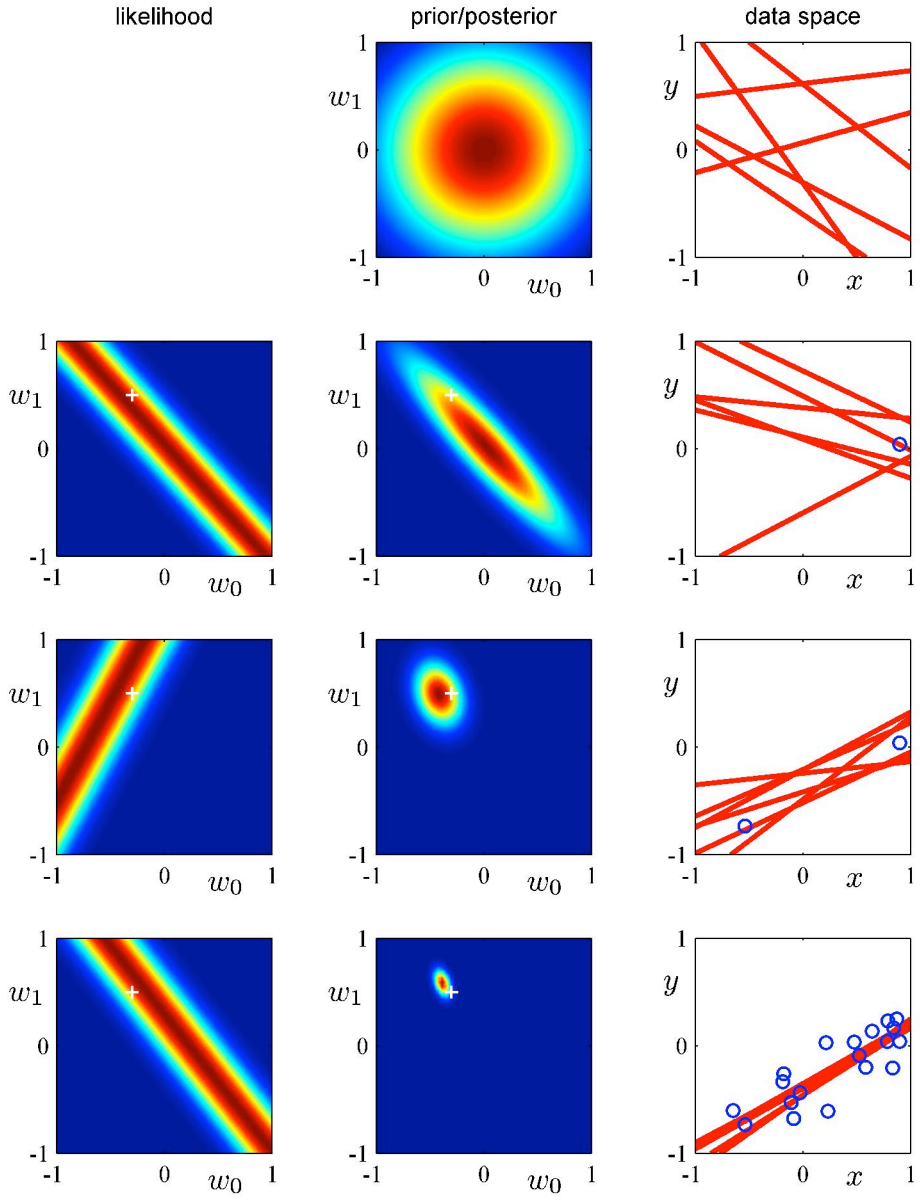
1 data point is observed:







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:

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

(exercise! – use p93, eq 2.115 of PRML)

where

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

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

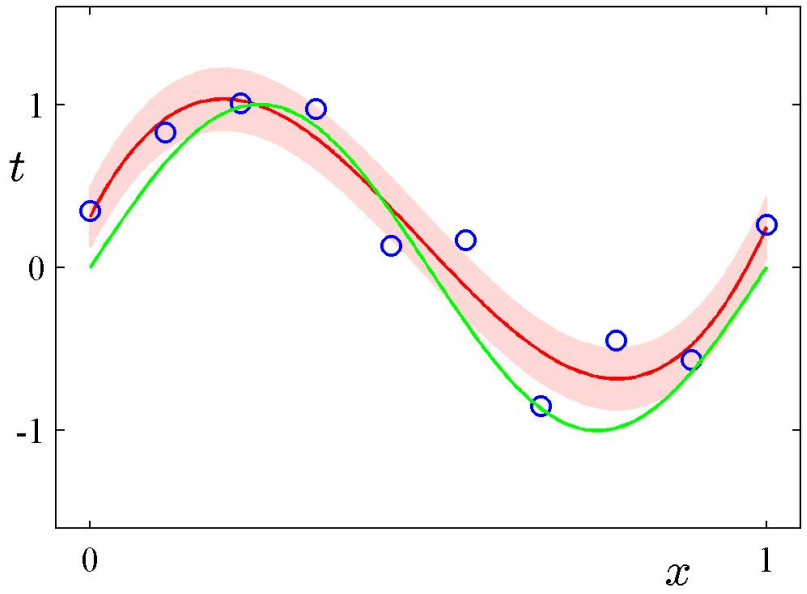
Noise in the target values

Uncertainty associated with parameter values.

• In the limit, as N -> infinity, the second term goes to zero.
• The variance of the predictive distribution arises only from the additive noise.

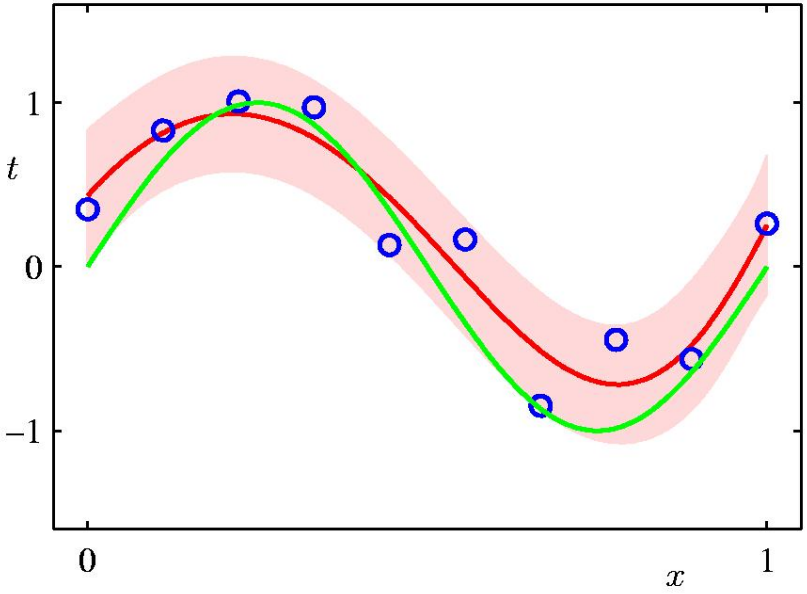
Predictive Distribution: Bayes vs. ML

Predictive distribution based on maximum likelihood estimates



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

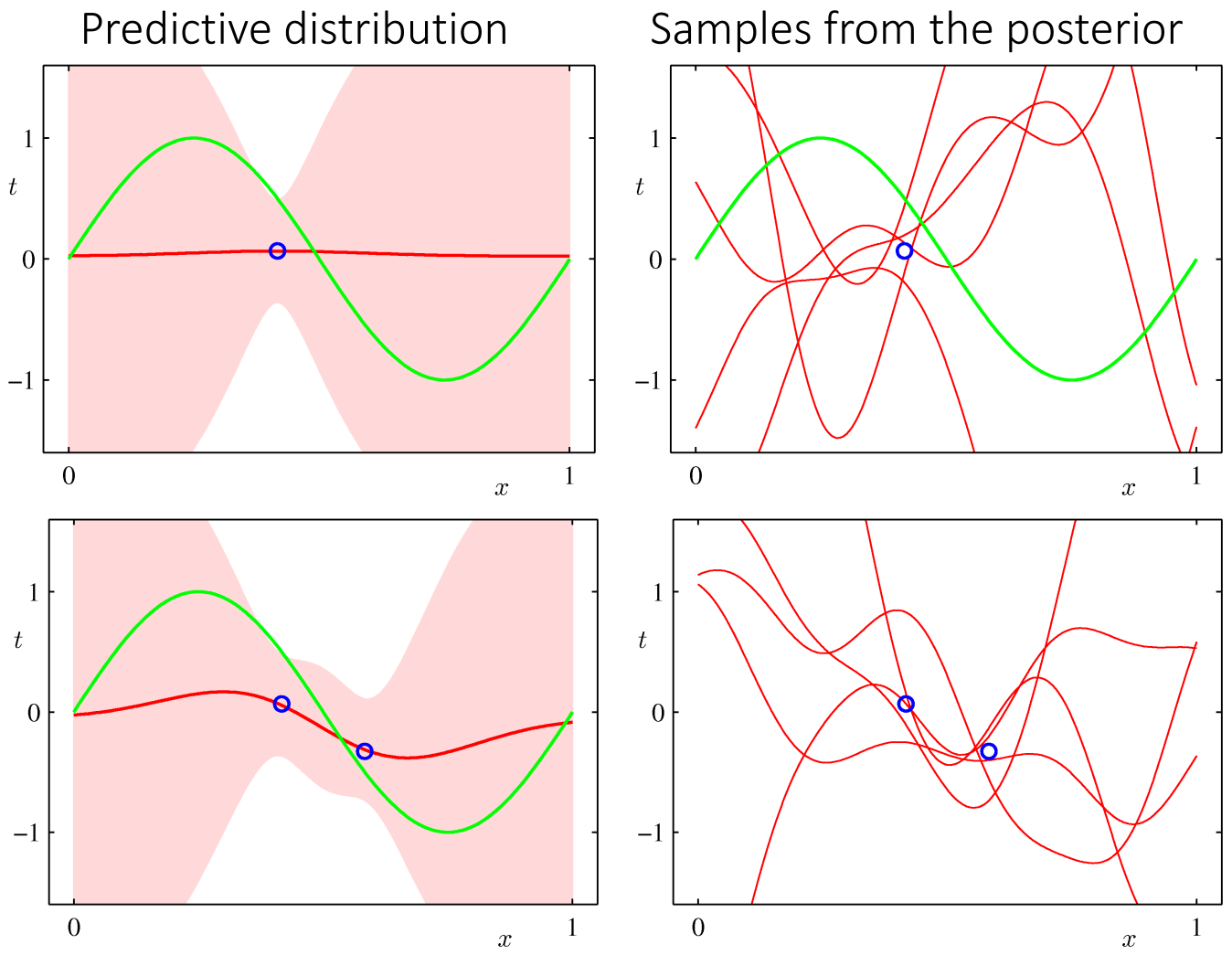
Bayesian predictive distribution



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

Predictive Distribution

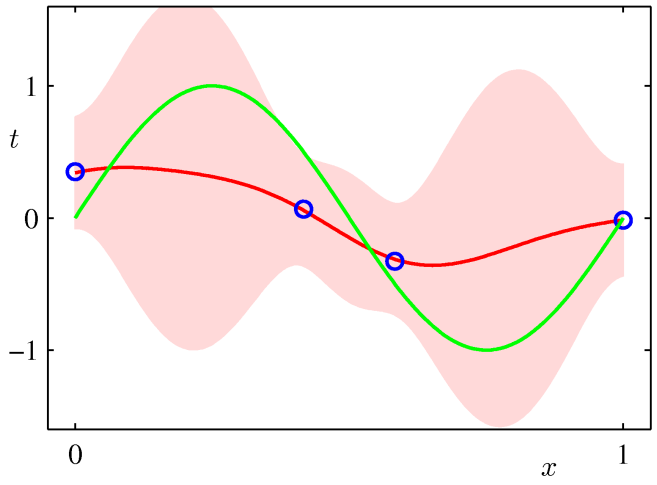
Sinusoidal dataset, 9 Gaussian basis functions.



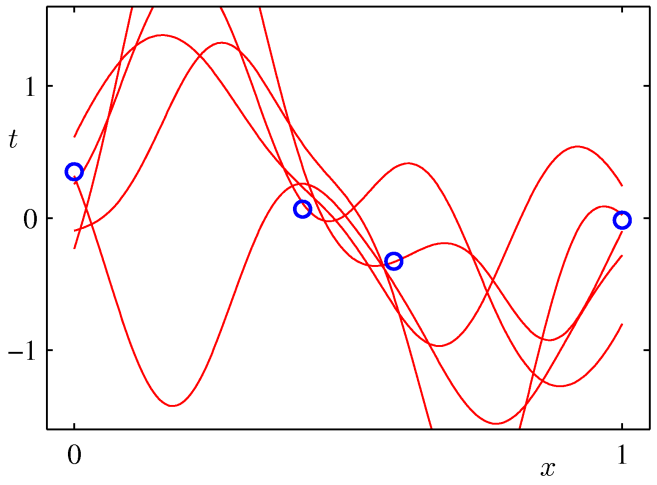
Predictive Distribution

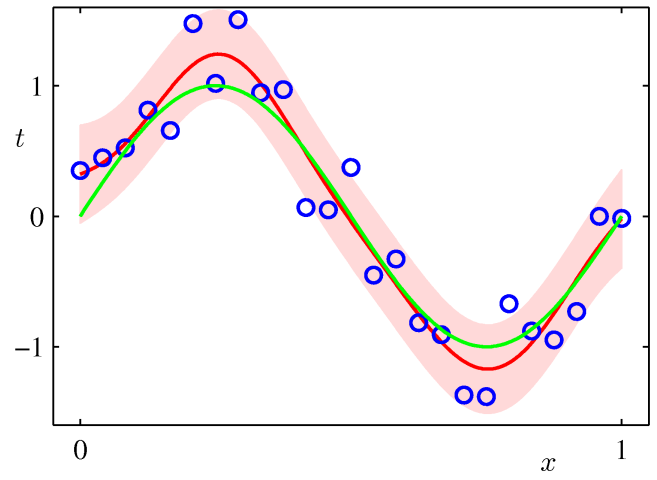
Sinusoidal dataset, 9 Gaussian basis functions.

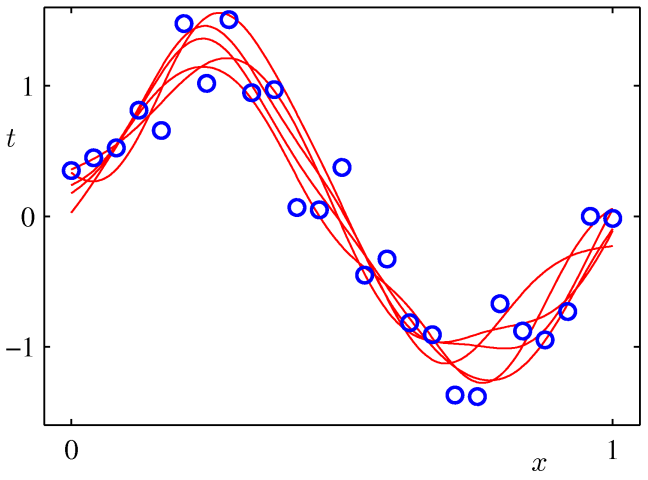
Predictive distribution



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:

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

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

Equivalent kernel.

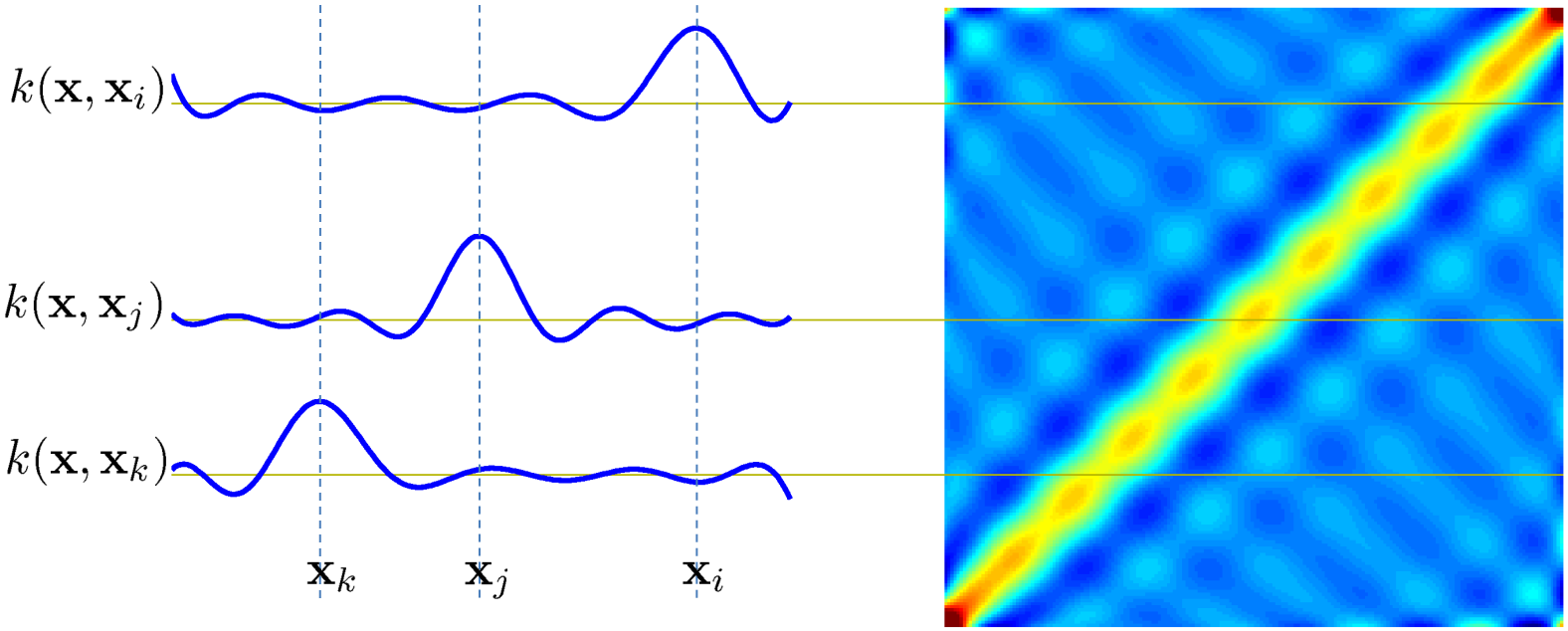
Think of this as inverse distance between x and x_n

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

- A kernel $k(\mathbf{x}, \mathbf{x}') = \beta \phi(\mathbf{x})^T \mathbf{S}_N \mathbf{x}'$ is measures the proximity between two points (inverse distance).

Equivalent Kernel

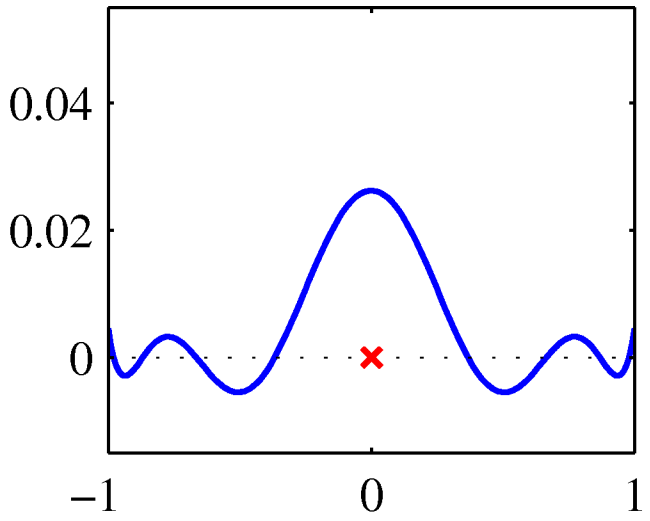
• The weight of t_n depends on distance between x and x_n; nearby x_n carry more weight. This is the equivalent kernel for the previous example w/ Gaussian basis fncs.



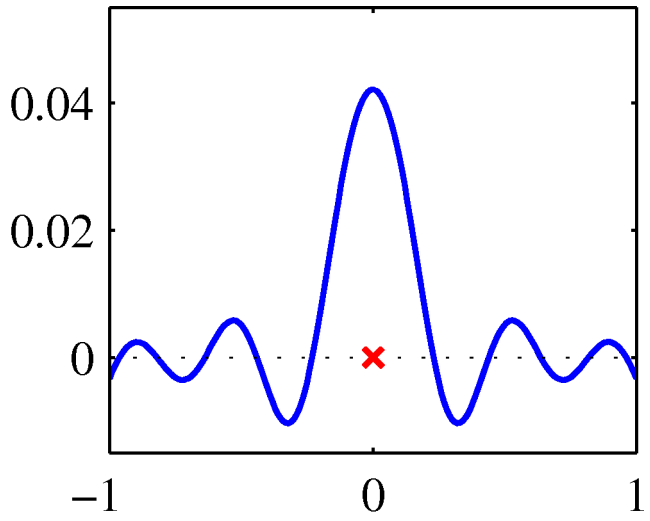
• We can avoid the use of basis functions and define the kernel function directly, leading to Gaussian Processes (soon!).

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'.

Limitations

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

•The data vectors typically lie close to a nonlinear low-dimensional manifold, whose intrinsic dimensionality is smaller than that of the input space. We will learn how to make use of this later.