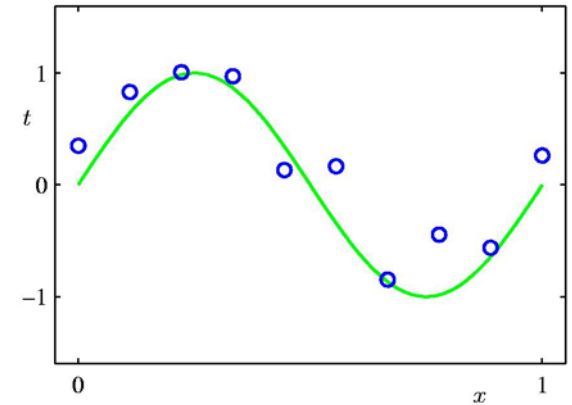


Linear Models for Regression using Basis Functions

Overview

- Supervised learning starting with regression
- Goal: predict value of one or more target variables t given value of D -dimensional vector \mathbf{x} of input variables
- When t is continuous valued we call it *regression*, if t has a value consisting of labels it is called *classification*
- Polynomial is a specific example of curve fitting called linear regression models

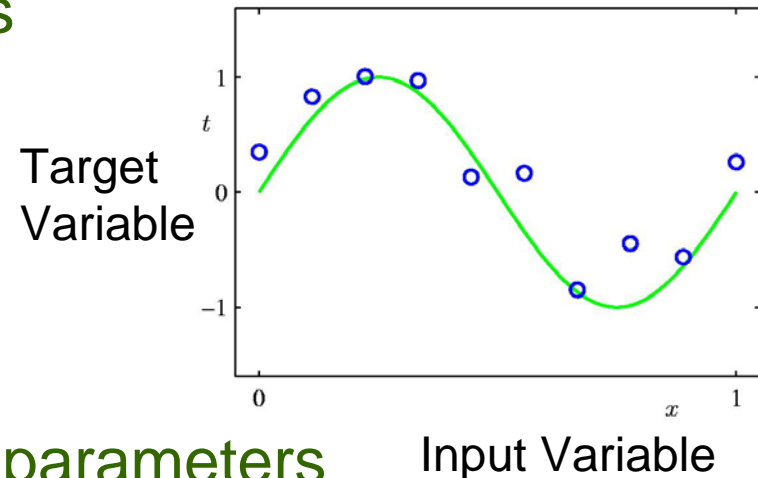


Types of Linear Regression Models

- Simplest form of linear regression models:
 - linear functions of input variables
- More useful class of functions:
 - linear combination of non-linear functions of input variables called *basis functions*
- They are linear functions of parameters (which gives them simple analytical properties), yet are nonlinear with respect to input variables

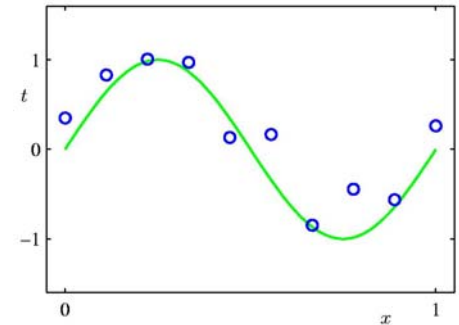
Task at Hand

- Predict value of continuous target variable t given value of D -dimensional input variable \mathbf{x}
 - t can also be a set of variables
- Seen earlier
 - Polynomial
 - Input variable scalar
- This discussion
 - Linear functions of adjustable parameters
 - Specifically linear combinations of nonlinear functions of input variable



Probabilistic Formulation

- Given:
 - Data set of n observations $\{x_n\}$, $n=1, \dots, N$
 - Corresponding target values $\{t_n\}$
- Goal:
 - Predict value of t for a new value of x
- Simplest approach:
 - Construct function $y(x)$ whose values are the predictions
- Probabilistic Approach:
 - Model predictive distribution $p(t|x)$
 - It expresses uncertainty about value of t for each x
 - From this conditional distribution
 - Predict t for any value of x so as to minimize a loss function
 - Typical loss is squared loss for which the solution is the conditional expectation of t



Linear Regression Model

- Simplest linear model for regression with D input variables

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_D x_D$$

where $\mathbf{x} = (x_1, \dots, x_D)^T$ are the input variables

- Called Linear Regression since it is a linear function of
 - parameters w_0, \dots, w_D
 - input variables

Linear Basis Function Models

- Extended by considering nonlinear functions of input variables

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

- where $\phi_j(\mathbf{x})$ are called Basis functions
- There are now M parameters instead of D parameters
- Can be written as

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

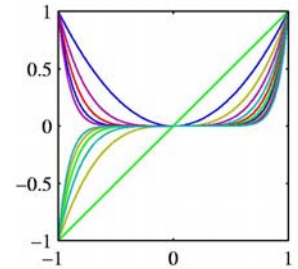
Some Basis Functions

- Linear Basis Function Model

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

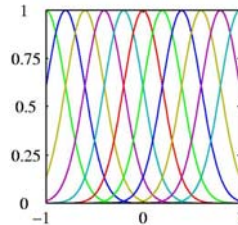
- Polynomial

- In polynomial regression seen earlier, there is a single variable x and $\phi_j(x) = x^j$ with degree M polynomial
- Disadvantage
 - Global



- Gaussian

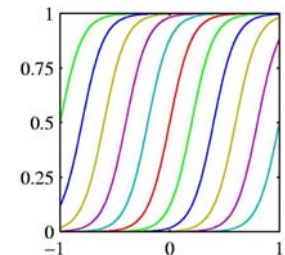
$$\phi_j = \exp\left(-\frac{(x - \mu_j)^2}{2s^2}\right)$$



- Sigmoid

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

Logistic Sigmoid



- Tan h

$$\tanh(a) = 2\sigma(a) - 1$$

Other Basis Functions

- Fourier
 - Expansion in sinusoidal functions
 - Infinite spatial extent
- Signal Processing
 - Functions localized in time and frequency
 - Called wavelets
 - Useful for lattices such as images and time series
- Further discussion independent of choice of basis

Maximum Likelihood Formulation

- Target variable t given by deterministic function $y(\mathbf{x}, \mathbf{w})$ with additive Gaussian noise

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

ε is zero-mean Gaussian with precision β

- Thus distribution of t is normal:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = N(t | \underbrace{y(\mathbf{x}, \mathbf{w})}_{\text{mean}}, \underbrace{\beta^{-1}}_{\text{variance}})$$

Likelihood Function

- Data set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ with values $t = \{t_1, \dots, t_N\}$

- Likelihood

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

- Log-likelihood

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

– Where

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

– Called Sum-of-squares Error Function

- Maximizing Likelihood with Gaussian noise is equivalent to minimizing $E_D(\mathbf{w})$

Maximum Likelihood for weight parameter \mathbf{w}

- Gradient of log-likelihood wrt \mathbf{w}

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

- Setting to zero and solving for \mathbf{w}

$$\mathbf{w}_{ML} = \Phi^+ \mathbf{t}$$

- Where $\Phi^+ = (\Phi^T \Phi)^{-1} \Phi^T$ is the Moore-Penrose pseudo inverse of $N \times M$ Design Matrix

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

Maximum Likelihood for precision β

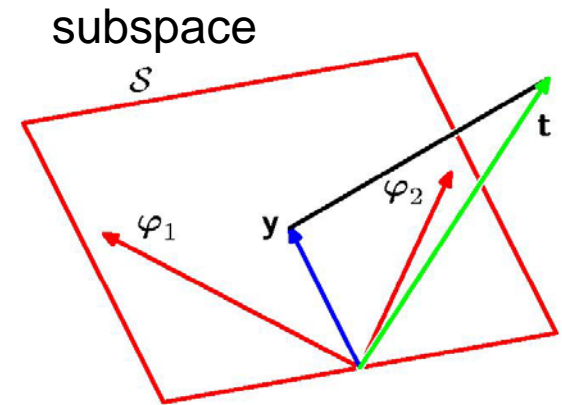
- Similarly gradient wrt β gives

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^N \left\{ t_n - \mathbf{w}_{ML}^T \phi(x_n) \right\}^2$$

← Residual variance of the target values around the regression function

Geometry of Least Squares Solution

- N -dimensional space (target values of N data points)
- Axes given by t_n
- Each basis function $\phi_j(\mathbf{x}_n)$, corresponding to j^{th} column of Φ , is represented in this space
- If $M < N$ then $\phi_j(\mathbf{x}_n)$ are in a subspace \mathbf{S} of dimensionality M
- Solution \mathbf{y} is choice of \mathbf{y} that lies in subspace \mathbf{S} that is closest to \mathbf{t}
 - Corresponds to orthogonal projection of \mathbf{t} onto \mathbf{S}
- When two or more basis functions are collinear, $\Phi^T \Phi$ is singular
 - Singular Value Decomposition is used



← M Basis functions →

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

↑ N Data ↓

Sequential Learning

- Batch techniques for m.l.e. can be computationally expensive for large data sets
- If error function is a sum over n data points $E = \sum_n E_n$ then update parameter vector \mathbf{w} using

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n$$

- For *Sum-of-squares Error Function*

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta (t_n - \mathbf{w}^{(\tau)} \phi(\mathbf{x}_n)) \phi(\mathbf{x}_n)$$

- Known as *Least Mean Squares Algorithm*

Regularized Least Squares

- Adding regularization term to error controls over-fitting

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

- Where λ is the regularization coefficient that controls importance of data-dependent error $E_D(\mathbf{w})$ and the regularization term $E_W(\mathbf{w})$

- Simple form of regularizer

$$E_W(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

- Total error function becomes

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(x_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

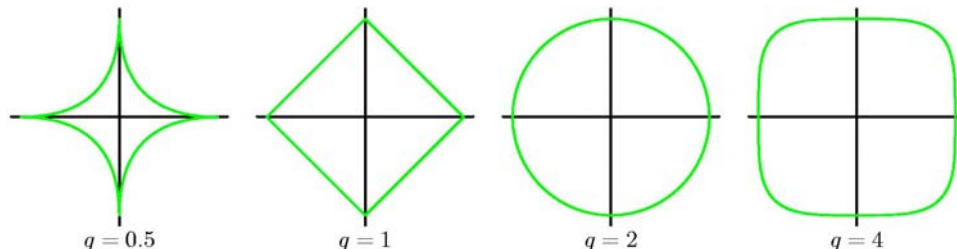
- Called weight decay because in sequential learning weight values decay towards zero unless supported by data

More general regularizer

- Regularized Error

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

- Where $q=2$ corresponds to the quadratic regularizer
- $q=1$ is known as lasso
- Regularization allows complex models to be trained on small data sets without severe overfitting
- Contours of regularization term: $|w_j|^q$

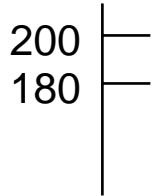


Height of Emperor of China

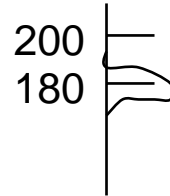
True height is 200 (measured in cm, about 6'6").

Poll a random American: ask “How tall is the emperor?”

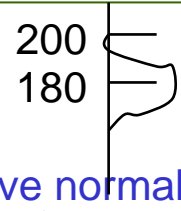
We want to determine how wrong they are, on average



- **Scenario 1**
- Every American believes it is 180
- The answer is always 180
- The error is always -20
- Average squared error is 400
- Average error is 20



- **Scenario 2**
- Americans have normally distributed beliefs with mean 180 and standard deviation 10
- Poll two Americans. One says 190 and other 170
- Bias Errors are -10 and -30
 - Average bias error is -20
- Squared errors: 100 and 900
 - Ave squared error: 500
- $500 = 400 + 100$



- **Scenario 3**
- Americans have normally distributed beliefs with mean 180 and standard deviation 20
- Poll two: One says 200 and other 160
- Errors: 0 and -40
 - Ave error is -20
- Squared errors: 0 and 1600
 - Ave squared error: 800
- $800 = 400 + 400$

Average Squared Error (500) = Square of bias (-20) + variance (100)

Total squared error = square of bias error + variance

Bias and Variance Formulation

- $y(\mathbf{x})$: estimate of the value of t for input \mathbf{x}
- $h(\mathbf{x})$: optimal prediction

$$h(\mathbf{x}) = E[t | \mathbf{x}] = \int tp(t | \mathbf{x})dt$$

- If we assume loss function $L(t, y(\mathbf{x})) = \{y(\mathbf{x}) - t\}^2$

- $E[L]$ can be written as

expected loss = (bias)² + variance + noise

- where

$$(\text{bias})^2 = \int \{E_D[y(\mathbf{x}; D)] - h(\mathbf{x})\}^2 p(\mathbf{x})dx$$

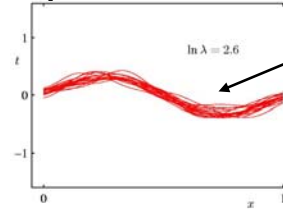
$$\text{variance} = \int E_D[\{\{y(\mathbf{x}; D)] - E_D[y(\mathbf{x}; D)]\}^2] p(\mathbf{x})dx$$

$$\text{noise} = \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)dxdt$$

Dependence of Bias-Variance on Model Complexity

- $h(x) = \sin(2\pi x)$
- Regularization parameter λ
- $L=100$ data sets
- Each has $N=25$ data points
- 24 Gaussian Basis functions
 - No of parameters $M=25$

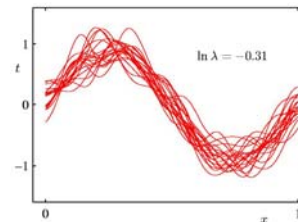
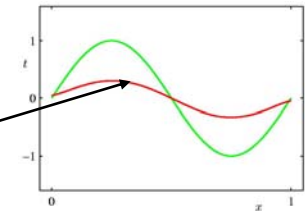
20 Fits for
25 data
points each



High λ

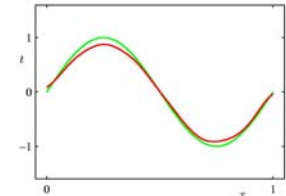
Low
Variance
High bias

Red: Average of Fits
Green: Sinusoid from which
data was generated



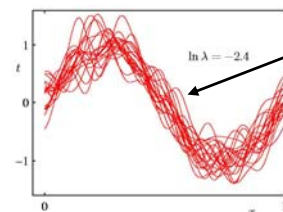
Low λ

High
Variance
Low bias



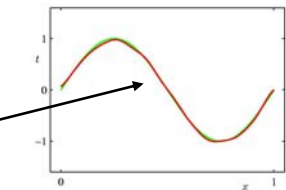
Result of averaging multiple solutions with complex model gives good fit

Weighted averaging of multiple solutions is at heart of Bayesian approach: not wrt multiple data sets but wrt posterior distribution of parameters



Low λ

High
Variance
Low bias



Bayesian Linear Regression

- Prior probability distribution over model parameters \mathbf{w}
- Assume precision β is known
- Since Likelihood function $p(t|\mathbf{w})$ with Gaussian noise has an exponential form
 - Conjugate prior is given by Gaussian $p(\mathbf{w}) = N(\mathbf{w} | \mathbf{m}_0, S_0)$ with mean \mathbf{m}_0 and covariance S_0

Posterior Distribution of Parameters

- Given by product of likelihood function and prior

$$- p(\mathbf{w}|D) = p(D|\mathbf{w})p(\mathbf{w})/p(D)$$

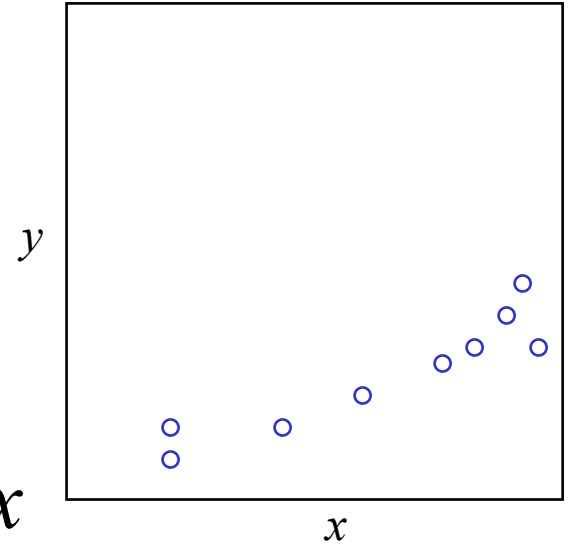
- Due to choice of conjugate Gaussian prior, posterior is also Gaussian
- Posterior can be written directly in the form

$$p(\mathbf{w}|\mathbf{t}) = N(\mathbf{w}|\mathbf{m}_N, S_N) \text{ where}$$

$$\mathbf{m}_N = S_N(S_0^{-1}\mathbf{m}_0 + \beta\Phi^T\mathbf{t}), \quad S_N^{-1} = S_0^{-1} + \beta\Phi^T\Phi$$

Bayesian Linear Regression Example

- Straight line fitting
- Single input variable x
- Single target variable t
- Linear model $y(x, \mathbf{w}) = w_0 + w_1 x$
 - t and y used interchangeably here
- Since there are only two parameters
 - We can plot prior and posterior distributions in parameter space



Sequential Bayesian Learning

Prior/
Posterior
 $p(w)$
gives $p(w|t)$

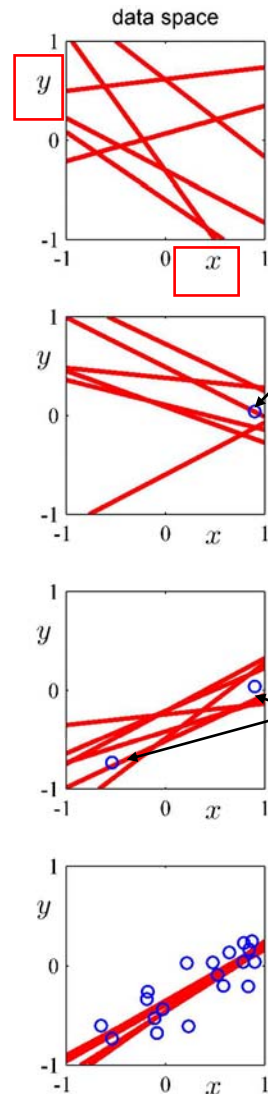
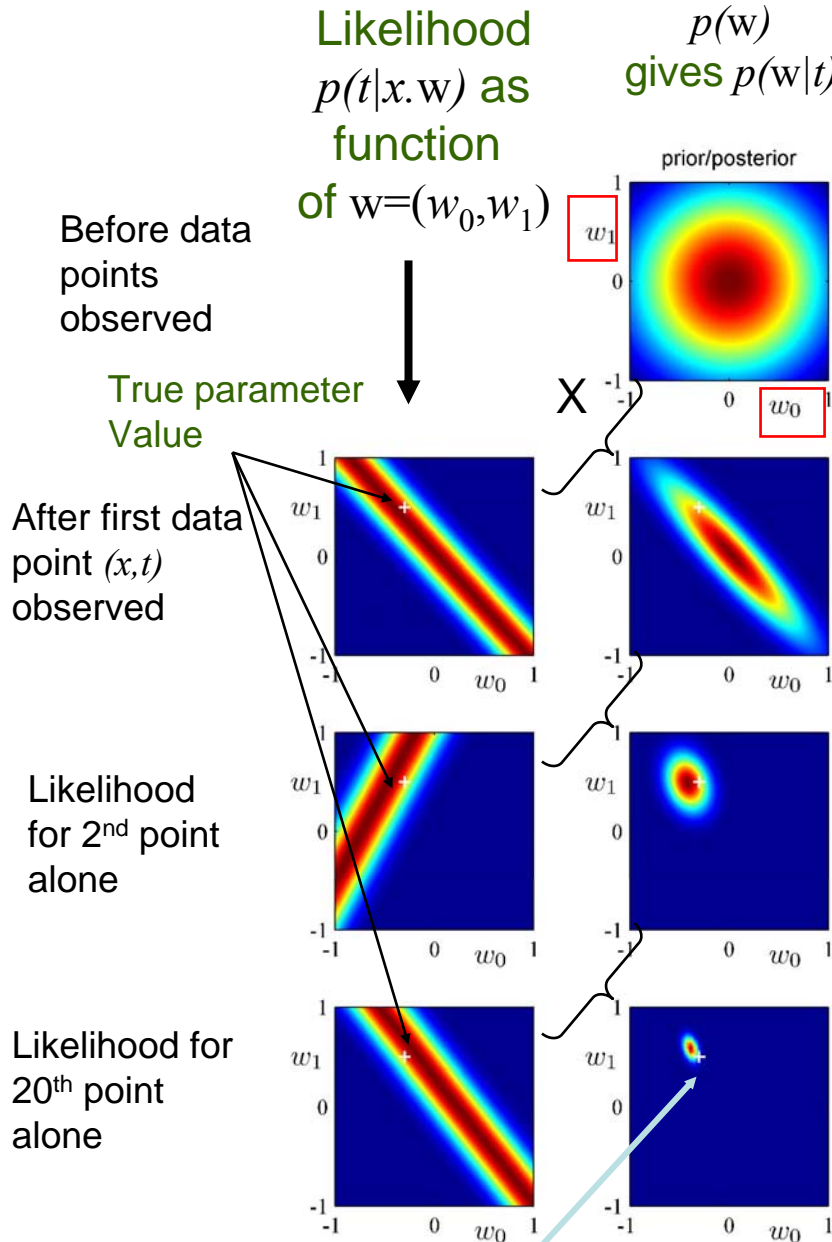
Six samples
(regression functions)
corresponding to $y(x, w)$
with w drawn from
posterior

- Synthetic data generated from $f(x, a) = a_0 + a_1 x$ with parameter values $a_0 = -0.3$ and $a_1 = 0.5$

- By first choosing x_n from $U(x|-1, 1)$, then evaluating $f(x_n, a)$ and then adding Gaussian noise with std dev 0.2 to obtain target values t_n

- Goal is to recover values of a_0 and a_1

With infinite points posterior is a delta function centered at true parameters (white cross)



No
Data
Point

One
Data
Point
 (x, t)

Two
Data
Points

Twenty
Data
Points

Predictive Distribution

- We are usually not interested in the value of w itself
- But predicting t for new values of x
- We evaluate the predictive distribution

$$p(t | x, \alpha, \beta) = N(t | m_N^T \phi(x), \sigma_N^2(x))$$

$$\text{where } \sigma_N^2(x) = \underbrace{\frac{1}{\beta}}_{\text{Noise in data}} + \underbrace{\phi(x)^T S_N \phi(x)}_{\text{Uncertainty associated with parameters } w}$$

Noise in data

Uncertainty associated
with parameters w