CSC411 Fall 2015
Machine Learning

Bayesian Methods

Slides from Rich Zemel

Bayesian approach to machine learning

- Most methods discussed early in course find single best model given the data
- Ensemble methods represent a different approach: develop multiple models (different parameter settings), and combine
- Bayesian alternative: compute distributions over models, make predictions by averaging over model predictions, weighted by their probability

- Can take into account uncertainty in models, also can take advantage of diverse models, which can make independent errors – power of ensemble approach

Bayesian Reasoning

- Quantify all forms of uncertainty using probabilities
- Update probability distributions after observation of new data using Bayes rule

$$D = \{x_i\}$$

$$p(\theta \mid D) \propto \left[\prod_{i=1}^{N} p(x_i \mid \theta)\right] p(\theta)$$

- Assume data sampled from some distribution P(x|θ), where θ is the parameter vector of the distribution

Bayesian Reasoning: Simple Examples

- Bernoulli (coin flip; heads = 1; tails=0; θ=prob of heads):

$$p(\theta \mid D) \propto p(\theta) \prod_{i=1}^{N} (\theta)^{[x_i=1]} (1-\theta)^{[x_i=0]}$$

- Multinomial (die roll; θ_v=prob die value = v):

$$p(\theta \mid D) \propto p(\theta) \prod_{i=1}^{N} \prod_{v=1}^{6} (\theta_v)^{[x_i=v]}$$

Bayesian Reasoning: Interesting Example

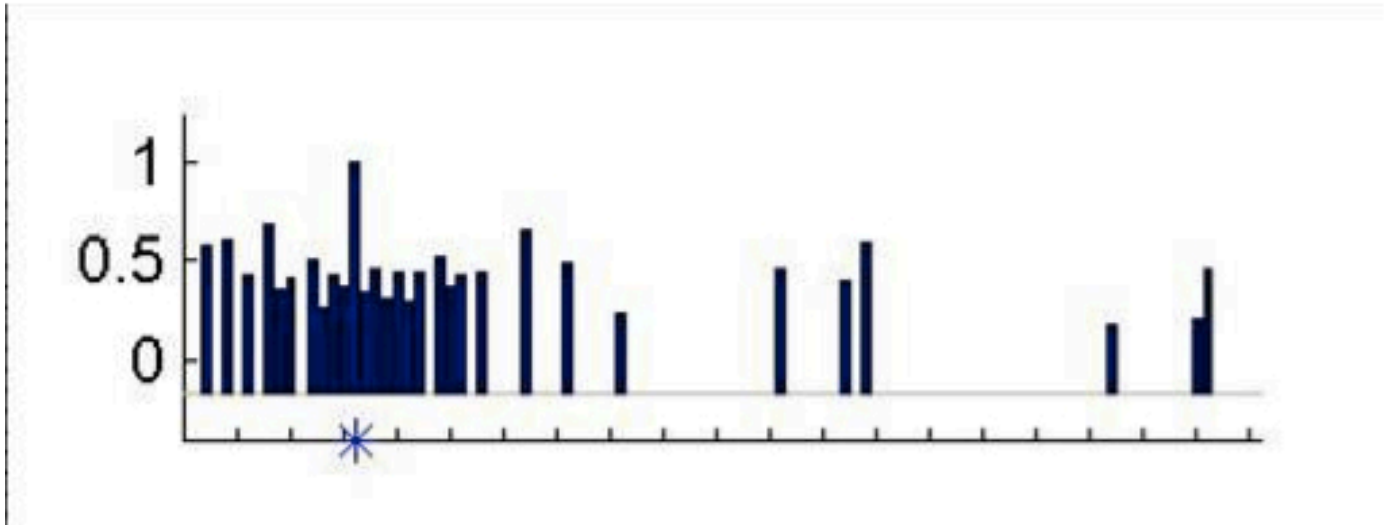
- Cognitive science problem: how do we learn to understand the meaning of a word from only positive examples?
- Parents and others point out positive examples of the concept: “Look at that big dog” or “Don’t lick the dog”
- But not many negative examples: “Check out that non-dog.”
- Can produce negative examples: “That’s not a dog, dummy, that’s a goldfish.”
- But research has shown people can learn from positive examples alone

Number Game

- Learning the meaning of a word = concept learning = binary classification
- Learn indicator function f, which returns a 1 if x is an element in the set C, and 0 otherwise
- Simple example: I tell you I am thinking of an arithmetical concept, such as “prime number” or “number between 15 and 25”
- I give you series of randomly chosen positive examples $D = \{x_1,\ldots x_N\}$ drawn from C; classify x’
- Only integers 1:100; I tell you 16 is a positive example – which other numbers are in C?

Number Game: First Sample

- X are only integers 1:100
- I tell you 16 is a positive example: D= {16}
- Which other numbers are in C?
- 17 is similar because it is nearby; 6 shares a digit; 32 is also even and a power of 2; 99?
- Posterior predictive distribution:



Hypothesis Space

- Imagine hypothesis space of concepts H, such as h_{odd} = {odd numbers}; h_{two} = {powers of two}
- Subset of H consistent with D is version space
- Shrinks as observe more examples

- But how are hypotheses combined to predict class of test example?

- Why is one consistent hypothesis favored over another?

Likelihood

- Given this assumption, prob. of independently sampling N items with replacement from h:

$$p(D \mid h) = \left[\frac{1}{size(h)}\right]^N$$

- Size principle – model favors simplest (smallest) hypothesis consistent with data (Occam's razor)
- Example: D = {16}.
  - $p(D \mid h_{two}) = 1/6$; $p(D \mid h_{even}) = 1/50$
- Likelihoods after four examples: D = {16,8,2,64}.
  - $p(D \mid h_{two}) = 1/6^4$; $p(D \mid h_{even}) = 1/50^4$
  - Likelihood ratio of ~ 5000:1 in favor of h_{two}

Priors

- D = {16,8,2,64}
- Concept h' = "powers of two except 32" may seem more likely than h = "powers of two"
- h' is maximum likelihood estimate
- But seems conceptually unnatural – capture by assigning low prior probability to such concepts
- Different priors: subjective aspect of Bayesian reasoning

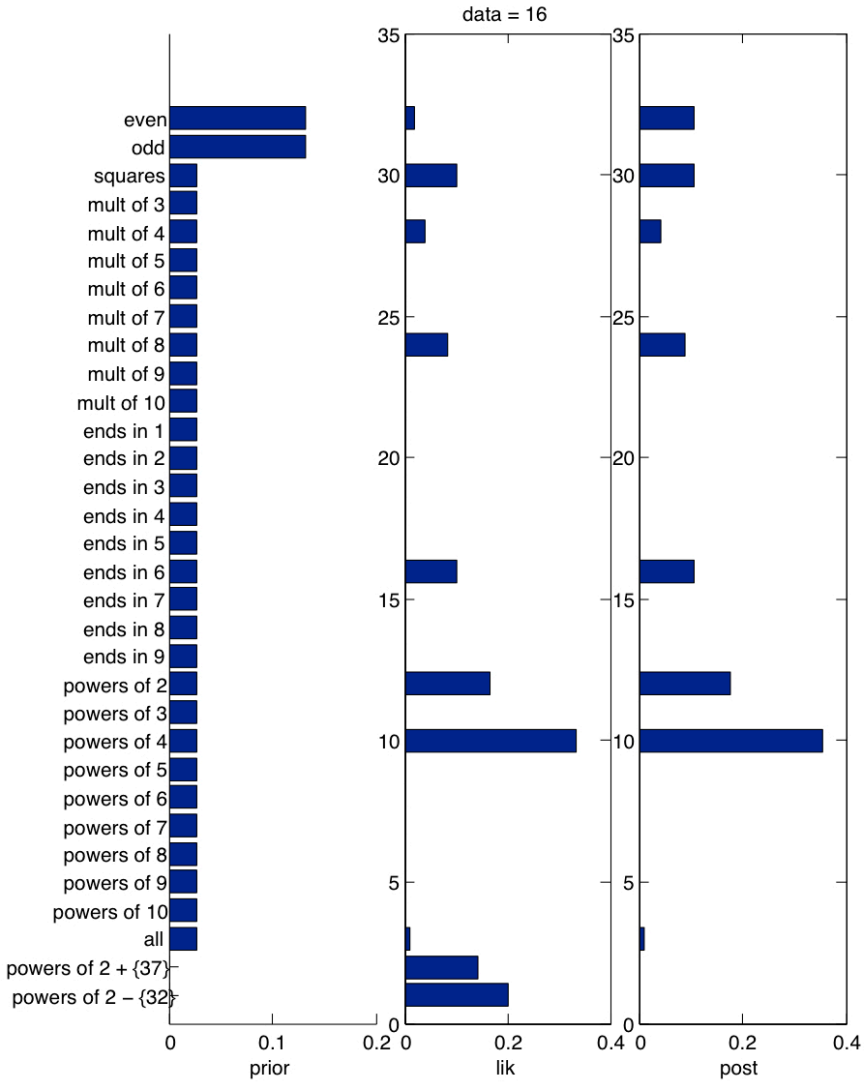
Posterior

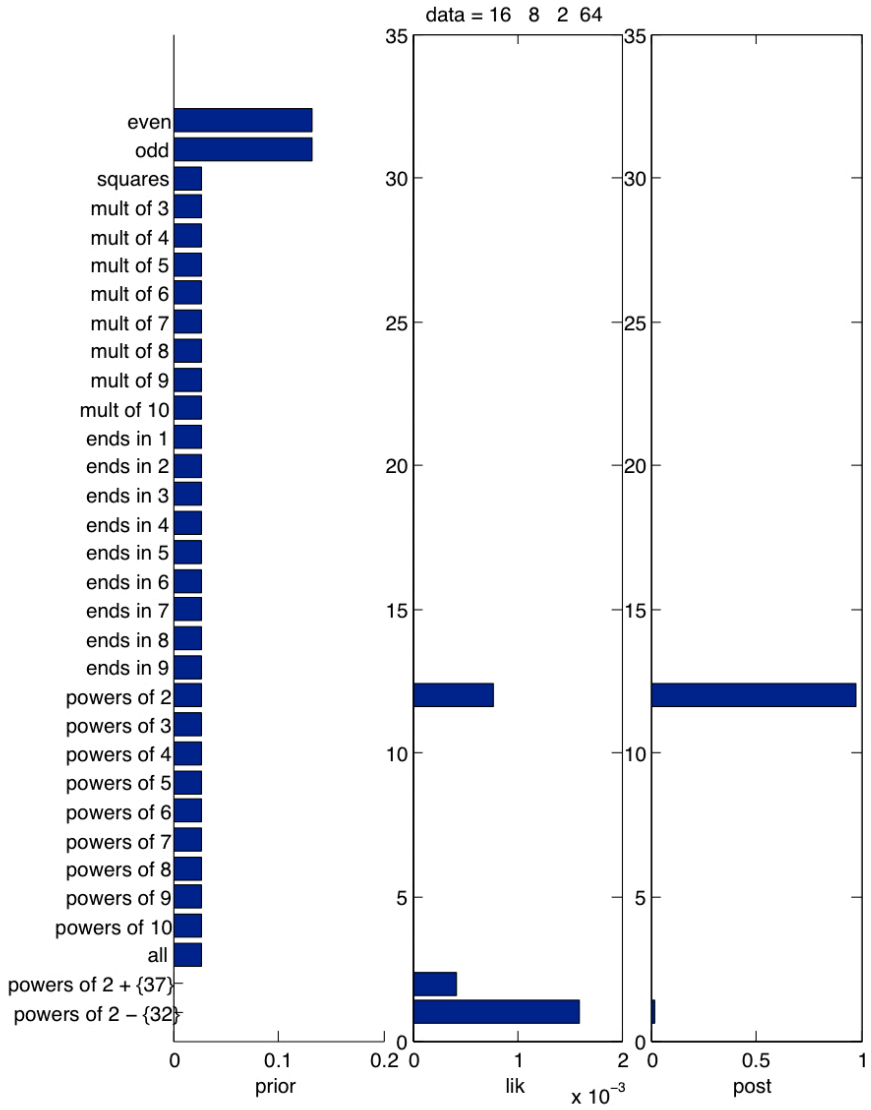
- D1 ={16}; D2 = {16,8,2,64}

$$p(h \mid D) = \frac{p(D \mid h)\,p(h)}{\sum_{h' \in H} p(D \mid h')\,p(h')} = \frac{p(h)[D \in h] / |h|^N}{\sum_{h' \in H} p(h')[D \in h'] / |h'|^N}$$

- Illustrate with simple prior, supports 30 arithmetical concepts, and two unnatural ones, with lower priors

Posteriors/Bayesian Updating





Bayesian Recipe

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

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

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.

Predictive distribution

For new data example y_{new}:

$$P(y_{new} | x_{new}, \boldsymbol{D}) = \int P(y_{new} | x_{new}, \theta)\, P(\theta | \boldsymbol{D})\, d\theta$$

Utility of this posterior distribution

1. Select most likely prediction: arg $\max_y P(y_{new} | \boldsymbol{D})$
2. Compute confidence in prediction: variance of $P(y_{new} | \boldsymbol{D})$
3. Make decisions so as to minimize posterior expected loss

Evaluating the posterior often difficult (intractable integral, exponentially large summations)

→ often rely on numerical approximations (e.g., Monte Carlo sampling)

Representing distributions using samples

For many prior and posterior distributions, often hard to represent or understand using formulas

An alternative, general technique is to represent the distribution using a sample of many values drawn randomly from it

- Can then use samples or their projections to visualize the distribution
- Can make Monte Carlo estimate for probabilities or expectations wrt distribution by taking averages over these sample values

Sampling is a very popular approach to Bayesian learning

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.

Example: Bayesian Regression

Observe pairs ($\mathbf{x}^n$,y^n) for n = 1:N; $\mathbf{x}$ = (x_1,x_2,…); y is real-valued output. Want to predict y given $\mathbf{x}$.

$$y = \mathbf{x}^T \mathbf{w} + n, \qquad n \sim N(0, \sigma^2)$$

Need prior distribution over weights *p(***w***)*

$$\mathbf{w} \sim N(0, \alpha^{-1}\mathbf{I})$$

Data likelihood

$$p(y_{1:N} \mid \mathbf{x}_{1:N}, \mathbf{w}) = \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i, \mathbf{w})$$

Posterior

$$p(\mathbf{w} \mid \mathbf{x}_{1:N}, y_{1:N}) = \frac{\prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i, \mathbf{w}) p(\mathbf{w})}{p(y_{1:N} \mid \mathbf{x}_{1:N})}$$

Bayesian Regression Posterior

$$
\begin{aligned}
&-\log p(\mathbf{w} \mid \mathbf{x}_{1:N}, y_{1:N}) \\
&\quad = -\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \mathbf{w}) - \log p(w) + \log p(y_{1:N} \mid \mathbf{x}_{1:N}) \\
&\quad = \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \mathbf{x}_i^T \mathbf{w})^2 + \frac{\alpha}{2} \|\mathbf{w}\|^2 + const. \\
&\quad = \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{x}^T \mathbf{w})^T (\mathbf{y} - \mathbf{x}^T \mathbf{w}) + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + const. \\
&\quad = \frac{1}{2\sigma^2} (\mathbf{w} - \overline{\mathbf{w}})^T \mathbf{K}^{-1} (\mathbf{w} - \overline{\mathbf{w}}) + const.
\end{aligned}
$$

$$
\mathbf{y} = [y_1, \ldots, y_N]^T \qquad \mathbf{K} = (\mathbf{x}^T \mathbf{x} / \sigma^2 + \alpha \mathbf{I})^{-1} \qquad \overline{\mathbf{w}} = \mathbf{K} \mathbf{x}^T \mathbf{y} / \sigma^2
$$

Bayesian Regression: Posterior

Derivation shows us that posterior distribution over parameters is a multidimensional Gaussian

$$p(\mathbf{w} \mid \mathbf{x}_{1:N}, y_{1:N}) = N(\mathbf{w}; \overline{\mathbf{w}}, \mathbf{K})$$

Most likely (MAP) estimate of model is the mean.

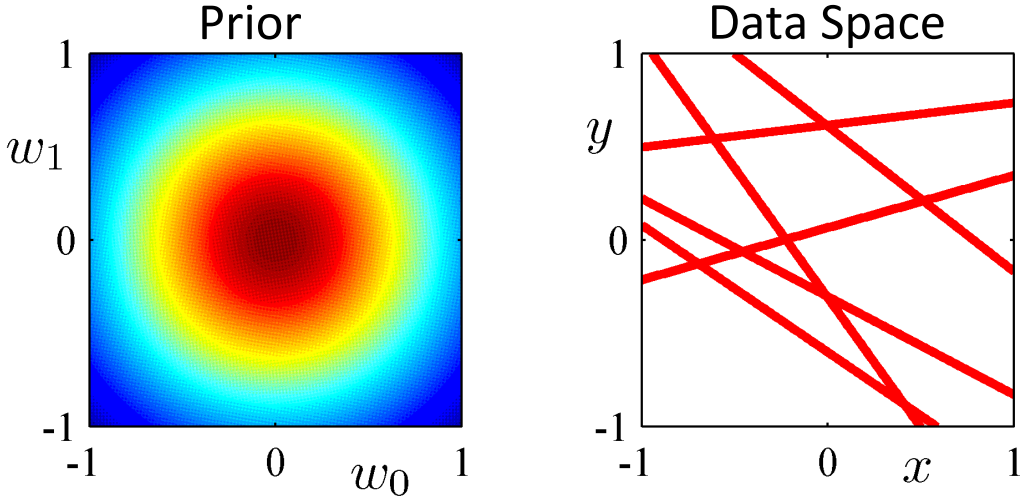
Covariance describes uncertainty in these parameters

Posterior distribution is Gaussian -- can visualize distributions over parameters via sampling

Bayesian Linear Regression

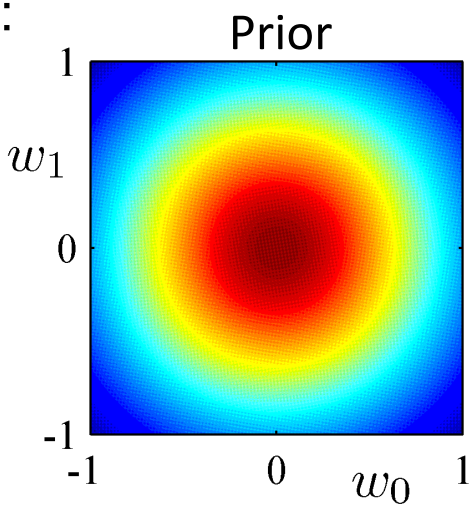
- Consider a linear model of the form: $y(x, \mathbf{w}) = w_0 + w_1 x.$
- The training data is generated from the function $f(x, \mathbf{a}) = a_0 + a_1 x$ with $a_0 = 0.3; a_1 = 0.5,$ by first choosing x_n uniformly from [-1;1], evaluating $f(x, \mathbf{a})$, and adding a small Gaussian noise.
- Goal: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

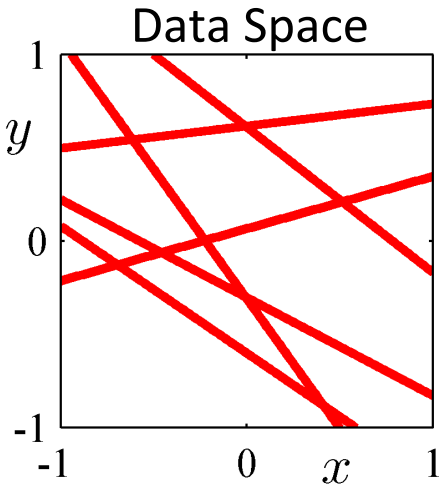
0 data points are observed:



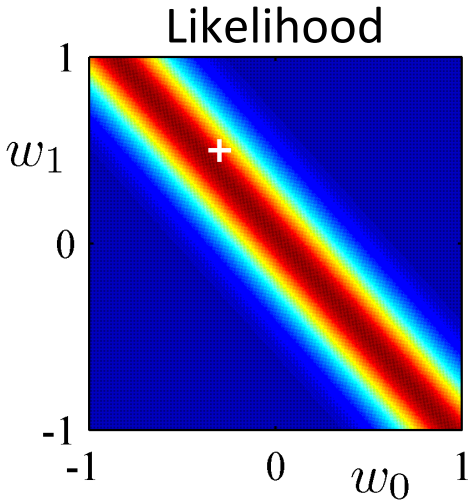
Bayesian Linear Regression

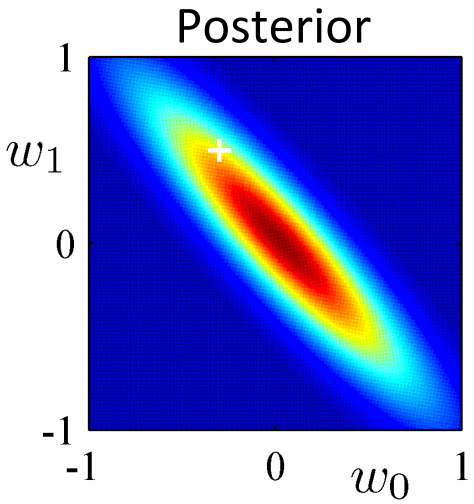
0 data points are observed:

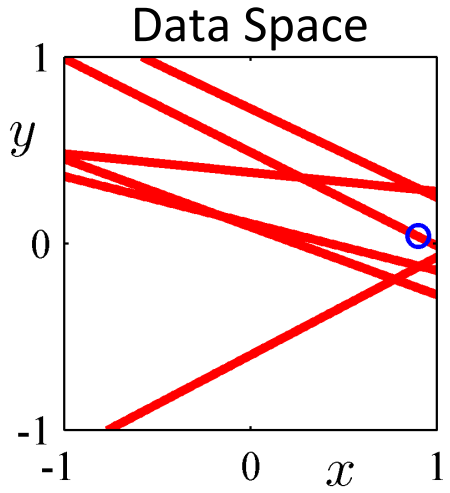




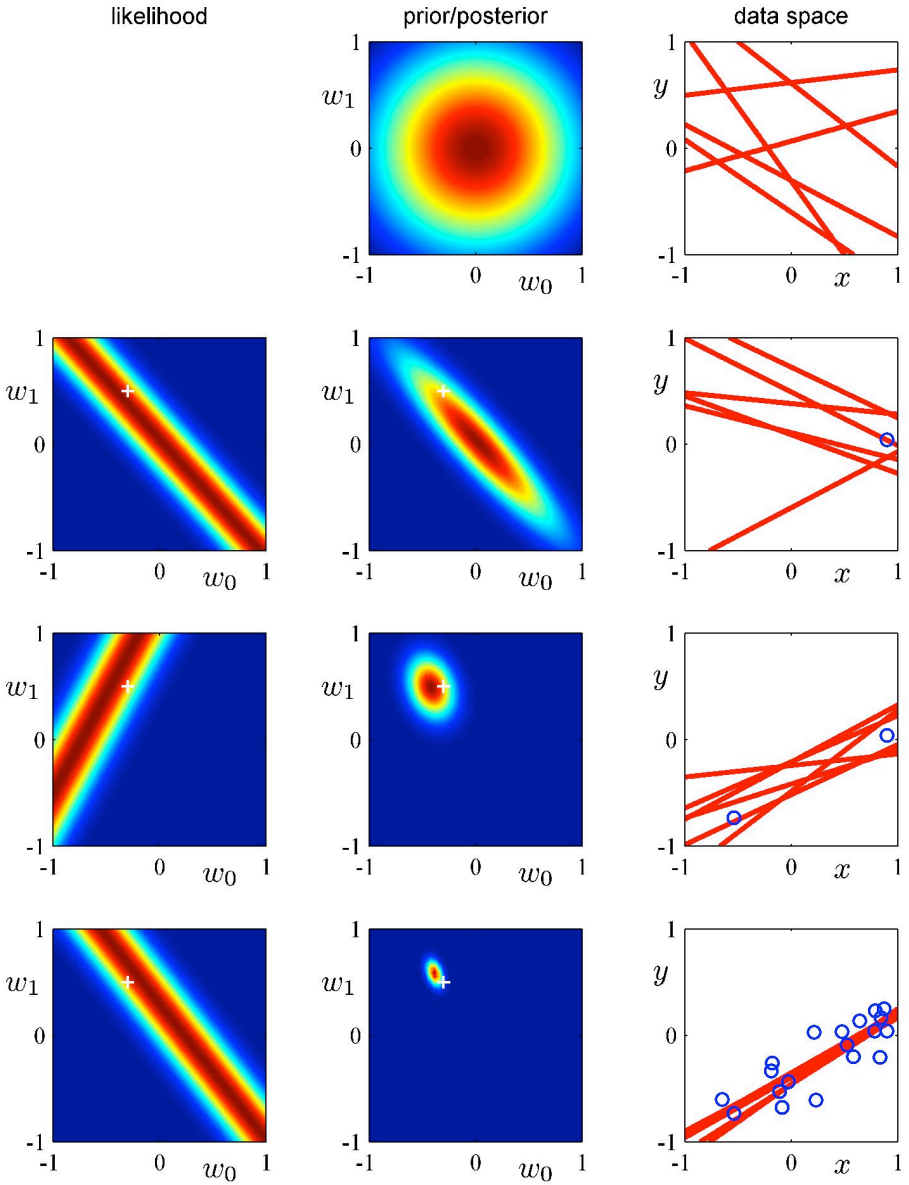
1 data point is observed:







Bayesian Linear Regression



0 data points are observed.

1 data point is observed.

2 data points are observed.

20 data points are observed.

Bayesian Regression: Prediction

Predictive distribution for new point:

$$p(y_{new} \mid \mathbf{x}_{new}, D) = \int p(y_{new} \mid \mathbf{x}_{new}, D, \mathbf{w})\, p(\mathbf{w} \mid D)\, d\mathbf{w}$$

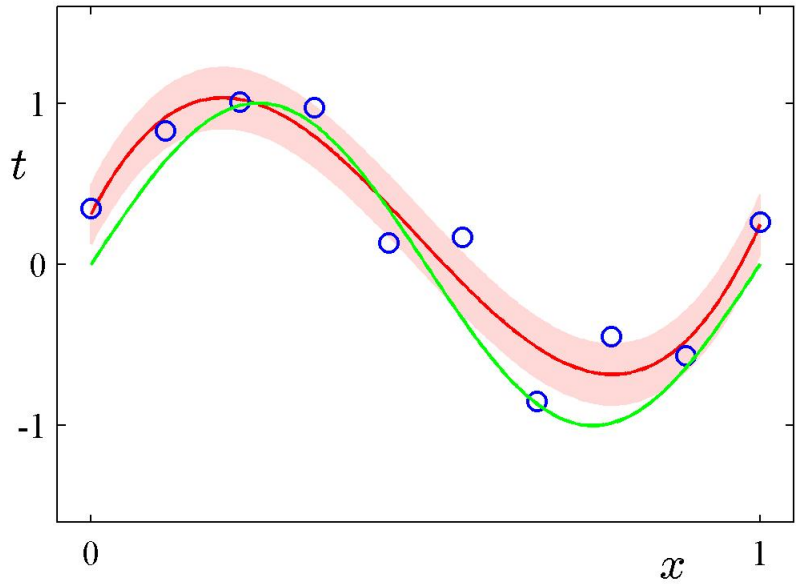
$$= N(y_{new}; \mathbf{x}_{new}{}^T \overline{\mathbf{w}}, \sigma^2 + \mathbf{x}_{new}{}^T \mathbf{K} \mathbf{x}_{new})$$

Simple case – everything Gaussian -- can visualize distributions over parameters via sampling

$$\mathbf{y} = [y_1, \ldots, y_N]^T \quad \mathbf{K} = (\mathbf{x}^T \mathbf{x} / \sigma^2 + \alpha \mathbf{I})^{-1} \quad \overline{\mathbf{w}} = \mathbf{K} \mathbf{x}^T \mathbf{y} / \sigma^2$$

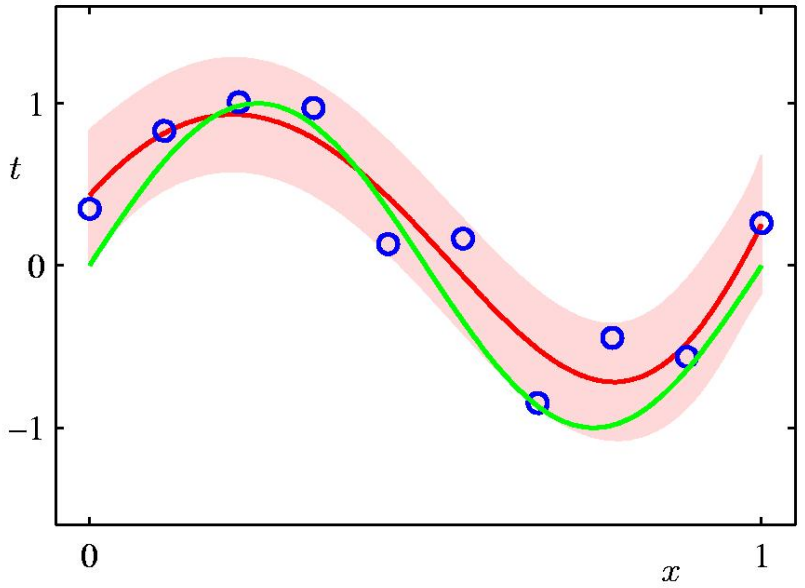
Predictive Distribution: Bayes vs. ML

Predictive distribution based on maximum likelihood estimates



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

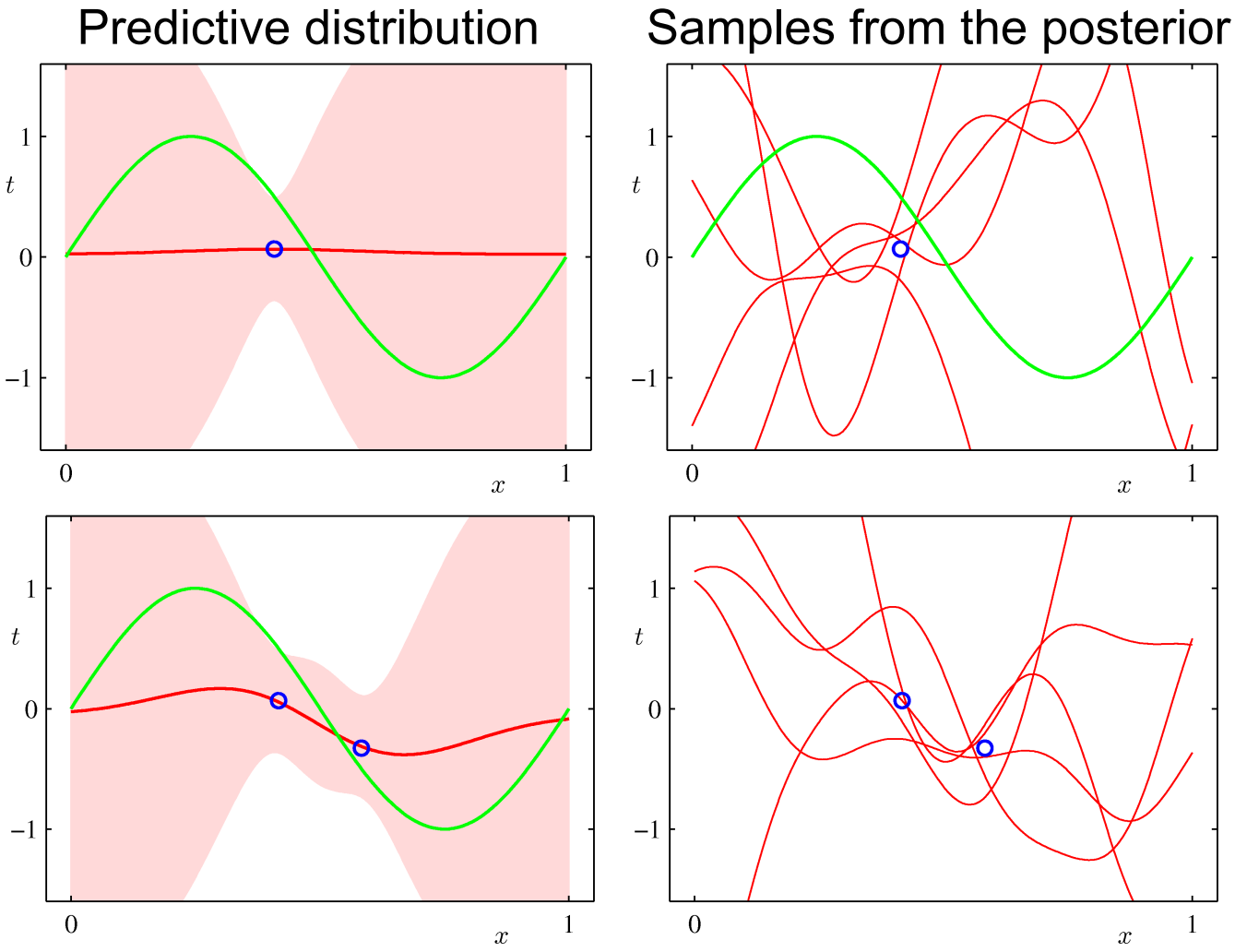
Bayesian predictive distribution



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

Predictive Distribution

Sinusoidal dataset, 9 Gaussian basis functions.



Predictive Distribution

Sinusoidal dataset, 9 Gaussian basis functions.