Introduction to Probabilistic Modeling and Inference

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

Why a Probabilistic Approach?

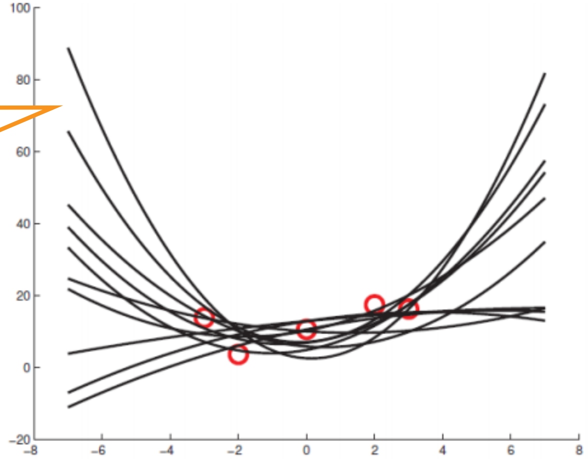
- In machine learning or learning from data in general, we usually want to
  - Learn a model for the data (model usually is defined by some parameters $\boldsymbol{\theta}$)
  - Use the learned model to make predictions
- How (un)certain we are about the model/parameters we have learned?
  - Crucial if we have limited data to learn from
- How (un)certain we are about the predictions made by the model?
  - Crucial if our model/parameters are uncertain
- How (un)certain we are about the data itself?
  - Important if the process that generated data is noisy/uncertain/unknown
- Also, many problems require us to make probabilistic/soft predictions, e.g.,
  - Predict the probability that a transaction is fraud, or that a person has cancer
- A probabilistic approach can naturally handle all of the above (and more)

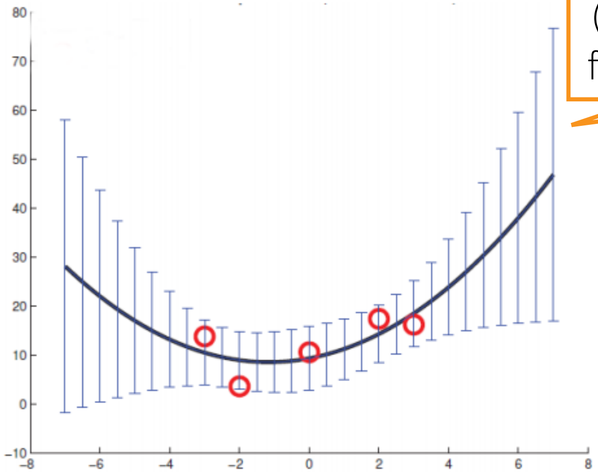
Why a Probabilistic Approach (Contd)?

- Uncertainty about parameter estimates
  - Don't report a single best parameter but a prob. distr. $p(\theta|D)$ over params given data

Each of these curves is generated by sampling from the learned probability distribution $p(\theta|D)$ of the parameters θ given data D

At the time of making predictions for test inputs, each of these curves will be used to predict the output and we will take a weighted average (will see later how the weighting is done)





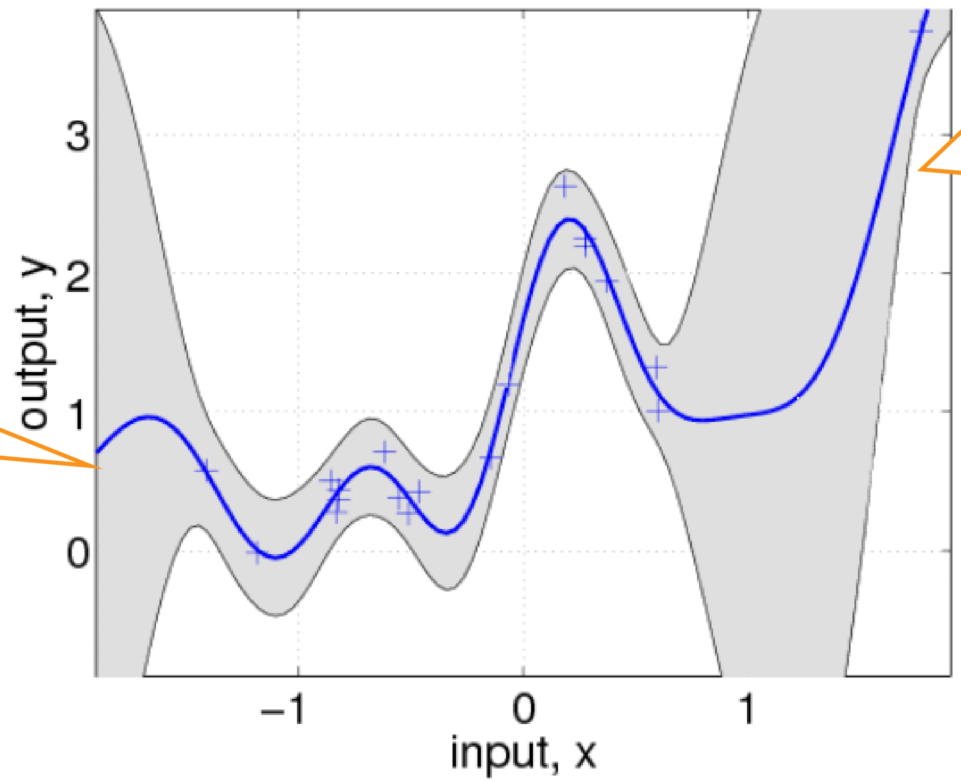
Predictions with error bars (mean with std deviation for each prediction

- Uncertainty about predictions
  - Output a probability distribution or report uncertainty estimate for the predictions

Why a Probabilistic Approach (Contd)?

- Sequential decision-making: Information about uncertainty can "guide" us, e.g.,

Blue curve is the mean of the function, shaded region denotes the predictive uncertainty



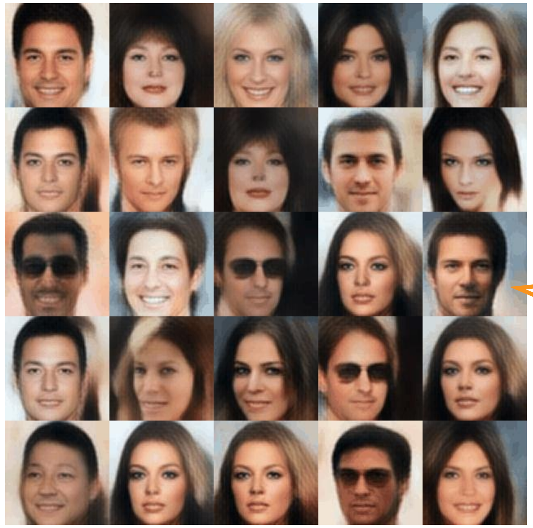
Given our current estimate of the regression function, which training input(s) should we add next to improve its estimate the most?

Uncertainty can help here: Acquire training inputs from regions where the function is most uncertain about its current predictions

- Applications in active learning, reinforcement learning, Bayesian optimization, etc

Why a Probabilistic Approach (Contd)?

- Often wish to learn the underlying probability distribution $p(\boldsymbol{x})$ of the data
- Useful for many tasks, e.g.,
  - Outlier/novelty detection: Outliers will have low probability under $p(\boldsymbol{x})$
  - Can sample from this distribution to generate new "artificial" but realistic-looking data



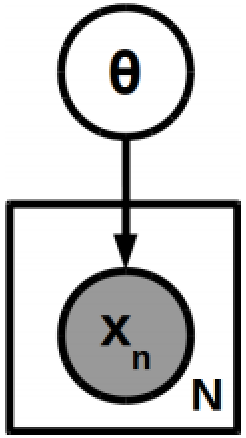
Several models, such as generative adversarial networks (GAN), variational auto-encoders (VAE), etc can do this

Pic credit: https://medium.com/analytics-vidhya/an-introduction-to-generative-deep-learning-792e93d1c6d4

Modeling Data Probabilistically: A Simplistic View

- Assume data $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ generated from a prob. model with params $\boldsymbol{\theta}$

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N \sim p(\boldsymbol{x}|\theta)$$

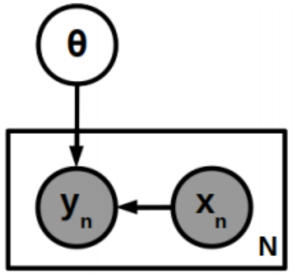


A plate diagram of this simplistic model

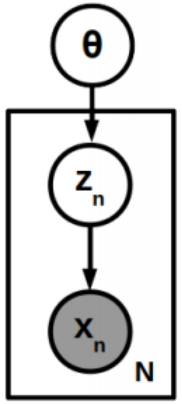
- Note: Shaded nodes = observed; unshaded nodes = unknown/unobserved
- Goal: To estimate the unknowns ($\boldsymbol{\theta}$ in this case), given the observed data $\mathbf{X}$
  - Many ways to do this (point estimate or the posterior distribution of $\boldsymbol{\theta}$)
- Can use the parameter estimates to make predictions, e.g.,
  - Probability density of a new input $\boldsymbol{x}_*$ under this model

Modeling Data Probabilistically

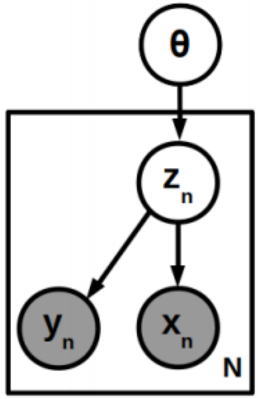
- This previous problem set-up can be generalized in various ways



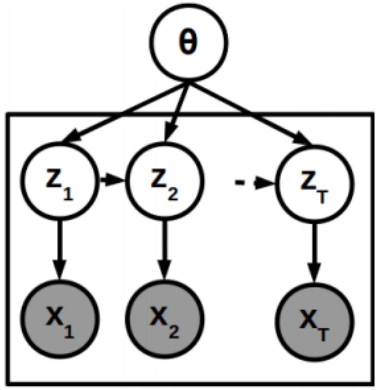
A simple supervised learning model



A latent variable model for unsupervised learning



A latent variable model for supervised learning

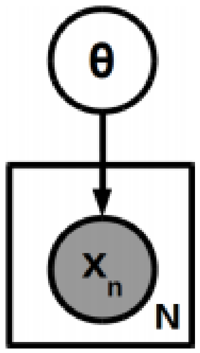


A latent variable model for sequential data

- Any node (even if observed) we are uncertain about is modeled by a prob. distribution
  - These nodes become the random variables of the model
- The full model is specified via a joint prob. distribution over all random variables
- The goal is to infer the distribution of unknowns of the model, given the observed data

Modeling Data Probabilistically

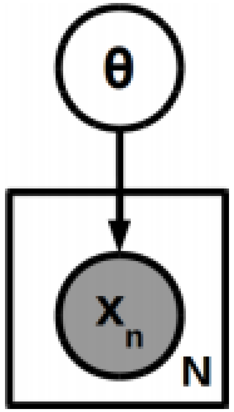
- Specification of prob. models requires two key ingredients: Likelihood and prior



- Likelihood $p(\boldsymbol{x}|\theta)$ or the "observation model" specifies how data is generated
  - Measures data fit (or "loss") w.r.t. the given parameter θ
- Prior distribution $p(\theta)$ specifies how likely different parameter values are *a priori*
  - Also corresponds to imposing a "regularizer" over θ
- Domain knowledge can help in the specification of the likelihood and the prior
  - A key benefit of probabilistic modeling

Estimation/Inference in Probabilistic Models

- A simple way: Find $\boldsymbol{\theta}$ for which the observed data is most likely or most probable



$$\hat{\theta} = \arg\max_{\theta} \log p(\mathbf{X}|\theta)$$

This "point estimate", however, does not provide us any information about uncertainty in $\boldsymbol{\theta}$

- More desirable: Estimate the full posterior distribution over $\boldsymbol{\theta}$ to get the uncertainty

Fully Bayesian inference. In general, an intractable problem, except for some simple cases (will study how to solve such problems)

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

- When making predictions, can use the full posterior rather than a single best $\boldsymbol{\theta}$
  - This is typically referred to as posterior averaging

Posterior Averaging

$$p(\boldsymbol{x}_*|\mathbf{X}) = \int p(x_*, \theta|\mathbf{X})\, d\theta$$
$$= \int p(x_*|\theta, \mathbf{X}) p(\theta|\mathbf{X})\, d\theta$$
$$= \int p(x_*|\theta) p(\theta|\mathbf{X})\, d\theta$$

Assuming observations are i.i.d. given θ

- Can use the posterior over parameters to compute "averaged prediction", e.g.,

$$p(\boldsymbol{x}_*|\mathbf{X}) = \int p(\boldsymbol{x}_*|\theta) p(\theta|\mathbf{X}) d\theta$$

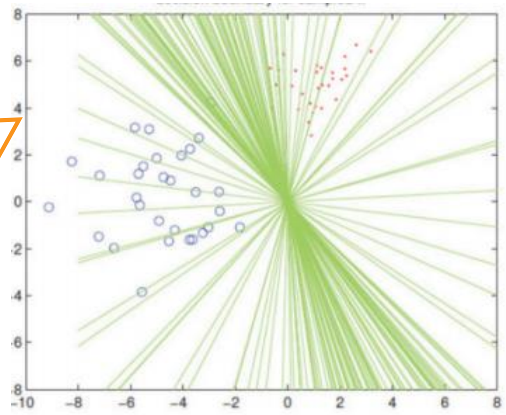
Posterior predictive distribution (obtained by doing an importance-weighted averaging over the posterior)

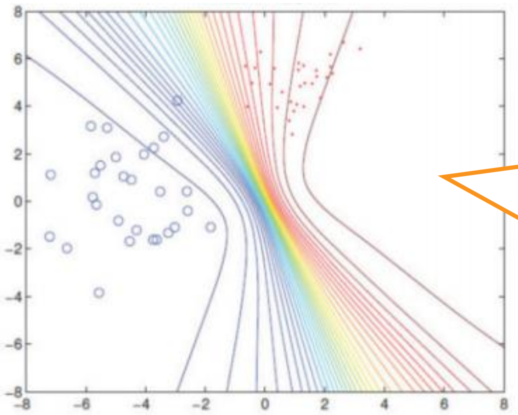
Plug-in predictive distribution

Tells us how important this value of $\boldsymbol{\theta}$ is

- Posterior averaging yields more robust predictions since we aren't trusting a single "optimal" value of $\boldsymbol{\theta}$ (can also think of it as giving an ensemble of models)

Samples of linear separators drawn from the posterior of a probabilistic binary classification model (each line will have a different importance in computing the posterior predictive)

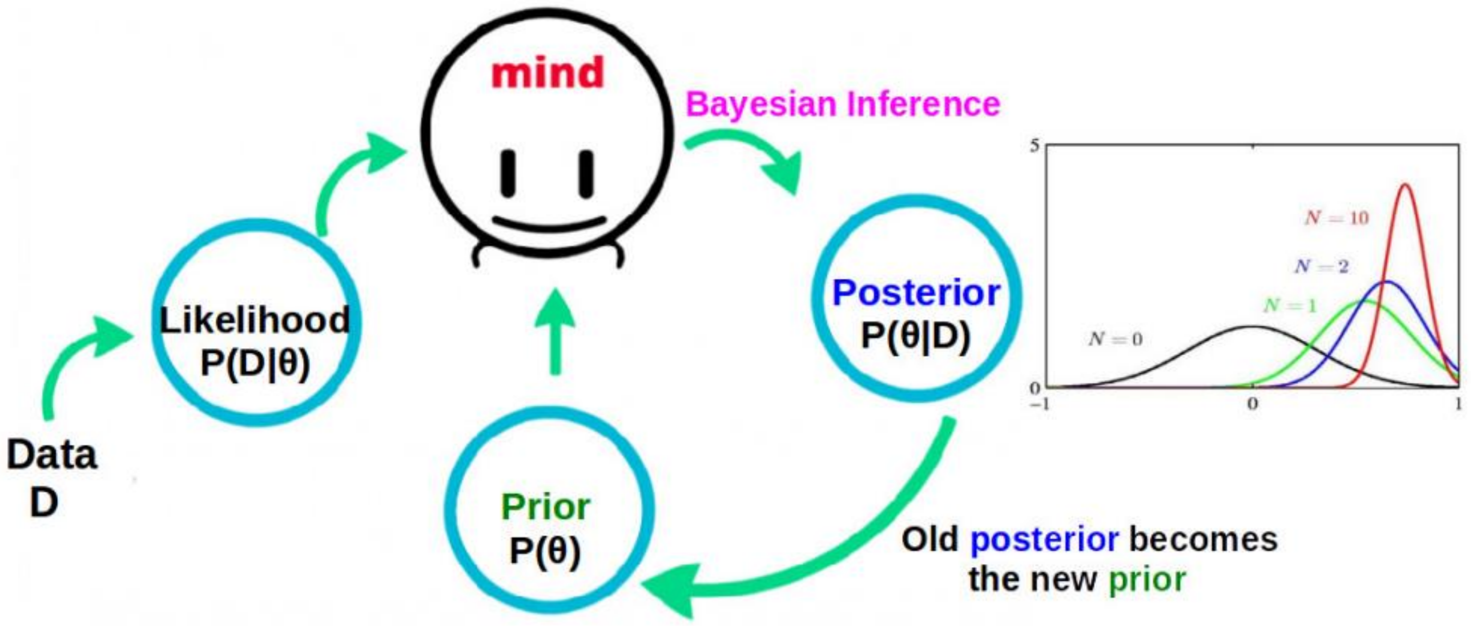




Effect of posterior averaging (each curve is an equal-probability contour, and is not a straight line!)

Bayesian Inference

- Bayesian inference can be seen in a sequential fashion



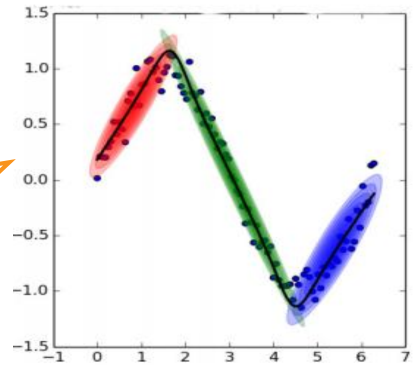
- Our belief about θ keeps getting updated as we see more and more data
  - Posterior keeps getting updates as more and more data is observed
  - Note: Updates may not be straightforward and approximations may be needed

Some Other Benefits

Modular Construction of Complex Models

- Can combine multiple simple probabilistic models to learn complex patterns

A combination of a mixture model for clustering and a probabilistic linear regression model: Result is a probabilistic nonlinear regression model



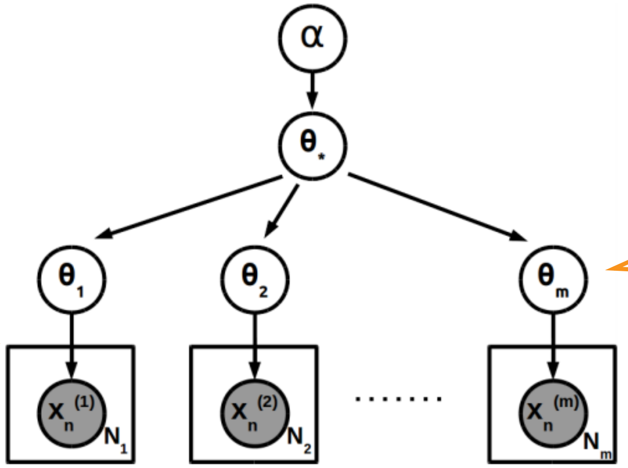
Can design a latent variable model to do this

Essentially a "mixture of experts" model

- Can design models that can jointly learn from multiple datasets and share information across multiple datasets using shared parameters with a prior distribution

Example: Estimating the means of m datasets, assuming the means are somewhat related. Can do this jointly rather than estimating independently

Easy to do it using a probabilistic approach with shared parameters (will see details later)

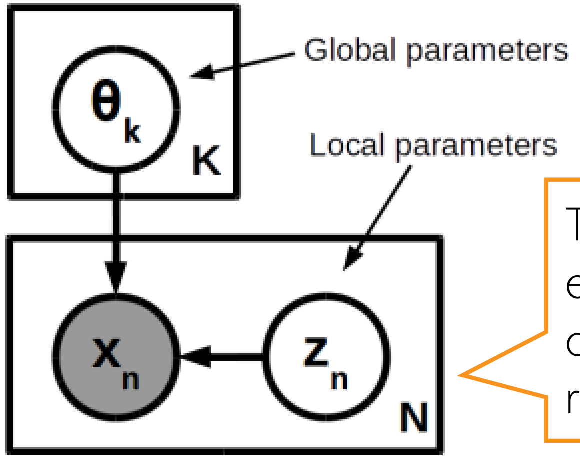


An example of transfer learning or multitask learning using a probabilistic approach

Generative Latent Variable Models

- Generative models of data can be naturally specified in a probabilistic framework

Each data point $\boldsymbol{x}_n$ is associated with a latent variable $\boldsymbol{z}_n$



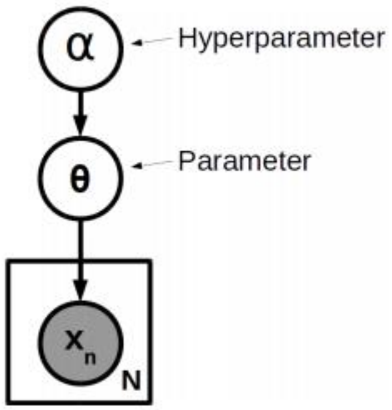
The latent variable $\boldsymbol{z}_n$ can be used to encode some property of $\boldsymbol{x}_n$ (e.g., its cluster membership, or its low-dim representation, or missing parts of $\boldsymbol{x}_n$)

- Such models are used in many problems, especially unsupervised learning: Gaussian mixture model, probabilistic PCA, topic models, deep generative models, etc.

- We will look at several of these in this course and way to learn such models

Hyperparameter Estimation

- ML models invariably have hyperparams, e.g., regularization h.p. in a linear regression model, or kernel h.p. in nonlinear regression of kernel SVM, etc.
- Can specify the hyperparams as additional unknown of the probabilistic model



A way to find the point estimate of the hyperparameters by maximizing the marginal likelihood of data (more on this later)

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmax}} \log p(\mathbf{X}|\alpha)$$
$$= \underset{\alpha}{\operatorname{argmax}} \log \int p(\mathbf{X}|\theta)p(\theta|\alpha)\theta$$

- Can now estimate them, e.g., using a point estimate or a posterior distribution
  - To find point estimate of hyperparameters, we can write the probability of data as a function of hyperparameters and maximize this quantity w.r.t. the hyperparameters (details later)
  - Posterior can also be estimated if we specify a prior on the hyperparameters as well (details later)

Model Comparison

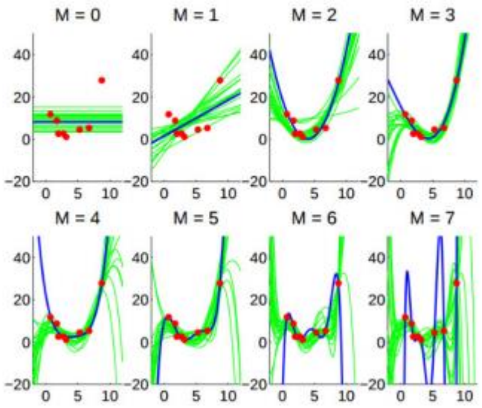
- Suppose we have a number of models $m = 1,2,\ldots,M$ to choose from
- The standard way to choose the best model is cross-validation
- Can also compute the posterior probability of each candidate model, using Bayes rule

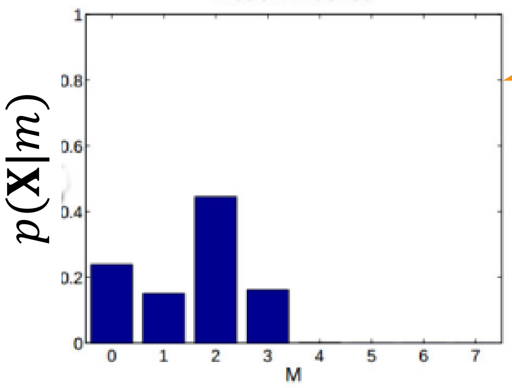
May not be easy to do exactly but can compute it approximately

$$p(m|\mathbf{X}) = \frac{p(m)p(\mathbf{X}|m)}{p(\mathbf{X})}$$

Marginal likelihood of model m

- If all models are equally likely a priori ($p(m)$ is uniform) then the best model can be selected as the one with largest marginal likelihood





This doesn't require a separate validation set unlike cross-validation

Therefore also useful for doing model selection/comparison for unsupervised learning problems

Tentative Outline

- Basics of probabilistic modeling and inference
  - Common probability distributions
  - Basic point estimation (MLE and MAP)
- Bayesian inference (simple and not-so-simple cases)
- Probabilistic models for regression and classification
- Probabilistic Graphical Models
- Gaussian Processes (probabilistic modeling meets kernels)
- Latent Variable Models (for i.i.d., sequential, and relational data)
- Approximate Bayesian inference (EM, variational inference, sampling, etc)
- Bayesian Deep Learning
- Nonparametric Bayesian methods
- Misc topics, e.g., deep generative models, black-box inference, sequential decision-making, reinforcement learning, etc

Coming Up Next

- Basics of probabilistic modeling and inference
  - Terminology
  - Basic methods for parameter estimation (point estimation and posterior)
  - Some simple examples