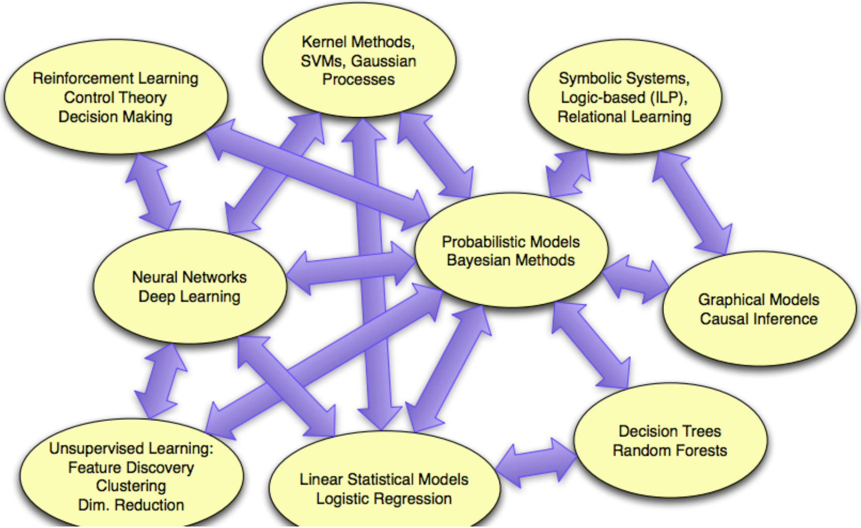
Lecture : Probabilistic Machine Learning

Riashat Islam

Reasoning and Learning Lab
McGill University

September 12, 2017

ML : Many Methods with Many Links



Modelling Views of Machine Learning

Machine Learning is the science of learning models from data

- Define space of possible models
- Learn parameters and structure of models from data
- Make predictions and decisions

Probabilistic Modelling

Provides a framework for understanding what learning is. It describes how to represent and manipulate uncertainty about models and predictions.

- Model describes data one could observe from a system
- We use mathematics of probability theory to express all forms of uncertainty and noise associated with the model
- Use inverse probability (Bayes rule) to infer unknown quantities, adapt our models, make accurate predictions and learn from data

Bayes Rule

- Bayes rule tells us how to do inference about **hypothesis** (the uncertain quantities) from **data** (measured quantities)
- Learning and prediction can be seen as a form of inference

$$P(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis})P(\text{hypothesis})}{P(\text{data})}$$

Rev'd Thomas Bayes (1702–1761)

Why do we care?

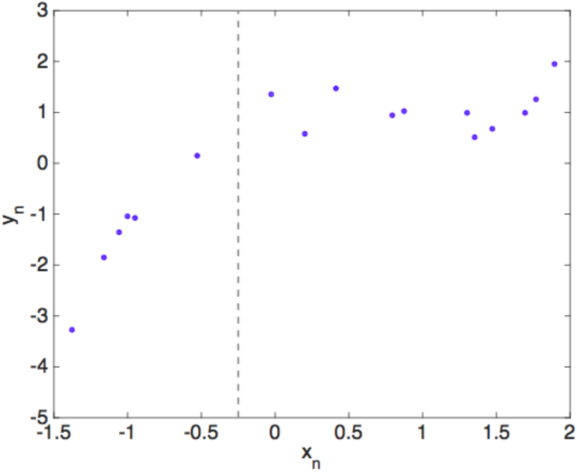
A robot, in order to behave intelligently, should be able to represent beliefs about propositions in the world :

- "charging station is at location (x,y,z)
- "range finder is malfunctioning"
- "that stormtrooper is hostile"

Using probabilistic models, we want to represent the **strengths** of these beliefs, and be able to manipulate these beliefs

- Probabilistic learning can also be used for **calibrated models** and **prediction uncertainty** - getting systems that know what they don't know (more on this later...)

Recall : How do we fit this dataset?



Goal: Predict target y_* associated to any arbitrary input x_*

Recall : Regression Problems

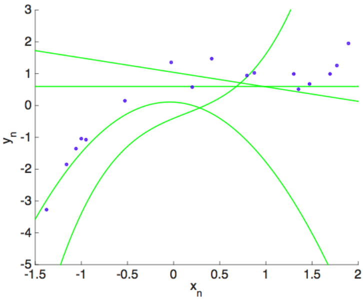
Data : $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$

- Given x_i
- y_i related to x_i by some function
- Plus noise independent of each i

Possible approaches :

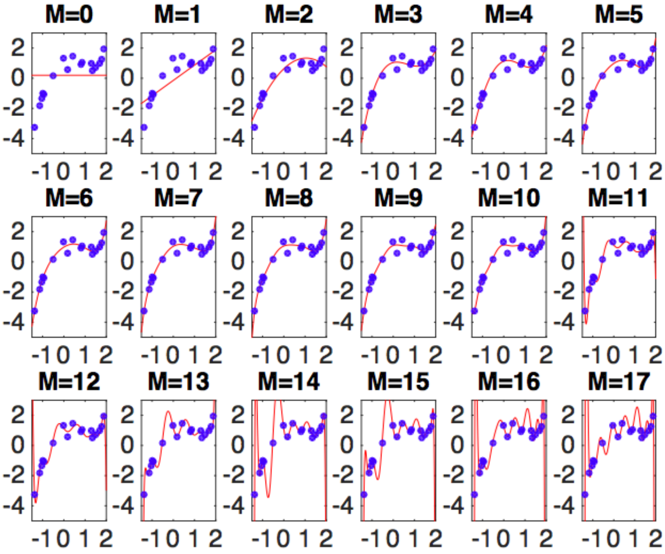
- Use a simple parametric model : y_i is a linear function of x_i plus noise
- Use a non-parametric method : e.g nearest-neighbour or other sophisticated local smoothing models (later...)
- Use a flexible model for the conditional distribution of y_i given x_i - e.g Gaussian Processes, Neural Networks (later...)

Recall : Model of the data

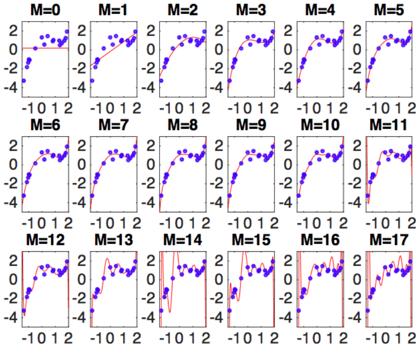


- To predict a new x_* we need to postulate a model of the data. We will estimate y_* with $f(x_*)$
- $f(x)$: Example : A polynomial
-
$$f_w(x) = w_0 + w_1 x + w_2 x^2 + \ldots w_M x^M \tag{1}$$
- w_M are the weights of the polynomial, the **parameters** of the model

Recall : Least squares fit for polynomials



Recall : Least squares fit for polynomials



- Should we choose a polynomial?
- What degree should we choose for the polynomial?
- For a given degree, how do we choose the weights?

Recall : Least squares fit for polynomials

Idea : measure the quality of the fit to the training data

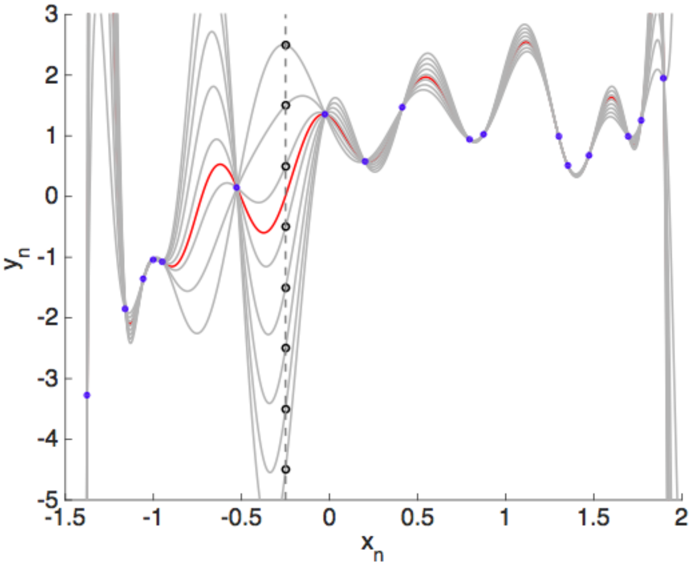
$$e_n^2 = (y_n - f(x_n))^2 \tag{2}$$

Find model parameters which minimizes the sum of squared errors

$$E(w) = \sum_{n=1}^{N} e_n^2 \tag{3}$$

$$f_w(x) = w_0 1 + w_2 x + w_3 x^2 ... = \sum_{m=0}^{M} w_m \phi_m(x) \tag{4}$$

Recall : Overfitting



Recall : Regularization

Intuition : Complicated hypothesis lead to overfitting
Idea : change error function to **penalize hypothesis complexity**

$$J(w) = J_D(w) + \lambda J_{pen}(w) \tag{5}$$

λ controls how much we value fitting the data well vs having a simple hypothesis

Some Open Questions

Do we think that all models are equally probable...before we see any data?

- What does the probability of a model even mean?

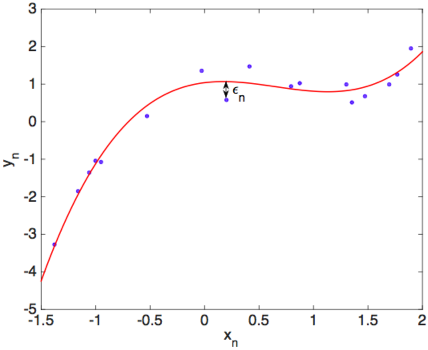
Do we need to choose a single ”best” model or can we consider several?

- We need a framework to answer such questions

Perhaps our training targets are contaminated with noise. What to do?

- We will start here

Observation Noise



- Assume the data is generated by the **red** function
- Each f(x) was independently contaminted by a noise term ϵ_n
- The observations are noisy

$$y_n = f_w(x_n) + \epsilon_n \tag{6}$$

- We characterize the noise by a probability density function $\epsilon_n \sim N(\epsilon_n; 0, \sigma^2_{noise})$

Probability of Observed Data given Model

Given that $y = f + \epsilon$. We can write the probability of $\mathbf{y}$ given $\mathbf{f}$

$$\begin{aligned} p(\mathbf{y}|\mathbf{f}, \sigma^2_{\text{noise}}) = \mathcal{N}(\mathbf{y};\ \mathbf{f}, \sigma^2_{\text{noise}}) &= \left(\frac{1}{\sqrt{2\pi\,\sigma^2_{\text{noise}}}}\right)^N \exp\left(-\frac{\|\mathbf{y}-\mathbf{f}\|^2}{2\,\sigma^2_{\text{noise}}}\right) \\ &= \left(\frac{1}{\sqrt{2\pi\,\sigma^2_{\text{noise}}}}\right)^N \exp\left(-\frac{E(\mathbf{w})}{2\,\sigma^2_{\text{noise}}}\right) \end{aligned}$$

E(w) is the sum of squared errors

$$E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - f_w(x_n))^2 = ||\mathbf{y} - \mathbf{\Phi w}||^2 \qquad (7)$$

Since $\mathbf{f} = \mathbf{\Phi w}$, we can write $p(\mathbf{y}|\mathbf{w}, \sigma^2_{noise}) = p(\mathbf{y}|\mathbf{f}, \sigma^2_{noise})$ for a given $\mathbf{\Phi}$

Statistical Parameter Fitting

- Given instances $x_1, x_2, ...x_m$ that are **i.i.d**
- Find a set of parameters w such that the data can be summarized by a probability $P(\mathbf{x}|w)$
- Parameters w depends on the family of probability distributions we consider (e.g Multinomial, Gaussian)
- For regression and supervised methods, we have special target variables we are interested in $P(y|\mathbf{x}, \mathbf{w})$

Likelihood Function

The **likelihood** of the parameters is the probability of the data given the parameters

- $p(\mathbf{y}|\mathbf{w}, \sigma^2_{noise})$ is the probability of the observed data given the weights
- $L(\mathbf{w}) \propto p(\mathbf{y}|\mathbf{x}, w, \sigma^2_{noise})$ is the likelihood of the weights

- Suppose we assume that the noise in the regression model is Gaussian (normal) with mean zero and some variance σ^2
- We therefore write down the likelihood function for the parameters **w**, which is the joint probability density of all the y_i as a function of **w**

$$\begin{aligned} L(w) &= P(y_1, ...y_n|x_1, ...x_n, w) \\ &= \prod_{i=1}^{n} N(y_i|w^T\phi(x_i), \sigma^2) \end{aligned} \tag{8}$$

Maximum Likelihood Estimation

The probability of y given f is therefore

$$p(y|f,\sigma^2) = N(y;f,\sigma^2) = (\frac{1}{\sqrt{2\pi\sigma^2}})\exp(-\frac{E(w)}{2\sigma^2}) \qquad (9)$$

where E(w) is the sum of squared errors $E(w) = \|\mathbf{y} - \mathbf{w}^{\mathbf{T}}\phi(\mathbf{x})\|^2$

Maximum Likelihood : We can fit the model weights to the data by maximising the likelihood

$$\hat{w} = \arg\max L(w) = \arg\max \exp(-\frac{E(w)}{2\sigma^2}) = \arg\min E(w) \qquad (10)$$

- With additive Gaussian independent noise model, the **maximum likelihood** and **least squares** solutions are same
- We still have not solved the prediction problem!
  We still overfit.

Probabilistic Approach

So far, we looked at a probabilistic approach to learn a parametric form of a function that could fit the data.

- For a Gaussian noise model, this approach will make the same predictions as using the squared loss error function

$$\log p(y|x, w, \sigma^2) \propto -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(f(x_i, w) - y(x_i))^2 \tag{11}$$

Regularization

- We still need to use a penalized loss function for regularization

$$\log p(\mathbf{y}|X, \mathbf{w}) \propto \overbrace{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(f(x_i, \mathbf{w}) - y(x_i)^2)}^{\text{model fit}} \overbrace{-\lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}}^{\text{complexity penalty}} .$$

Probabilistic Approach

- The probabilistic approach helps us interpret the error measure in a deterministic approach, and gives us a sense of the noise level σ^2
- Probabilistic methods thus provides an intuitive framework for representing uncertainty, and model development

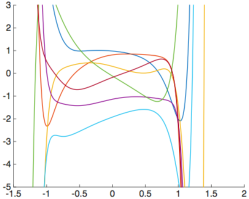
However, both approaches so far are prone to **over-fitting** for flexible $f(x, w)$: low error on the training data, high error on the test set

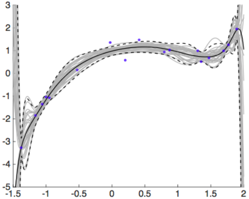
Multiple Explanations of the Data

- We do not believe all models are equally probable to explain the data
- We may believe that a simpler model is more probable than a complex one

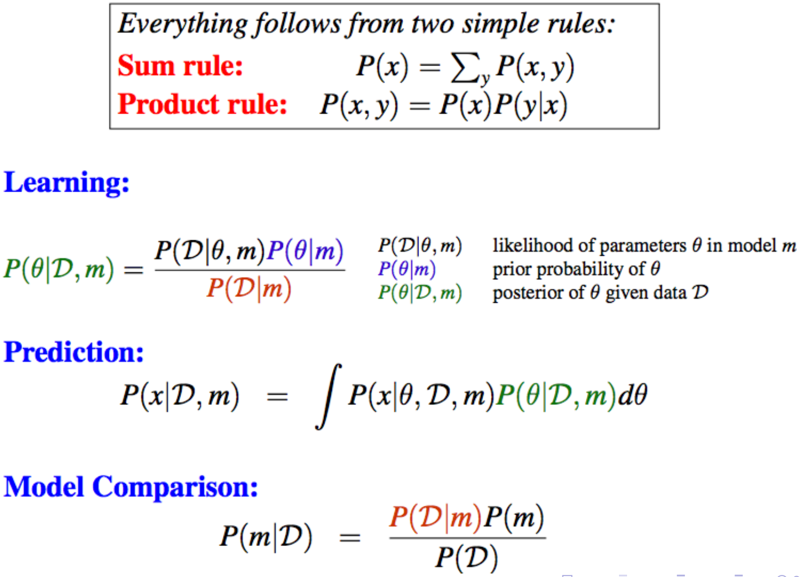
Model complexity and uncertainty :

- We do not know what particular function generated the data
- More than one of our models can perfectly fit the data
- We believe more than one of our models could have generated the data
- We want to reason in terms of a set of possible explanations, not just one





Bayesian Machine Learning



Everything follows from two simple rules:

Sum rule: $P(x) = \sum_y P(x,y)$

Product rule: $P(x,y) = P(x)P(y|x)$

Learning:

$$P(\theta|\mathcal{D},m) = \frac{P(\mathcal{D}|\theta,m)P(\theta|m)}{P(\mathcal{D}|m)}$$

$P(\mathcal{D}\|\theta,m)$	likelihood of parameters θ in model m
$P(\theta\|m)$	prior probability of θ
$P(\theta\|\mathcal{D},m)$	posterior of θ given data $\mathcal{D}$

Prediction:

$$P(x|\mathcal{D},m) = \int P(x|\theta,\mathcal{D},m)P(\theta|\mathcal{D},m)d\theta$$

Model Comparison:

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}$$

What does it mean by being Bayesian

- Dealing with all sources of parameter uncertainty
- Potentially dealing with structure uncertainty

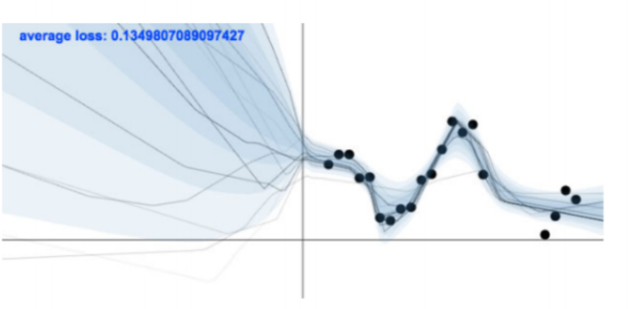


Figure: Yarin Gal's thesis, Uncertainty in Deep Learning (2016)

Bayesian View

- Formulate knowledge about situation probabilistically
  - Define model that expresses qualitative aspects of our knowledge. The model will have unknown parameters
  - Specify a prior probability distribution for these unknown parameters that expresses our beliefs about which values are likely, before seeing the data
- Gather the data
- Compute a posterior distribution for the parameters given the observed data
- Use this posterior distribution to
  - Make predictions by averaging over the posterior distribution
  - Make decisions so as to minimize posterior expected loss

Maximum A Posteriori (MAP)

Recall in MLE : Choose value that maximizes the probability of observed data

$$\theta_{MLE} = \arg\max_{\theta} P(D|\theta) \tag{12}$$

In **MAP estimates**, we use prior beliefs. Choose value that is most probable given observed data and prior belief

$$\theta_{MAP} = \arg\max_{\theta} P(\theta|D) = \arg\max_{\theta} P(D|\theta)P(\theta) \tag{13}$$

In MAP, we seek the value of θ which maximizes the posterior $P(\theta|D)$.

- It estimates θ as the *mode* of the posterior distribution
- These are *point estimates* and not *fully Bayesian*. In Bayesian methods, we use distributions to summarize and draw inferences from data

Prior on Parameters induce Priors on Functions

A model $\mathcal{M}$ is the choice of a model structure and of parameter values.

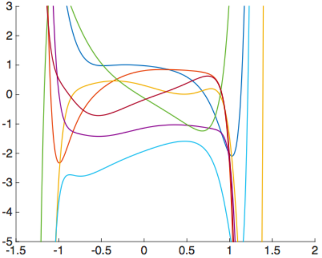
$$f_{\mathbf{w}}(x) = \sum_{m=0}^{M} w_m \, \phi_m(x)$$

The prior $p(\mathbf{w}|\mathcal{M})$ determines what functions this model can generate. Example:

- Imagine we choose $M = 17$, and $p(w_m) = \mathcal{N}(w_m;\ 0, \sigma_{\mathbf{w}}^2)$.
- We have actually defined a prior distribution over functions $p(\mathbf{f}|\mathcal{M})$.

This figure is generated as follows:

- Use polynomial basis functions, $\phi_m(x) = x^m$.
- Define a uniform grid of $n = 100$ values in x from $[-1.5, 2]$.
- Generate matrix $\mathbf{\Phi}$ for $M = 17$.
- Draw $w_m \sim \mathcal{N}(0, 1)$.
- Compute and plot $\mathbf{f} = \mathbf{\Phi}_{n\times 18}\,\mathbf{w}$.



Conjugate Priors

For most Bayesian inference problems, the integrals needed to do inference and prediction are not analytically tractable — hence the need for numerical quadrature, Monte Carlo methods, or various approximations.

Most of the exceptions involve *conjugate priors*, which combine nicely with the likelihood to give a posterior distribution of the same form. Examples:

1) Independent observations from a finite set, with Beta / Dirichlet priors.

2) Independent observations of Gaussian variables with Gaussian prior for the mean, and either known variance or inverse-Gamma prior for the variance.

3) Linear regression with Gaussian prior for the regression coefficients, and Gaussian noise, with known variance or inverse-Gamma prior for the variance.

It's nice when a tractable model and prior are appropriate for the problem. Unfortunatley, people are tempted to use such models and priors even when they aren't appropriate.

Finding the Posterior Distribution

Given the prior over functions (parameters), we want to find the posterior distribution and use it to make predictions

The posterior distribution for the model parameters given the observed data is found by combining the prior distribution with the likelihood for the parameters given the data (Bayes rule)

$$P(\text{parameters} \,|\, \text{data}) = \frac{P(\text{parameters})\, P(\text{data} \,|\, \text{parameters})}{P(\text{data})}$$

The denominator is just the normalizing constant. So as a proportionality, we can write

$$P(\text{parameters} \,|\, \text{data}) \propto P(\text{parameters})\, P(\text{data} \,|\, \text{parameters})$$

Predictive Distribution

Posterior distribution

$$p(\mathbf{w}|\mathbf{y},X,\sigma^2) = \frac{p(\mathbf{y}|X,\mathbf{w},\sigma^2)p(\mathbf{w})}{p(\mathbf{y}|X,\sigma^2)}$$

We make predictions by integrating with respect to the posterior

$$P(\text{new data}\,|\,\text{data}) \quad = \int_{\text{parameters}} P(\text{new data}\,|\,\text{parameters})\,P(\text{parameters}\,|\,\text{data})$$

$$p(y|x_*,\mathbf{y},X) = \int p(y|x_*,\mathbf{w})p(\mathbf{w}|\mathbf{y},X)d\mathbf{w}$$

- Think of each setting of w as a different model.
- Equation above is a **Bayesian model average**, an average of infinitely many models weighted by their posterior probabilities
- No overfitting - automatically calibrated complexity

Maximum Likelihood, Parametric Model

- data : **x**, **y**
- model M : $y = f_w(x) + \epsilon$

Gaussian Likelihood

$$p(y|x, w, M) \propto \prod_{n=1}^{N} \exp(-\frac{1}{2}(y_n - f_w(x_n))^2/\sigma^2_{noise} \qquad (14)$$

Maximize the likelihood

$$w_{ML} = argmax_w p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) \qquad (15)$$

Make predictions by plugging in the ML estimate

$$p(y_*|x_*, w_{ML}, M) \qquad (16)$$

Bayesian Inference, Parametric Model

- data : $\mathbf{x}$, $\mathbf{y}$
- model M : $y = f_w(x) + \epsilon$

Gaussian Likelihood

$$p(y|x, w, M) \propto \prod_{n=1}^{N} \exp(-\frac{1}{2}(y_n - f_w(x_n))^2/\sigma^2_{noise} \tag{17}$$

Parameter prior $p(w|M)$

Posterior parameter distribution (Bayes rule)

$$p(\mathbf{w}|\mathbf{x}, \mathbf{y}, \mathbf{M}) = \frac{p(\mathbf{w}|M)p(\mathbf{y}|\mathbf{w}, \mathbf{x}, M)}{p(\mathbf{y}|\mathbf{x}, M)} \tag{18}$$

Making predictions (marginalizing out the parameters):

$$p(y_*|x_*, x, y, M) = \int p(y_*|w, x_*, M)p(w|x, y, M)dw \tag{19}$$

Prior, Likelihood, Posterior and Predictive distributions are all Gaussian Distributions.

Posterior and Predictive Distributions

For a linear-in-the-parameters model with Gaussian priors and Gaussian noise:

- Gaussian *prior* on the weights: $p(\mathbf{w}|\mathcal{M}) = \mathcal{N}(\mathbf{w};\ \mathbf{0},\ \sigma_{\mathbf{w}}^2\,\mathbf{I})$
- Gaussian *likelihood* of the weights: $p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathcal{M}) = \mathcal{N}(\mathbf{y};\ \mathbf{\Phi}\,\mathbf{w},\ \sigma_{\text{noise}}^2\,\mathbf{I})$

Posterior parameter distribution by Bayes rule $p(a|b) = p(a)p(b|a)/p(b)$:

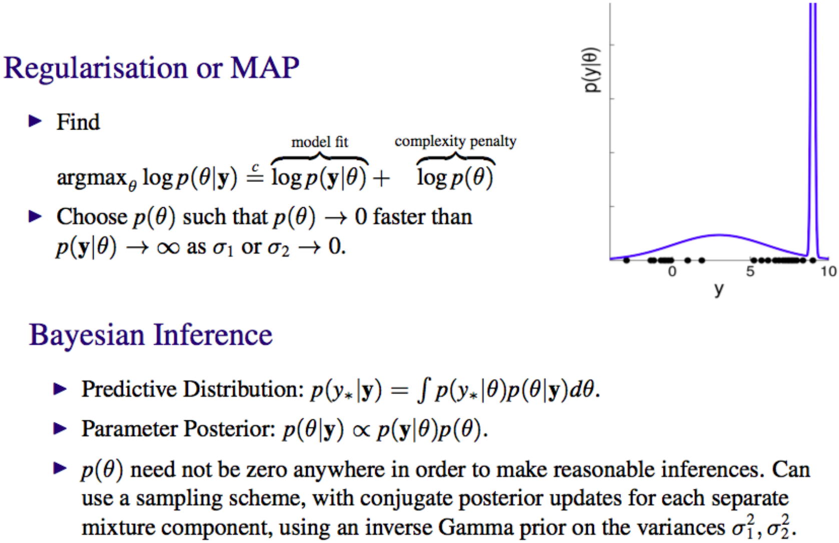
$$p(\mathbf{w}|\mathbf{x},\mathbf{y},\mathcal{M}) \;=\; \frac{p(\mathbf{w}|\mathcal{M})p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathcal{M})}{p(\mathbf{y}|\mathbf{x},\mathcal{M})} \;=\; \mathcal{N}(\mathbf{w};\ \boldsymbol{\mu},\ \mathbf{\Sigma})$$

$$\mathbf{\Sigma} \;=\; \big(\sigma_{\text{noise}}^{-2}\mathbf{\Phi}^\top\mathbf{\Phi} + \sigma_{\mathbf{w}}^{-2}\,\mathbf{I}\big)^{-1} \quad \text{and} \quad \boldsymbol{\mu} \;=\; \Big(\mathbf{\Phi}^\top\mathbf{\Phi} + \frac{\sigma_{\text{noise}}^2}{\sigma_{\mathbf{w}}^2}\,\mathbf{I}\Big)^{-1}\mathbf{\Phi}^\top\mathbf{y}$$

The predictive distribution is given by:

$$p(y_*|x_*,\mathbf{x},\mathbf{y},\mathcal{M}) \;=\; \mathcal{N}(y_*;\ \boldsymbol{\phi}(x_*)^\top\boldsymbol{\mu},\ \boldsymbol{\phi}(x_*)^\top\mathbf{\Sigma}\boldsymbol{\phi}(x_*) + \sigma_{\text{noise}}^2)$$

Regularisation $=$ MAP $\neq$ Bayesian Inference



Regularisation or MAP

- Find

$$\operatorname{argmax}_\theta \log p(\theta|\mathbf{y}) \overset{c}{=} \overbrace{\log p(\mathbf{y}|\theta)}^{\text{model fit}} + \overbrace{\log p(\theta)}^{\text{complexity penalty}}$$

- Choose $p(\theta)$ such that $p(\theta) \to 0$ faster than $p(\mathbf{y}|\theta) \to \infty$ as σ_1 or $\sigma_2 \to 0$.

Bayesian Inference

- Predictive Distribution: $p(y_*|\mathbf{y}) = \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta$.
- Parameter Posterior: $p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)$.
- $p(\theta)$ need not be zero anywhere in order to make reasonable inferences. Can use a sampling scheme, with conjugate posterior updates for each separate mixture component, using an inverse Gamma prior on the variances σ_1^2, σ_2^2.

Representing Prior and Posterior by Samples

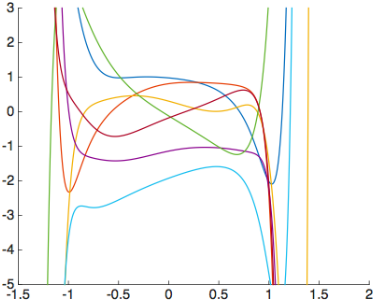
The complex distributions we will often use as priors, or obtain as posteriors, may not be easily represented

A general technique is to represent a distribution by **sampling** of many values drawn randomly from it. We can then

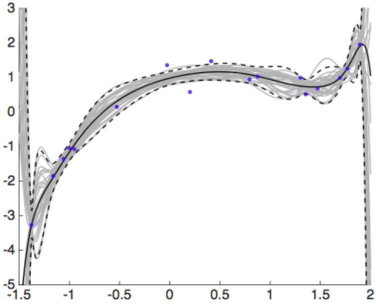
- Visualize the distribution by viewing these sample values, or low dimensional projections of them (PCA..later)
- Make *Monte Carlo* estimates for probabilities or expectations with respect to the distribution, by taking averages over these sample values

Obtaining a sample from the prior is easy! Obtaining a sample from the posterior is usually more difficult - nevertheless a dominant approach to Bayesian computation

Representing Prior and Posterior by Samples



Some samples from the prior



Samples from the posterior

Comparing Models : Marginal Likelihood

What if we are unsure which model is right? So far we assumed we were able to start by making a definite choice of model.

We can compare models based on **marginal likelihoods** (also known as model evidence) for each model - this is the probability the model assigns to the observed data. This is the normalizing constant in Bayes rule which we ignored previously

$$P(\text{data} \mid M_1) \;=\; \int_{\text{parameters}} P(\text{data} \,|\, \text{parameters}, M_1)\, P(\text{parameters} \,|\, M_1)$$

Here, M_1 represents the condition that model M_1 is the correct one. Similarly, we can compute $P(data|M_2)$ for some other model

- We might choose the model that gives higher probability to the data, or average predictions from both sides with weights based on the marginal likelihood, multiplied by any prior preference we have over M_1 versus M_2

Marginal Likelihood

From looking at the equation of posterior distribution, the marginal likelihood is given by

$$p(\mathbf{y}|\mathbf{x}, M) = \int p(\mathbf{w}|M)p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M)dw \tag{20}$$

Second level inference : model comparison

$$p(M|\mathbf{y}, \mathbf{x}) = \frac{p(\mathbf{y}|\mathbf{x}, M)p(M)}{p(y|x)} \propto p(\mathbf{y}|\mathbf{x}, M)p(M) \tag{21}$$

The marginal likelihood is used to **select between models**

Conclusions

Probability machine learning provides a framework for

- making inferences from data in a model
- making probabilistic predictions

Thank You

Slides dedicated to David MacKay

I wish I had known you better...

Information Theory, Inference, and Learning Algorithms