Bayesian approaches to machine learning

Pierre Geurts

Institut Montefiore, University of Liège, Belgium

INFO8004
Advanced Machine Learning
Feb 15, 2018

Outline

Outline

Bayesian methods: general principles

The main idea of Bayesian methods is to use probability theory to model all kind of uncertainties, ie.,

- Uncertainties due to **physical randomness** associated with the data, e.g., noises, errors in labeling.
- But also uncertainties about **models** and their **parameters**.

Then, all prediction problems are reduced to making probabilistic inference using this model (by inverting conditional probabilities through Bayes' rule).

A non-Bayesian approach would only model the first kind of uncertainties using probability theory and will not explicitly model uncertainties in model and parameters.

Bayesian methods: modelling

Let assume that we have observed some data $\mathcal{D}$.

This data can be for example a set of points from IR^p
$\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ (but it does not need to be a set).

1) We construct a model m that can generate/predict the data. This model has a number of free parameters, denoted collectively by θ.

Using this model, one can compute the probability of the data for any set of parameters, ie.:

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

This is called the **likelihood** of the parameters.

Example:
In the context of supervised learning, m_1 could be a linear model $y = w_0 + w_1 x + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and its parameters $\theta_1 = (w0, w_1, \sigma^2)$ and m_2 could be a degree-2 polynomial model $y = w_0 + w_1 x + w_2 x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and its parameters $\theta_2 = (w_0, w_1, w_2, \sigma^2)$.

Bayesian methods: modelling

2) We specify a probability distribution for these unknown parameters that expresses our beliefs about which values are more or less likely, **before seeing the data**:

$$P(\theta|m).$$

This distribution is the called the **prior**.

Example: the coefficient of the quadratic term in m_2 should be small: $w_2 \sim \mathcal{N}(0, s^2)$, with s^2 small.

NB: Probabilities can not be interpreted here as limit of relative frequencies of event in a large number of trials but instead are used to express our belief or our uncertainty about what the value of a given parameter should be.

Bayesian methods: inference

3) From the likelihood, the prior, and the data $\mathcal{D}$, we compute the **posterior distribution** $P(\theta|\mathcal{D}, m)$ using Bayes' theorem:

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

Bayesian learning transforms prior knowledge $P(\theta|m)$ about the parameters into posterior knowledge $P(\theta|\mathcal{D}, m)$, by exploiting the data.

The posterior expresses our belief/uncertainty in the parameters **once we have seen the data**. One of the most distinctive features of Bayesian methods is to keep the full posterior distribution instead of extracting a point estimate from it.

$P(\mathcal{D}|m)$ acts as a normalizing constant and is called the **marginal likelihood**. It can be computed as:

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

Bayesian methods: inference

4) We use this posterior distribution to:

- Make predictions by averaging over the posterior
- Examine/Account for uncertainty in the parameter values
- Make decisions by minimizing expected posterior loss

Predictions of new data points x_{new} are obtained from the **predictive distribution**

$$p(x_{new}|\mathcal{D}, m) = \int p(x_{new}|\theta, \mathcal{D}, m)p(\theta|\mathcal{D}, m)d\theta.$$

computed by integrating over the posterior distribution.

The predictive distribution translates uncertainties in the model parameters into **uncertainties in the predictions**.

Note that this is very different from frequentists' uncertainty, which is uncertainty due to randomness in the data. For a Bayesian, the data is fixed.

Two main challenges

There are two main challenges associated with the application of Bayesian approaches:

Modelling challenge: how to specify suitable models and prior distributions:

- 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.
- If the model/prior are chosen without regard for the actual situation, there is no justification for believing the results of Bayesian inference.

Computational challenge: how to compute the posterior distribution?

- Computing the marginal likelihood $P(\mathcal{D}|m)$ is indeed often intractable.
- Fortunately, several approaches exist to solve this issue.

Maximum Likelihood and Maximum a Posteriori

Two other related popular **non-bayesian** kinds of learning based on point estimates:

- **Maximum likelihood** (ML) learning: find the parameters $\hat{\theta}_{ML}$ that maximizes the likelihood:

$$\hat{\theta}_{ML} = \arg\max_{\theta} P(\mathcal{D}|\theta, m)$$

- **Maximum a posteriori** (MAP) learning: find the parameters $\hat{\theta}_{MAP}$ that maximizes the posterior:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta|\mathcal{D}, m)$$

Both coincide when the prior $p(\theta|m)$ is uniform.

NB: Although it exploits the priors, computing the MAP estimate is not considered as a full Bayesian approach.

Illustration

We would like to learn about the average height of Belgian male adult.
Our data could be a sample of measured heights of Belgian men
$\mathcal{D} = \{y_1, y_2, \ldots, y_n\}$.

The likelihood:
Model m: Each measurement is drawn from a Gaussian distribution of mean μ and variance σ^2, where σ is supposed to be known (to simplify things).
Parameters θ: μ.

$$p(\mathcal{D}|\theta, m) = p(\{y_1, y_2, \ldots, y_N\}|\mu, m) = \prod_{i=1}^{N} \mathcal{N}(y_i|\mu, \sigma^2).$$

The prior:
The average height should be very close to $\mu_o = 175cm$. This can be expressed by a Gaussian prior on μ of mean μ_0 and variance s^2:

$$p(\theta|m) = p(\mu|m) = \mathcal{N}(\mu|\mu_0, s^2),$$

Illustration

The posterior:

$$\begin{aligned} p(\mu|\mathcal{D}, m) &\propto p(\mathcal{D}|\mu, m)p(\mu|m) \\ &\propto \prod_{i=1}^{N} \mathcal{N}(y_i|\mu, \sigma^2)\mathcal{N}(\mu|\mu_0, s^2) \end{aligned}$$

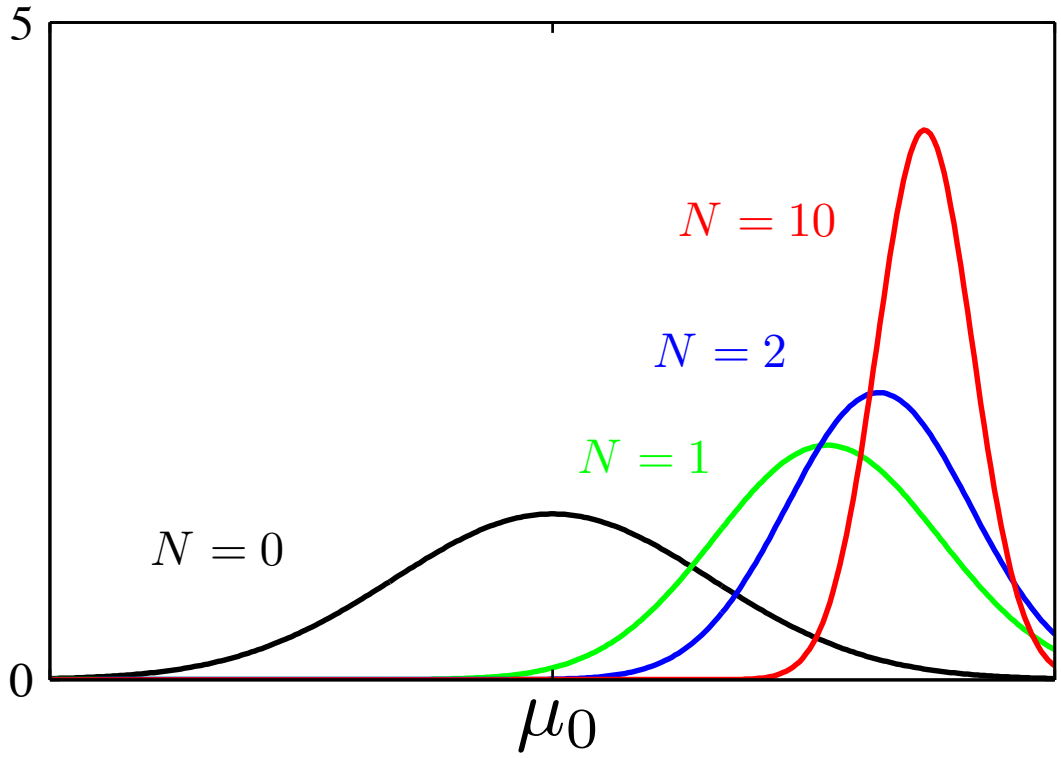
Since the product of two Gaussian densities (on μ here) is a Gaussian, we can show that (on black board):

$$p(\mu|\mathcal{D}, m) = \mathcal{N}(\mu|\mu_N, \sigma_N^2),$$

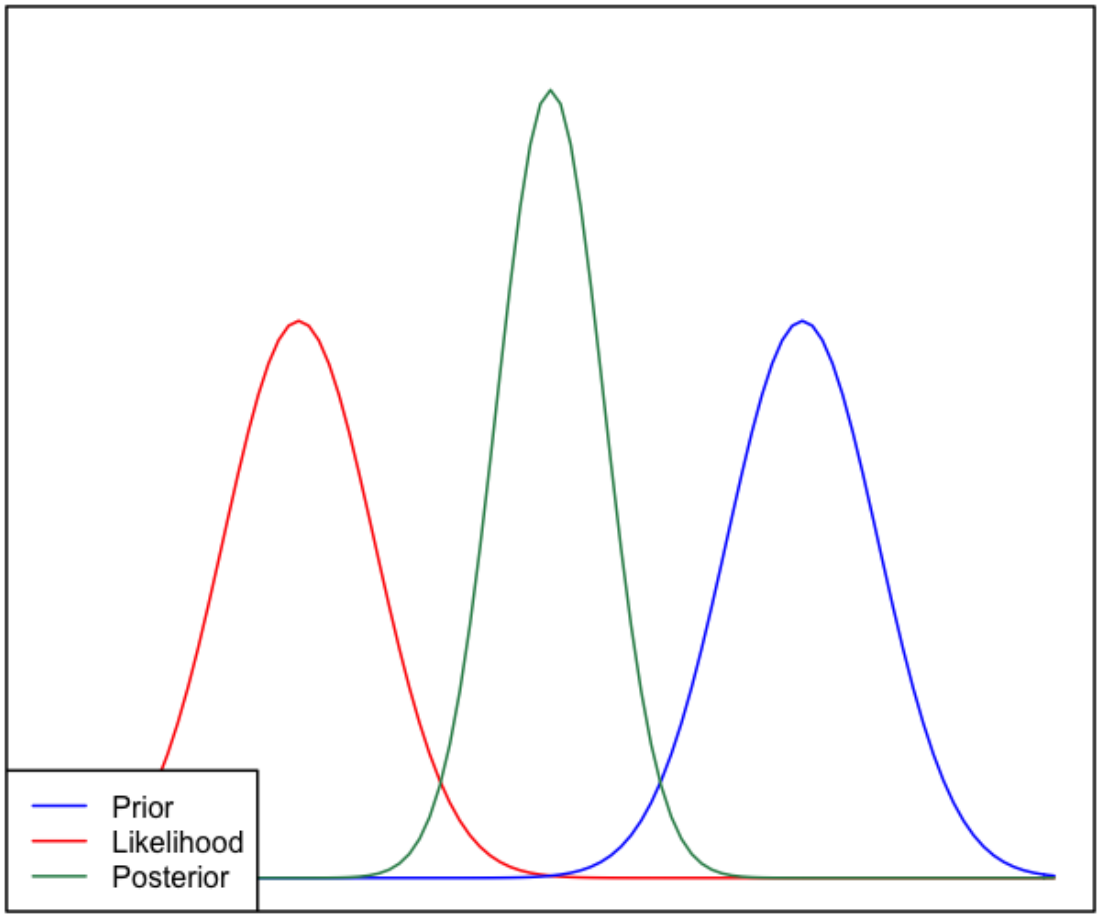
with $\sigma_N^2 = (\frac{1}{\mathbf{s^2}} + \frac{N}{\boldsymbol{\sigma^2}})^{-1}$, $\mu_N = \sigma_N^2(\frac{\boldsymbol{\mu_0}}{\mathbf{s^2}} + \frac{N\bar{y}}{\boldsymbol{\sigma^2}})$, and $\bar{y} = \frac{1}{N}\sum_i y_i$.

- As $N \to +\infty$, μ_N converges to the MLE estimate $\bar{y}$, independently on the prior. Our uncertainty about μ_N also decreases to 0 with N.
- If variance of the prior s^2 increases to infinity, then μ_N also converges to MLE.

Illustration



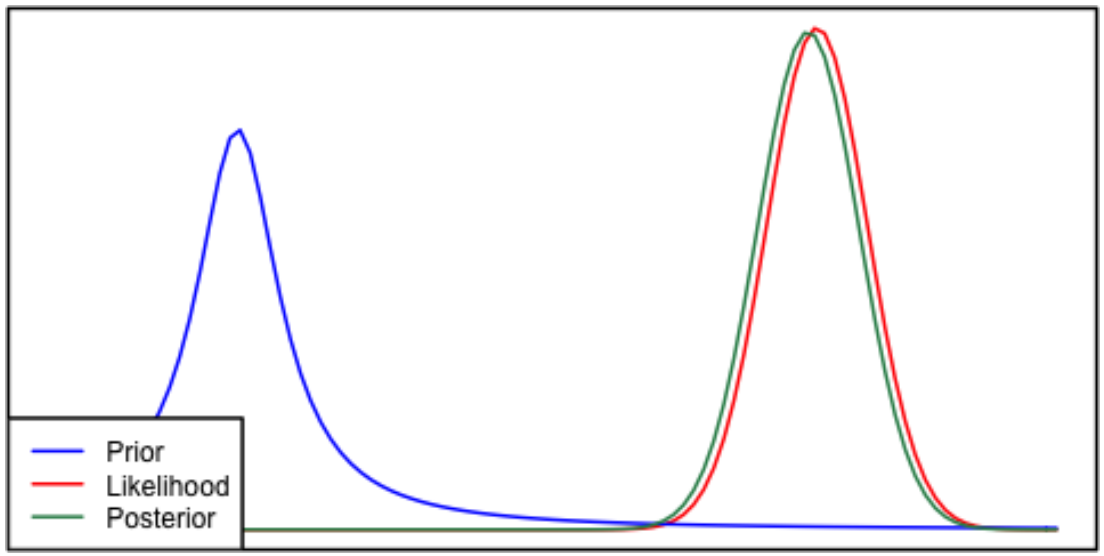
Illustration



A Bayesian is one who, vaguely expecting a horse, and catching a glimpse of a donkey, strongly believes he has seen a mule.

Karl Pearson?

Illustration



Using a prior with a longer tail (here, a Cauchy distribution).

A note on conjugate prior

If the posterior distribution $p(\theta|\mathcal{D}, m)$ is in the same family as the prior distribution $p(\theta|m)$, the prior and posterior are called **conjugate distributions**, and the prior is called a **conjugate prior** for the likelihood.

This is very convenient computationally as in such case, the posterior can be computed analytically (without worrying about computing the marginal likelihood).

Examples of conjugate (prior-likelihood) pairs:

Prior	Likelihood
Gaussian	Gaussian
Beta	Binomial
Gamma	Gaussian
Dirichlet	Multinomial

Outline

Bayesian approaches for supervised learning

In supervised learning (SL), we observe a set of pairs $\mathcal{D} = \{(x_i, y_i)|i = 1, \ldots, N\}$, with $x_i \in \mathcal{R}^p$ and $y_i \in \mathcal{R}$ and we are interested in predicting y for a new x.

In what follows, we will denote the input matrix $X = [x_1, \ldots, x_N]^T \in \mathcal{R}^{N \times p}$ and the output vector $y = [y_1, \ldots, y_N]^T \in \mathcal{R}^N$, such that $\mathcal{D} = \{X, y\}$.

In SL, we are only interested in modelling and predicting the outputs. The X part of the data will always belong to the conditionings, assuming inputs are fixed.

Bayes' theorem applied to compute the posterior will thus take the following form:

$$p(\theta|X, y, m) = \frac{p(y|X, \theta, m)P(\theta|m)}{P(y|X, m)}.$$

(X does not appear in the conditioning of the prior $P(\theta|m)$ as it is not supposed to be used to define this prior)

Bayesian linear regression: likelihood

Linear regression assumes that there is a linear relationship between inputs and output, up to some residuals $\epsilon \in \mathcal{R}^N$:

$$y = Xw + \epsilon,$$

where $w \in \mathcal{R}^p$ are free parameters. the residuals are assumed to be independent, unbiased, and small. This can be modeled with:

$$p(\epsilon|\sigma^2) = \mathcal{N}(\epsilon|0, \sigma^2 I_N),$$

with I_N the identity matrix of size $N \times N$. We will assume that we know the true value of σ^2. The parameters of the model, θ, thus reduce in this case to the vector w.

The **likelihood** is thus written

$$p(y|X, w, \sigma^2) = \mathcal{N}(y|Xw, \sigma^2 I_N)$$

Bayesian linear regression: prior and posterior

A priori, we will assume that entries of w are small and that all directions for w are equally likely. This can be encoded by the following **prior** distribution:

$$p(w|s^2) = \mathcal{N}(w|0, s^2 I_p).$$

We want to compute the **posterior** distribution:

$$p(w|X, y, \sigma^2, s^2) \propto p(y|X, w, \sigma^2) p(w|s^2)$$

This can be done analytically in this specific case. Indeed, since we use a conjugate prior, the posterior will be Gaussian:

$$p(w|X, y, \sigma^2, s^2) = \mathcal{N}(w|\mu_w, \Sigma_w),$$

and μ_w and Σ_w can be computed by identification.

Bayesian linear regression: posterior

From conjugacy:

$$\begin{aligned} p(w|X, y, \sigma^2, s^2) &\propto \exp\left(-\frac{1}{2}(w-\mu_w)^T\Sigma_w^{-1}(w-\mu_w)\right) \\ &\propto \exp\left(-\frac{1}{2}(w^T\Sigma_w^{-1}w - 2\mu_w^T\Sigma_w^{-1}w)\right) \end{aligned}$$

From prior and likelihood definitions:

$$\begin{aligned} p(w|X, y, \sigma^2, s^2) &\propto \mathcal{N}(y|Xw, \sigma^2 I_N)\mathcal{N}(w|0, s^2 I_p) \\ &\propto \exp\left(-\frac{1}{2}(\frac{1}{\sigma^2}(y-Xw)^T(y-Xw) + \frac{1}{s^2}w^Tw)\right) \\ &\propto \exp\left(-\frac{1}{2}(w^T(\frac{1}{\sigma^2}X^TX + \frac{1}{s^2}I_p)w - \frac{2}{\sigma^2}y^TXw)\right) \end{aligned}$$

By identification:

$$\Sigma_w = \left(\frac{1}{\sigma^2}X^TX + \frac{1}{s^2}I_p\right)^{-1},\ \mu_w = \left(\frac{1}{\sigma^2}X^TX + \frac{1}{s^2}I_p\right)^{-1}\frac{1}{\sigma^2}X^Ty$$

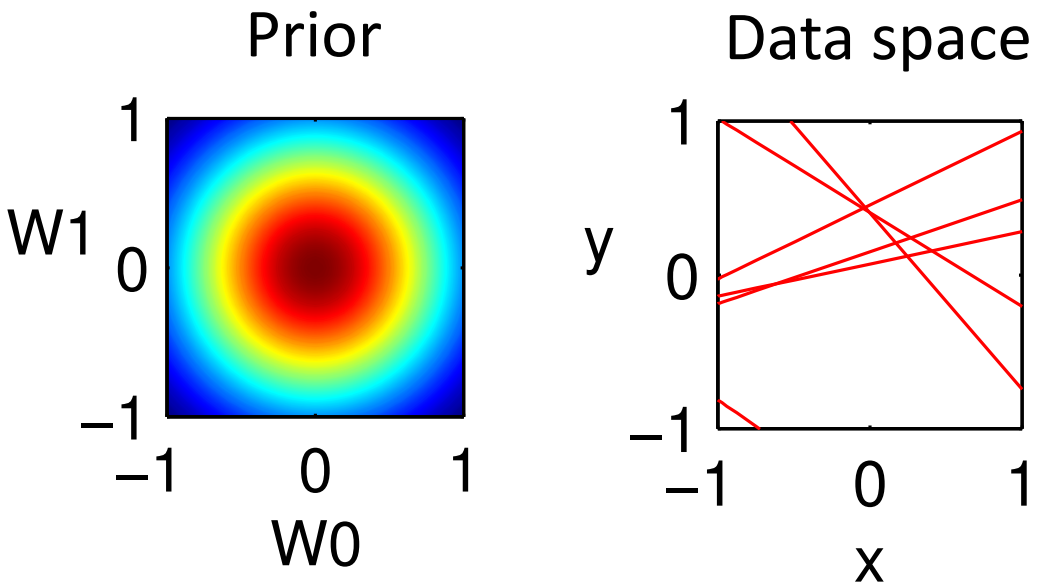
What happens when $N \to +\infty$? When $s^2 \to +\infty$?

Bayesian linear regression: posterior illustration

Let us consider BLR with a model of the form: $y(x, w) = w_0 + w_1 x$.

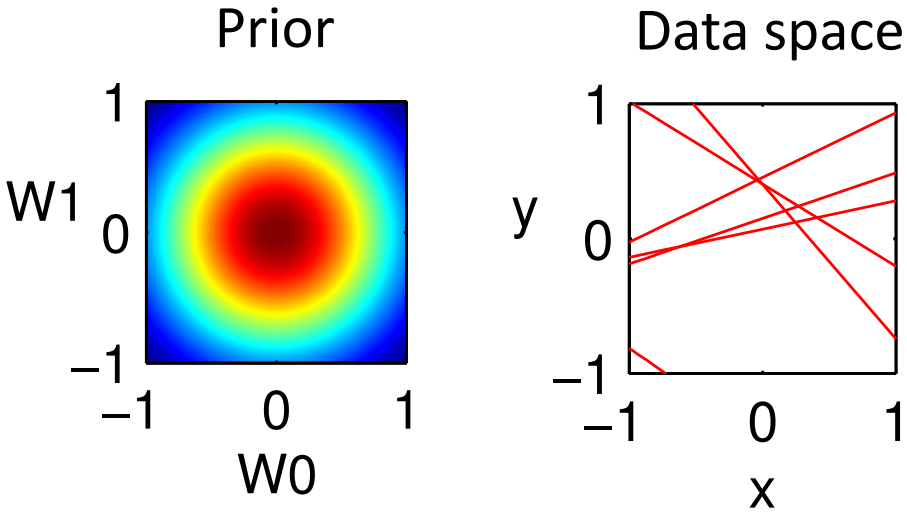
The training data is generated from the function $y(x) = -0.3 + 0.5x$ by first choosing x uniformly from $[-1, 1]$, evaluating $y(x)$, and adding a small Gaussian noise ($\sigma = 0.2$).

When zero data points have been observed:

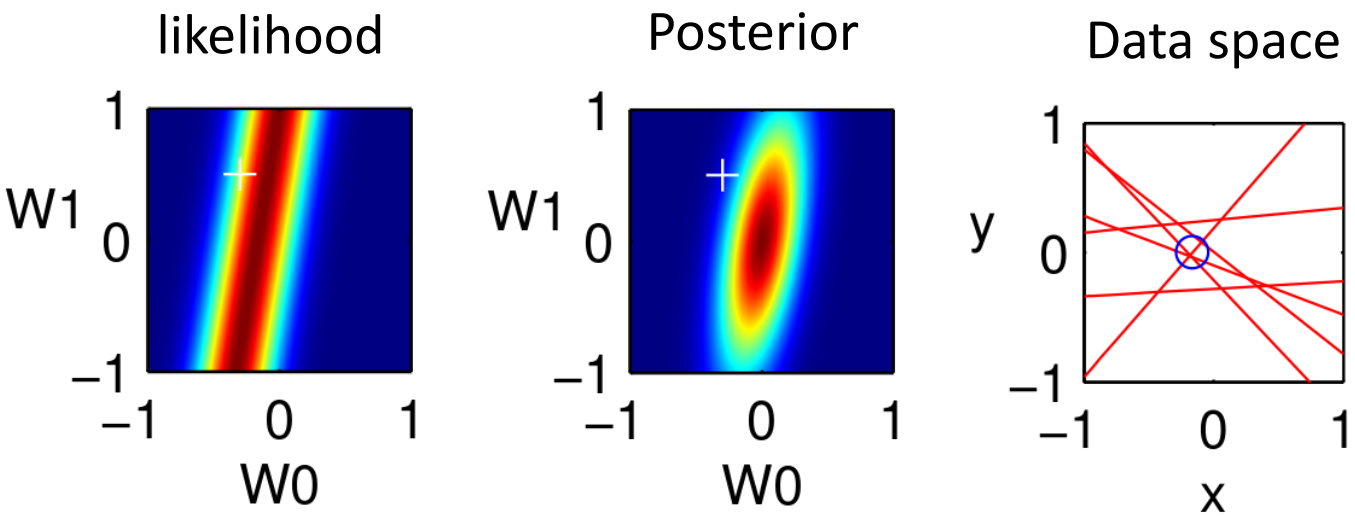


Bayesian linear regression: posterior illustration

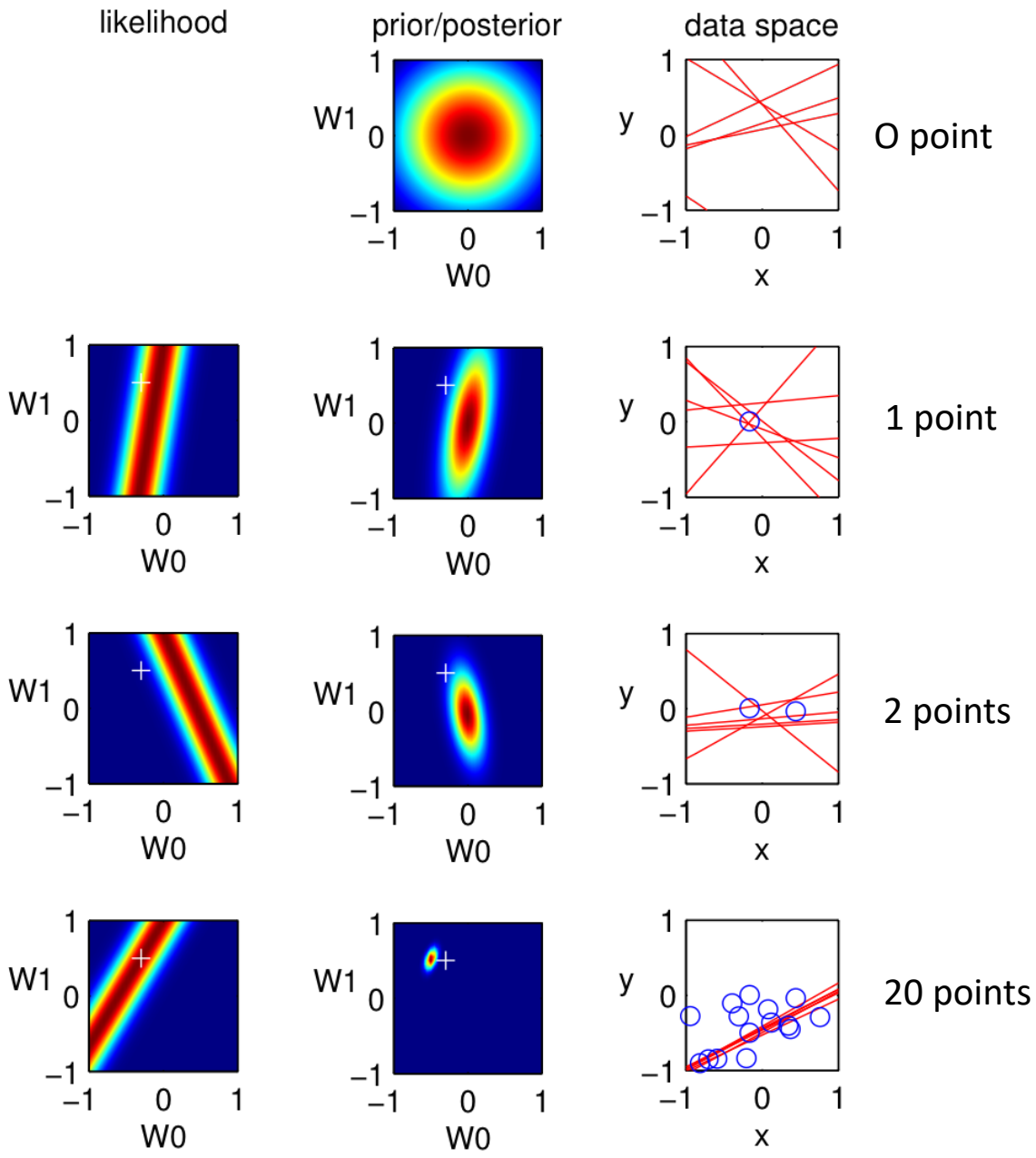
0 data points are observed:



1 data point is observed:



Bayesian linear regression: posterior illustration



(Murphy, 2012)

Bayesian linear regression: predictive distribution

Given a new observation x_{new}, we are interested in the density:

$$p(y_{new}|x_{new}, X, y, \sigma^2, s^2) = \int p(y_{new}|x_{new}, w, \sigma^2)p(w|X, y, \sigma^2, s^2)dw$$

Given the definition of our model, we have $y_{new} = x_{new}^T w + \epsilon$. Given that $w \sim \mathcal{N}(\mu_w, \Sigma_w)$, we have $x_{new}^T w \sim \mathcal{N}(x_{new}^T \mu_w, x_{new}^T \Sigma_w x_{new})$. Since in addition, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is independent of w, we have finally:

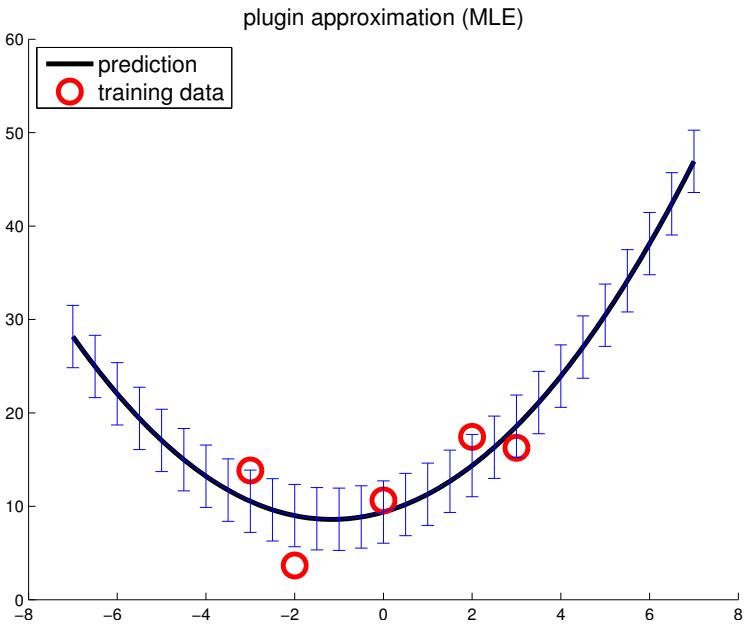
$$p(y_{new}|x_{new}, X, y, \sigma^2, s^2) = \mathcal{N}(y_{new}|x_{new}^T \mu_w, \sigma^2 + x_{new}^T \Sigma_w x_{new})$$

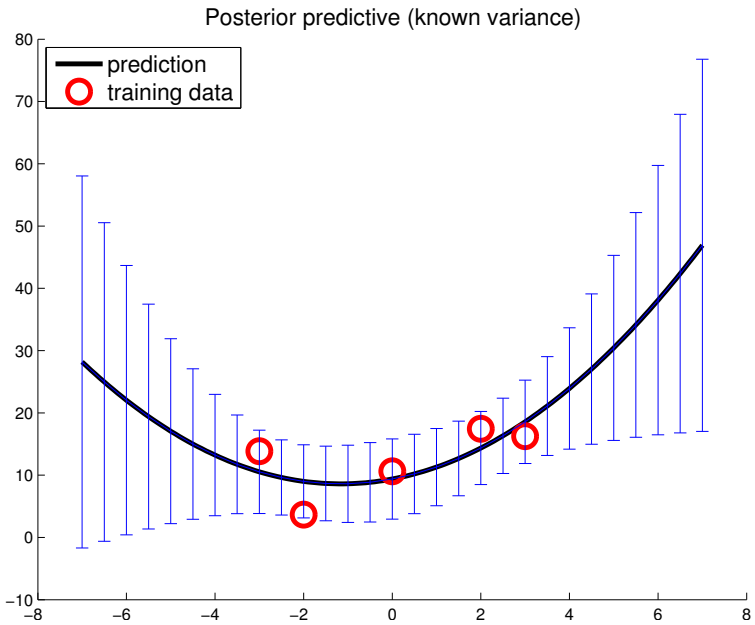
A point estimate can be obtained by minimizing a given loss function L:

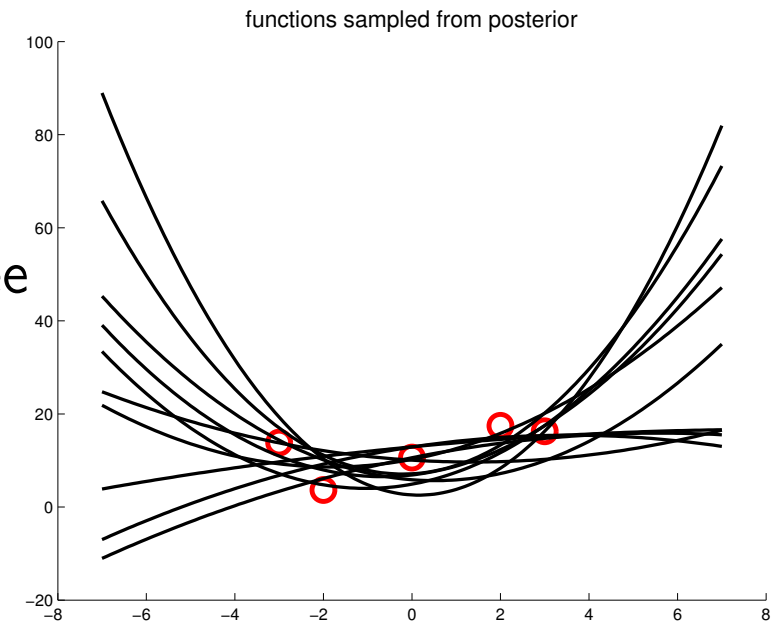
$$\arg\min_{y'} \int L(y', y_{new})p(y_{new}|x_{new}, X, y, \sigma^2, s^2)dy_{new}$$

If $L(y', y_{new}) = (y' - y_{new})^2$, the optimal prediction is the mean of the distribution $x_{new}^T \mu_w$.

Bayesian linear regression: predictive distribution







Using MLE and BLR to fit a degree 2 polynomial in one dimension

(Rogers and Girolami, 2012)

Link with ridge regression

We have found that the posterior mean (which is also the MAP estimate) is given by:

$$\mu_w = \left(\frac{1}{\sigma^2}X^TX + \frac{1}{s^2}I_p\right)^{-1}\frac{1}{\sigma^2}X^Ty = \left(X^TX + \frac{\sigma^2}{s^2}I_p\right)^{-1}X^Ty,$$

which is exactly the ridge regression solution when $\lambda = \frac{\sigma^2}{s^2}$.

This is not surprising as the MAP estimate minimizes exactly the ridge regression objective function.
Indeed, we have

$$\hat{w}_{MAP} = \arg\max_w p(w|X, y, \sigma^2, s^2) = \arg\max_w p(y|X, w, \sigma^2)p(w|s^2)$$

Since

$$p(y|X, w, \sigma^2) = \mathcal{N}(y|Xw, \sigma^2 I_n) = \prod_{i=1}^{N}\mathcal{N}(y_i|x_i^Tw, \sigma^2),$$

one can show that (by maximising the log of the posterior):

$$\hat{w}_{MAP} = \arg\max_w -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i^Tw - y_i)^2 - \frac{1}{2s^2}\sum_{i=1}^{p}w_i^2$$

Link with ridge regression

...and thus:

$$\hat{w}_{MAP} = \arg\min_{w} \sum_{i=1}^{N} (x_i^T w - y_i)^2 + \frac{\sigma^2}{s^2} ||w||_2^2.$$

Squared error in the objective function comes from the hypothesis of an iid Gaussian noise assumption and ℓ^2 penalisation comes from a Gaussian prior on w.

One can obtain different objective functions by changing the noise model and/or the prior.

For example, one can retrieve LASSO by using Laplace priors:

$$P(w|\lambda) = \prod_{j=1}^{p} \text{Lap}(w_j|0, 1/\lambda), \text{ with } \text{Lap}(x|\mu, b) = \frac{1}{2b} \exp(-\frac{|x-\mu|}{b})$$

(Posteriors can not be computed analytically anymore however)

Discussion

When using conjugate prior, it is possible to compute everything in closed-form, which makes the approach very efficient.

To a specific choice of model and prior corresponds a specific objective function for the MAP. Bayesian modeling is a way to design principled loss function and regularization terms.

With respect to the MAP, full Bayesian predictions are more uncertain for x values that are far away from training examples (as expected).

Outline

Bayesian logistic regression: likelihood

We observe a set of pairs $\mathcal{D} = \{(x_i, y_i)|i = 1, \ldots, N\}$, with $x_i \in \mathcal{R}^p$ and $y_i \in \{0, 1\}$ a binary classification response.

We assume that the class-conditional probability of belonging to the "1" class is given by a nonlinear transformation of a linear function of x:

$$P(y = 1|x, w) = \sigma(x^T w).$$

The most-commonly used function σ is the logistic function:

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

If we assume that predictions for different x are independent given w, then the **likelihood** can be written:

$$p(y|X, w) = \prod_{i=1}^{N} \left(\frac{1}{1 + \exp(-x_i^t w)} \right)^{y_i} \left(\frac{\exp(-x_i^t w)}{1 + \exp(-x_i^t w)} \right)^{1-y_i}$$

Bayesian logistic regression: prior and posterior

It seems reasonable to use the same **prior** as for linear regression, ie.:

$$p(w|s^2) = \mathcal{N}(w|0, s^2 I_p).$$

We want to compute the **posterior** distribution from the prior and the likelihood:

$$p(w|X, y, s^2) = \frac{p(y|X, w)p(w|s^2)}{\int p(y|X, w)p(w|s^2)dw} = \frac{p(y|X, w)p(w|s^2)}{p(y|X, s^2)},$$

so as to predict the response for new inputs x_{new}:

$$p(y_{new} = 1|x_{new}, X, y, s^2) = E_{p(w|X,y,s^2)}\left\{\frac{1}{1 + \exp(-x_{new}^T w)}\right\}$$

Unfortunately, the posterior distribution does not belong to a nice parametric family and the integral $p(y|X, s^2)$ is intractable as well.

How to solve intractability of posterior computation?

Three main approaches:

- Get a point estimate of w such as the MAP or the MLE
- Approximate $p(w|X, y, s^2)$ with some other density
- Obtain samples from the posterior $p(w|X, y, s^2)$

A point estimate: the MAP solution

The MAP estimate is defined as:

$$\begin{aligned}\hat{w}_{MAP} &= \arg\max_w p(w|X, y, s^2) = \arg\max_w p(y|X, w)p(w|s^2) \\ &= \arg\min_w \sum_{i=1}^{N} -y_n \log(\sigma(x_i^T w)) - (1 - y_n) \log(1 - \sigma(x_i^T w)) + \frac{1}{2s^2}||w||_2^2\end{aligned}$$

NB: our noise model leads to cross-entropy as the loss function.

Unlike in linear regression, $\hat{w}_{MAP}$ can not be computed analytically.

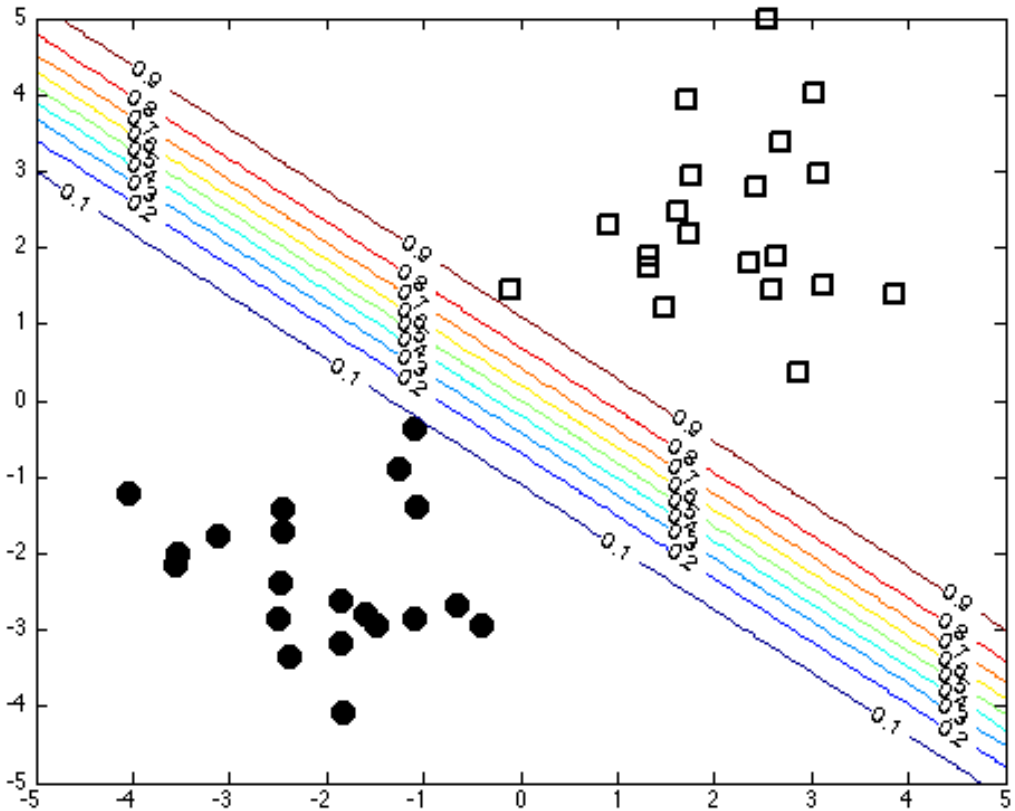
Optimization methods can however be used to compute it, e.g. stochastic gradient descent.

It can be shown that the objective function is **convex** and there is thus a unique global minimum.

Main disadvantage: we do not maintain a density over w, which is the main interest of Bayesian methods.

A point estimate: the MAP solution

$$p(y_{new} = 1 | x_{new}, \hat{w}_{MAP}) = \sigma(x_{new}^T \hat{w}_{MAP})$$



(Rogers and Girolami, 2012)

Laplace approximation

The idea of Laplace approximation is to approximate the posterior density with a Gaussian:

$$p(\theta|\mathcal{D}, m) \approx \mathcal{N}(\theta|\mu, \Sigma)$$

The Laplace approximation is based on a Taylor expansion of the negative log of the unnormalized posterior:

$$\Psi(\theta) = -\log p(\mathcal{D}|\theta, m) - \log p(\theta|m),$$

around its maximum $\hat{\theta}_{MAP} = \arg\max_\theta p(\theta|\mathcal{D}, m)$.

The second-order Taylor expansion of $\Psi(\theta)$ at $\hat{\theta}$ is given by:

$$\Psi(\theta) \approx \Psi(\hat{\theta}) + (\theta - \hat{\theta})^T g + \frac{1}{2}(\theta - \hat{\theta})^T H(\theta - \hat{\theta}),$$

where g and H are resp. the gradient and the Hessian of $\Psi(\theta)$ evaluated at $\hat{\theta}$:

$$g = \nabla\Psi(\theta)|_{\theta=\hat{\theta}}\,,\; H = \nabla\nabla\Psi(\theta)|_{\theta=\hat{\theta}}\,.$$

Laplace approximation

Since the gradient is zero at $\hat{\theta}_{MAP}$, we have:

$$\Psi(\theta) \approx \Psi(\hat{\theta}_{MAP}) + \frac{1}{2}(\theta - \hat{\theta}_{MAP})^T H(\theta - \hat{\theta}_{MAP}),$$

Negating and exponentiating both sides, one gets:

$$p(\theta|\mathcal{D}) \propto \exp(-\Psi(\hat{\theta}_{MAP})) \exp\left(-\frac{1}{2}(\theta - \hat{\theta}_{MAP})^T H(\theta - \hat{\theta}_{MAP})\right),$$

which is proportional to a Gaussian distribution:

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta|\hat{\theta}_{MAP}, H^{-1}).$$

Note that this approximation makes the computation of the marginal likelihood $P(\mathcal{D}|m)$ possible using the known normalization of the Gaussian.

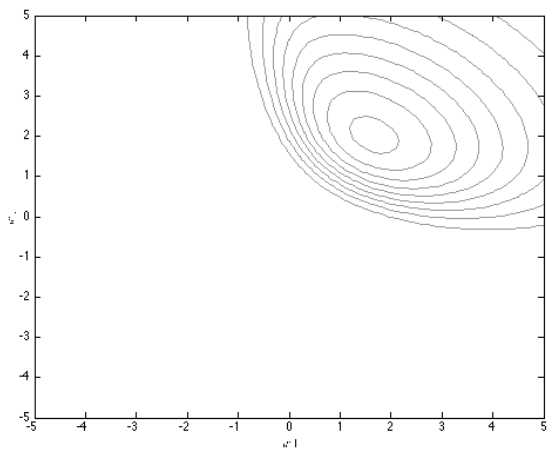
Laplace approximation for logistic regression: posterior

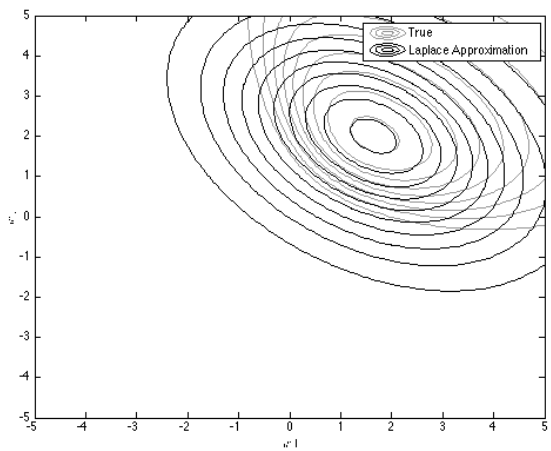
In the case of logistic regression, one can thus approximate the posterior as follows:

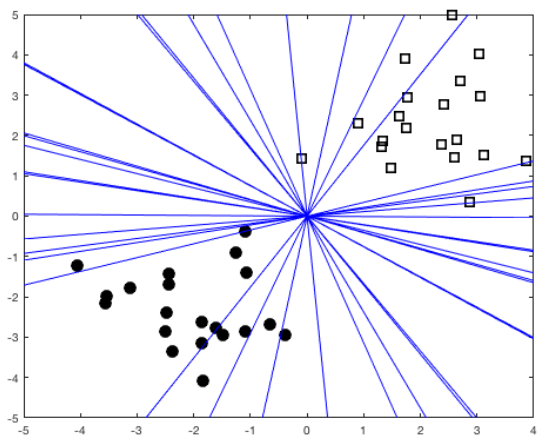
$$p(w|X, y, s^2) \approx \mathcal{N}(w|\hat{w}_{MAP}, H^{-1}),$$

where $\hat{w}_{MAP}$ can be obtained by optimization and H is given by:

$$H = \frac{1}{s^2} I_p + \sum_{i=1}^{N} x_i x_i^T P_i(1 - P_i), \text{ with } P_i = \sigma(x_i^T \hat{w}_{MAP})$$







(Rogers and Girolami, 2012)

Laplace approximation for logistic regression: prediction

To make a prediction, one needs the predictive distribution:

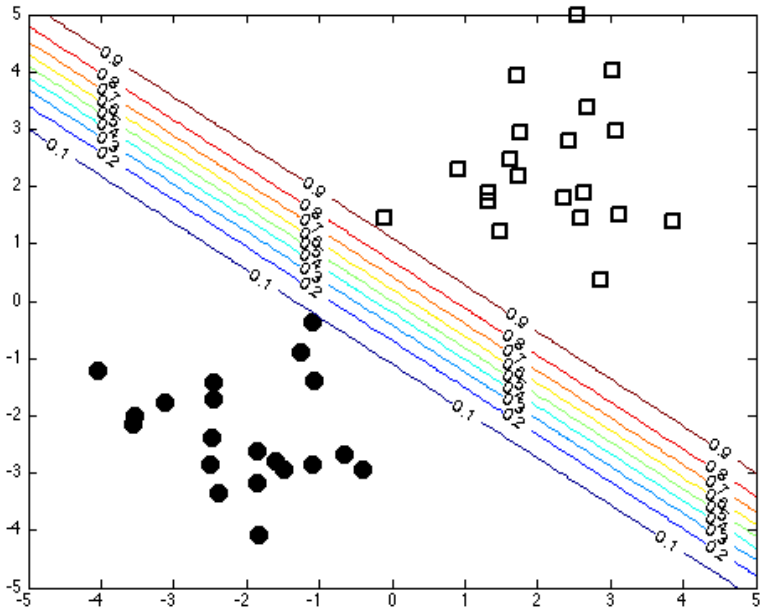
$$\begin{aligned} P(y_{new} = 1|x_{new}, X, y, s^2) &= \int P(y_{new} = 1|x_{new}, w)p(w|X, y, s^2)dw \\ &= E_{p(w|X,y,s^2)}\{P(y_{new} = 1|x_{new}, w)\}. \end{aligned}$$

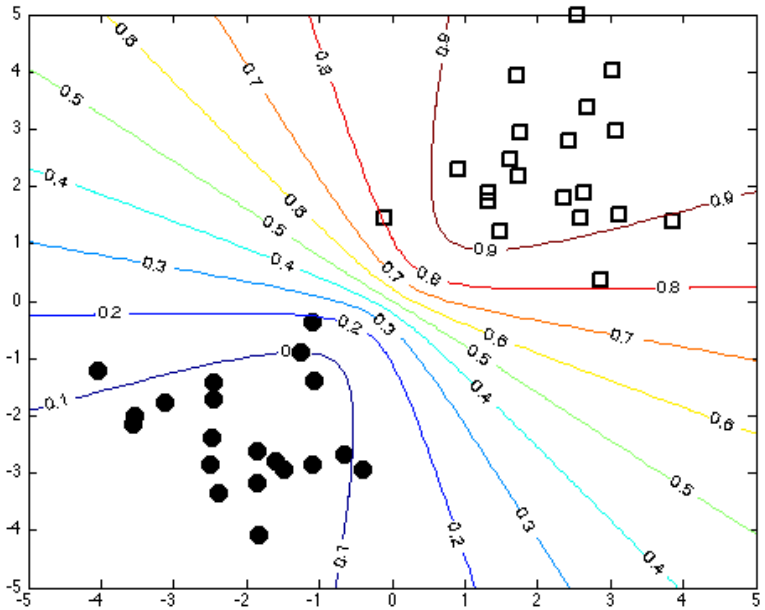
Unfortunately, this integral can not be computed analytically even with the Gaussian approximation of the posterior. A numerical approximation can however be easily obtained by Monte-Carlo sampling:

$$P(y_{new} = 1|x_{new}, X, y, s^2) \approx \frac{1}{S}\sum_{i=1}^{S} \sigma(x_{new}^T w^s),$$

where w^s are independently sampled from $\mathcal{N}(w|\hat{w}_{MAP}, H^{-1})$.

Laplace approximation for logistic regression: prediction





(Rogers and Girolami, 2012)

Predictive probability contours are now curved and better reflect expected uncertainty.

Sampling techniques

In previous examples, Laplace approximation allowed us to sample easily from the (approximate) posterior. There exist general techniques to sample directly from the posterior without requiring to approximate it first.

We will explain and illustrate, without proof, one of these techniques, **Metropolis-Hastings**, in the case of logistic regression.

Links to other techniques will be given later.

Metropolis-Hastings algorithm

Input: a target distribution $p(x)$ from which we would like to sample and a proposal distribution $q(x'|x)$.
Output: a sequence of random samples $\{x_0, x_1, x_2, \ldots\}$.

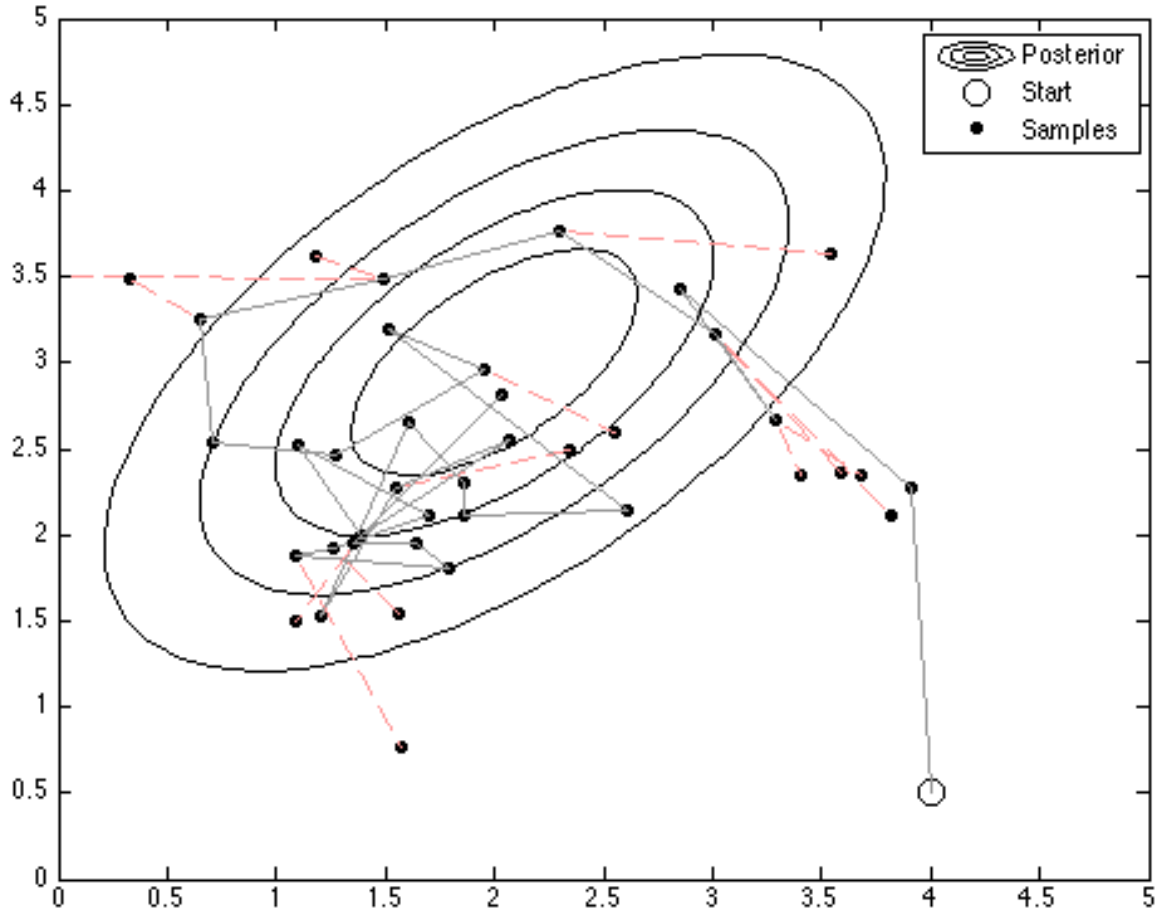
- Initialize x^0
- for $s = 0, 1, 2, \ldots$:
  1. Sample $x' \sim q(x'|x^s)$
  2. Compute acceptance probability

$$\alpha = min\left(1, \frac{p(x')q(x^s|x')}{p(x^s)q(x'|x^s)}\right)$$

  3. Sample $u \sim U(0, 1)$
  4. Set new sample to

$$x^{s+1} = \begin{cases} x' & \text{if } u < \alpha \\ x^s & \text{if } u \geq \alpha \end{cases}$$

Metropolis-Hastings algorithm: illustration



(Rogers and Girolami, 2012)

Metropolis-Hastings algorithm: some comments

- MH algorithm is an instance of Markov Chain Monte Carlo (MCMC) algorithms. It defines a Markov chain (new state x^{s+1} only depends on previous state x^s), which stationary distribution coincides with $p(x)$.
- One should wait before collecting, to let the chain reach convergence (waiting time is called burn-in period). There are tools to diagnose whether convergence has occurred or not.
- The proposal distribution $q(x'|x)$ has an influence on convergence/acceptance rate and should be chosen wisely.
- If one takes a symetric proposal distribution, ie. $q(x|x') = q(x'|x)$, then the acceptance rate only depends on target density $p(x)$.
- The algorithm only requires to be able to compute ratios $\frac{p(x')}{p(x^s)}$. This will be very convenient in our setting.

MH with logistic regression

The target density is our posterior $p(w|X, y, s^2)$. We can not compute $p(w|X, y, s^2)$ because the denominator is unknown. We can however easily compute ratios:

$$\frac{p(w_1|X, y, s^2)}{p(w_2|X, y, s^2)} = \frac{p(y|X, w_1)p(w_1|s^2)}{p(y|X, w_2)p(w_2|s^2)}$$

We can use a Gaussian as a proposal density:

$$q(w|w') = \mathcal{N}(w'|w, \gamma^2 I_p),$$

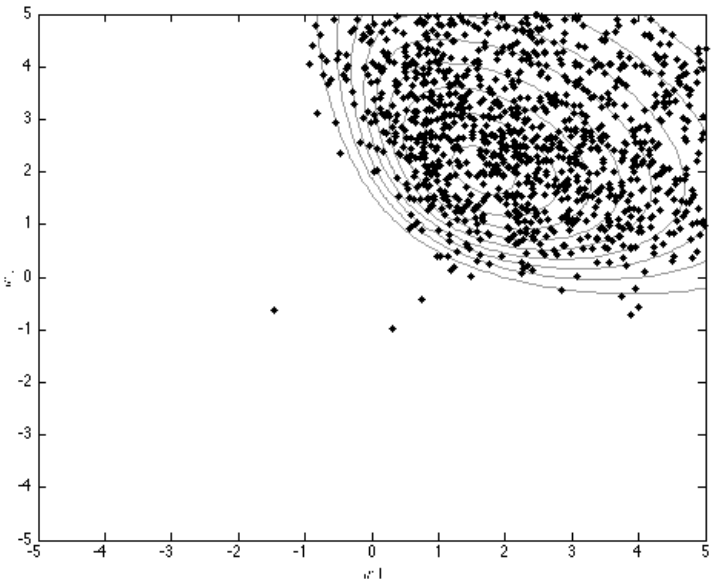
with γ a user-defined parameter. The later being symmetric, the new sample will be accepted as soon as its posterior is higher than that of the old sample.

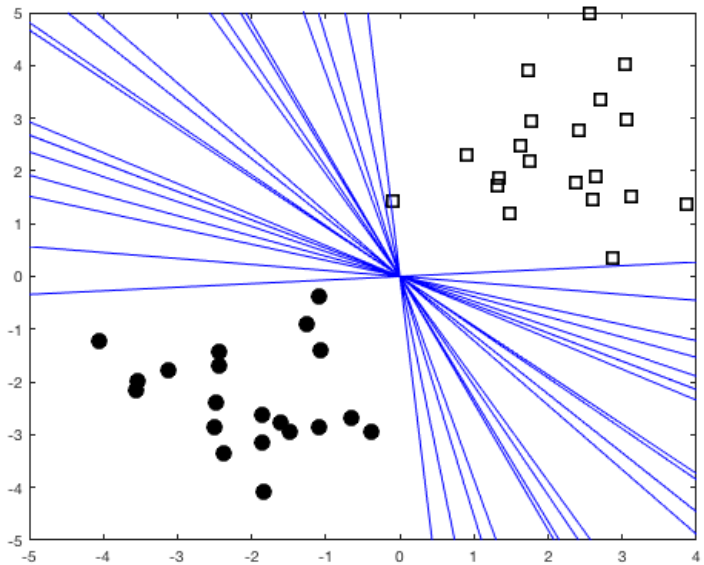
Predictions can be obtained by computing the following average:

$$P(y_{new} = 1|x_{new}, X, y, s^2) \approx \frac{1}{S} \sum_{i=1}^{S} \sigma(x_{new}^T w^s),$$

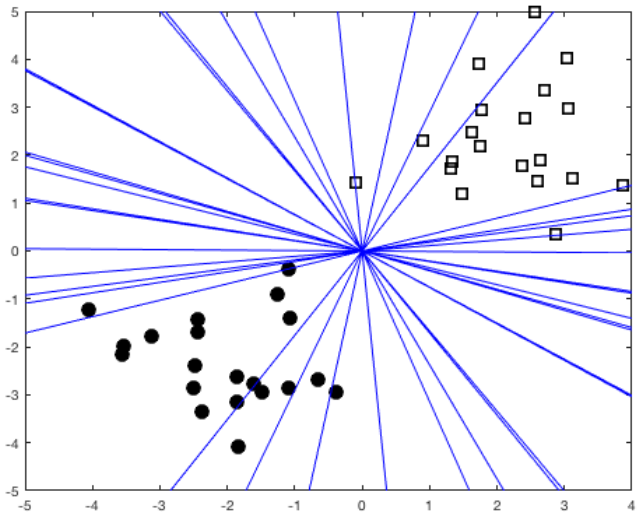
where w^s are samples generated by MH algorithm.

MH with logistic regression: sampling



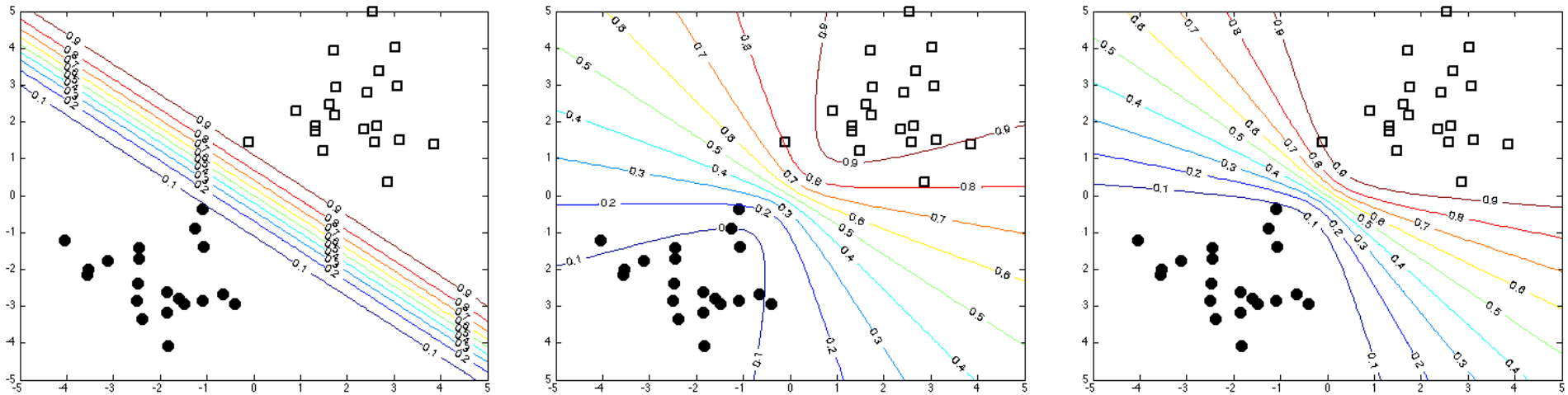


To be compared with samples from Laplace approximation:



(Rogers and Girolami, 2012)

MH with logistic regression: predictions



(Rogers and Girolami, 2012)

From left to right: MAP, Laplace approximation, Metropolis-Hastings.

Discussion

Having no analytical solution should not stop us using Bayesian approaches.

There are many solutions to approximate the posterior and the marginal likelihood when conjugate prior distributions can not be used, among which:

- Laplace approximation
- Markov Chain Monte Carlo methods
- Variational approximations
- Bayesian Information Criterion (BIC)
- Expectation Propagation (EP)
- Sequential Monte Carlo methods
- ...

See Gilles' lecture on differentiable inference later in this course and MATH2022 (Yvik Swan) next year for a more mathematical treatment of some of these methods

Outline

Bayesian model selection

Let us assume that we have L models at our disposal: $\{m_1, m_2, \ldots, m_L\}$, each with its own parameters. How do we select between them using the data $\mathcal{D}$, in a bayesian way?

We can compare models on the basis of the following **model posteriors**:

$$P(m_i|\mathcal{D}) = \frac{p(\mathcal{D}|m_i)P(m_i)}{\sum_j p(\mathcal{D}|m_j)P(m_j)},$$

where $p(m_i)$ are user-defined model priors and

$$p(\mathcal{D}|m_i) = \int p(\mathcal{D}|\theta_i, m_i)p(\theta_i|m_i)d\theta_i$$

is the marginal likelihood or, in the context of model selection, the **model evidence**.

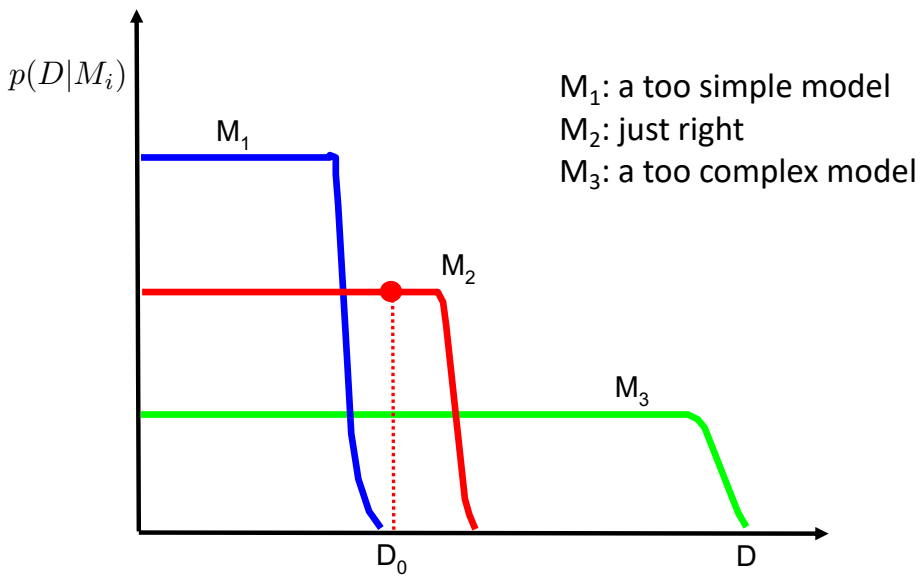
If all models are a priori equally likely, then model evidence expresses the preference shown by the data for different models. It measures the probability of generating the dataset from a model whose parameters are sampled at random from the prior.

The ratio of two model evidences is known as a **Bayes factor**: $\frac{P(\mathcal{D}|m_1)}{P(\mathcal{D}|m_2)}$.

Bayesian Occam's razor effect

If two models can both explain the data equally well (in terms of likelihood), then $P(\mathcal{D}|m)$ has an intrinsic bias towards the simplest one among them. This is called the Bayesian Occam's razor effect.

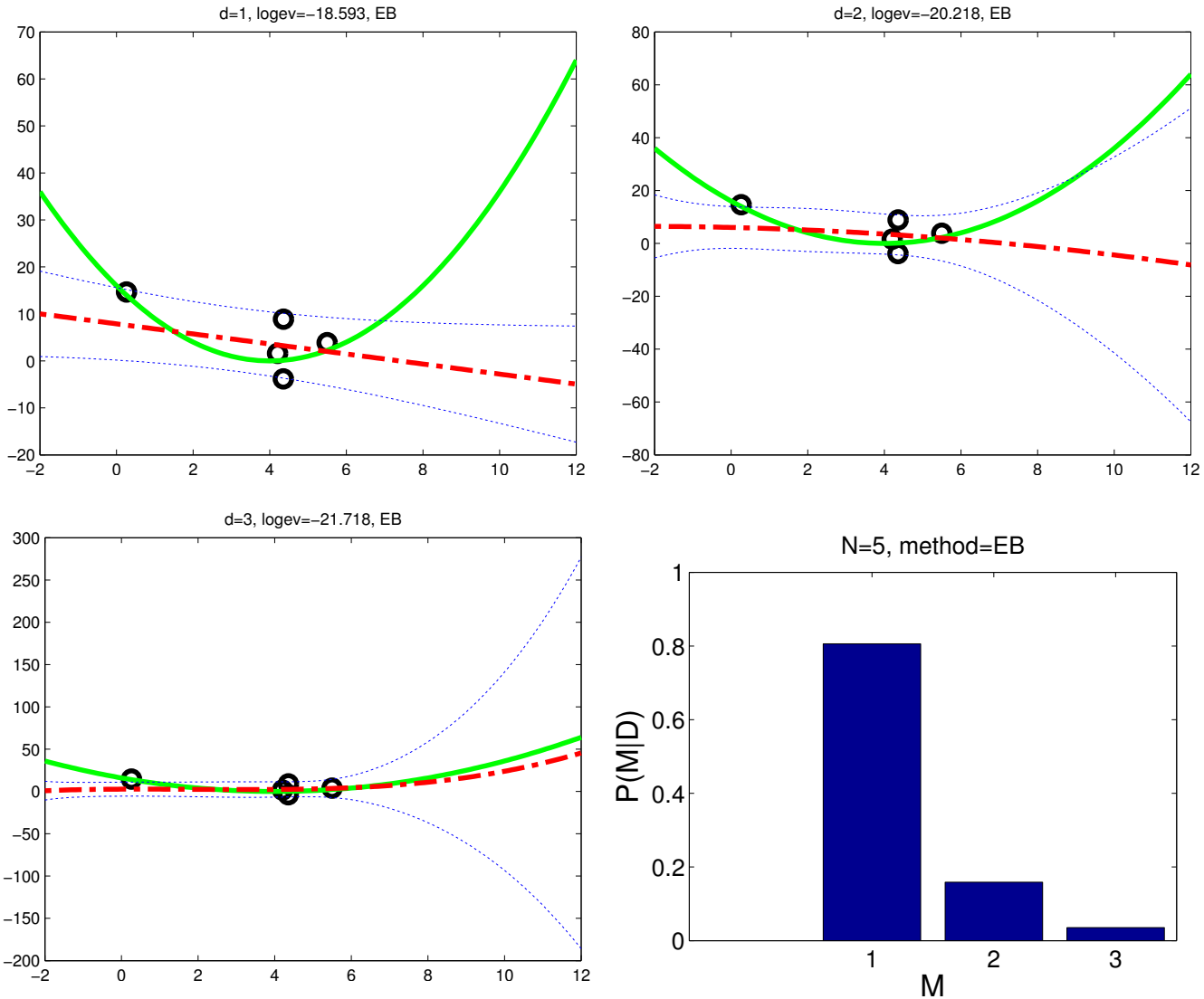
Since $\sum_{\mathcal{D}'} p(\mathcal{D}'|m) = 1$, a complex model being able to fit more datasets will attribute less probabilistic weight to each of them. A simple model, on the other hand, will be able to fit less datasets but therefore will put more weight to each of them.



(Murphy, 2012)

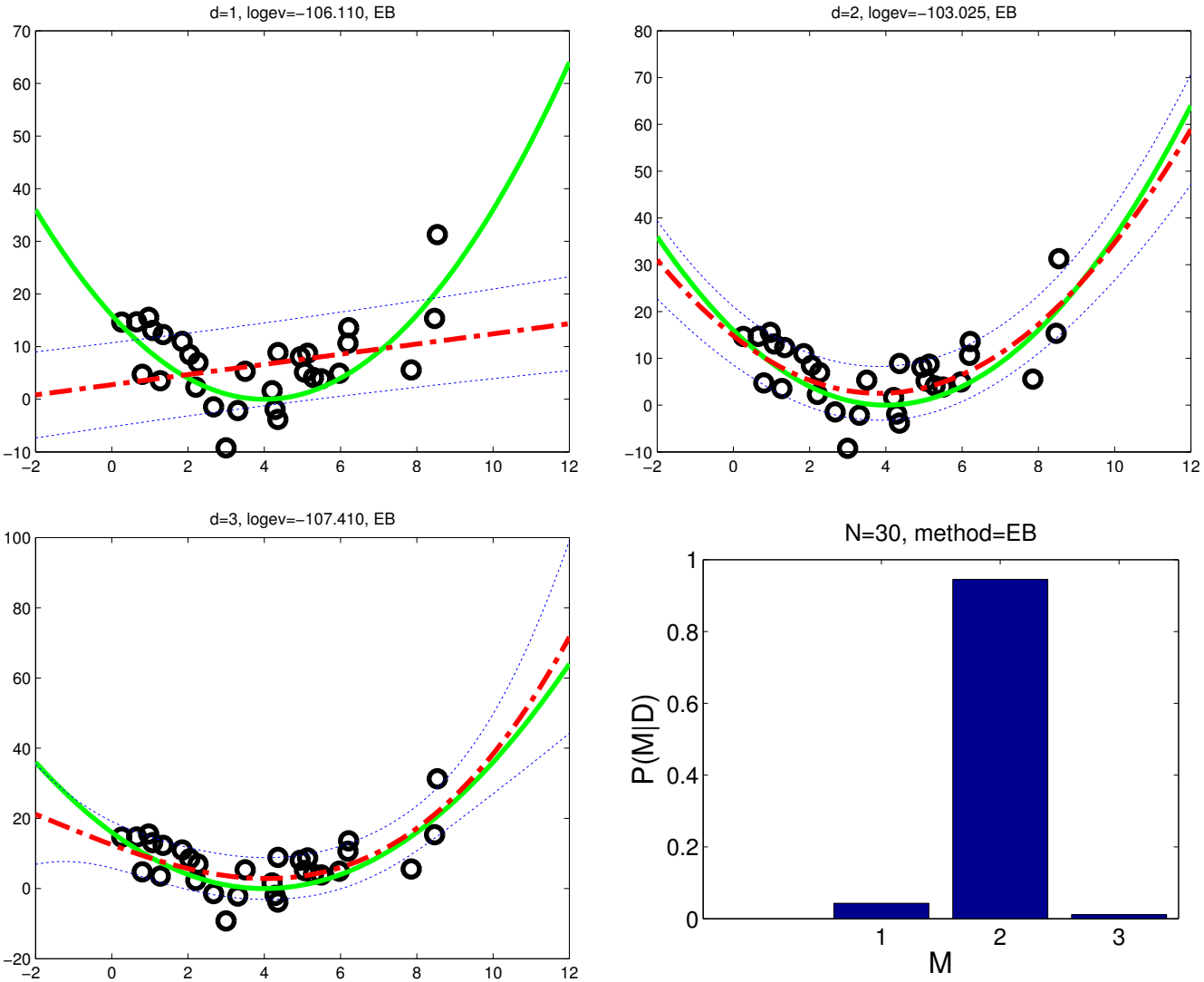
Note however that $P(\mathcal{D}|\hat{\theta}_{ML}, m_i)$ would be biased towards more complex models.

Bayesian Occam's razor effect: illustration



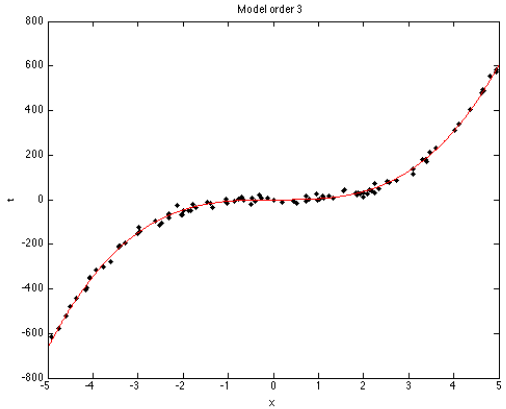
(Murphy, 2012)

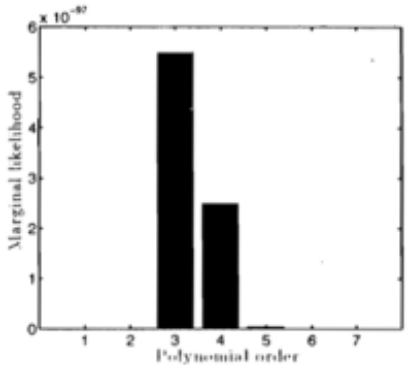
Bayesian Occam's razor effect: illustration



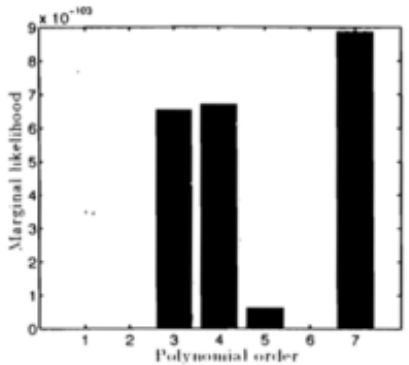
(Murphy, 2012)

Bayesian Occam's razor effect: impact of the prior

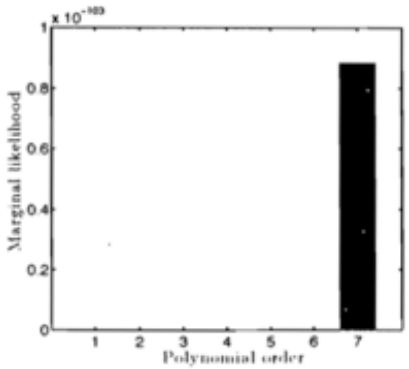




$s^2 = 0.7$



$s^2 = 0.4$



$s^2 = 0.3$

Bayesian LR with different polynomial order (from 1 to 7). The true function is $y = 5x^3 - x^2 + x$. Three prior variances are considered: $s^2 = \{0.7, 0.4, 0.3\}$. Why does the prior affect model selection?

(Rogers and Girolami, 2012)

Bayesian model averaging

Bayesian model averaging is an alternative to model selection. The predictive distribution is given by:

$$p(y_{new}|x_{new}, X, y) = \sum_{i=1}^{L} p(y_{new}|x_{new}, X, y, m_i)p(m_i|X, y)$$

The overall predictive distribution is obtained by averaging the predictive distributions of individual models, weighted by the posterior probabilities.

Note that if two models are combined with peaky distributions, the resulting predictive distribution will end up being multi-modal.

Model selection is still often preferred because it decreases complexity.

Outline

Advantages and limitations of Bayesian approaches

Advantages:

- They provide a very coherent and principled framework
- They are conceptually straightforward
- They are modular
- Often good performance (averaging effect, robust against overfitting)
- They increase model interpretability
- Adding uncertainties to predictions is crucial in many IA applications (e.g., autonomous driving)

Limitations:

- They are subjective
- It is hard to come up with a prior. Usually, our assumptions are wrong
- The closed world assumption: need to consider all possible hypotheses before observing the data
- They can be computationally demanding
- The use of approximations weakens the coherence argument

Other related topics

- Applications in unsupervised learning. Eg., gaussian mixture models, latent dirichlet allocation.
- Debate between frequentists and bayesians.
- Techniques to define priors.
- Analysis of the theoretical properties of these methods. There exist generalization error bounds for Bayesian methods (PAC-Bayes)
- Bayesian non-parametric methods: how to deal with models with an infinite number of parameters. Eg., gaussian processes (see next week), dirichlet processes.
- Advanced inference methods. Eg., ADVI.
- Bayesian optimization
- Bayesian additive regression trees
- **Bayesian deep learning**
- **Probabilistic programming**

Bayesian deep learning

Deep learning methods are prominent in ML these days but these methods suffer from several limitations:

- Often crucially rely on large data sets
- Uninterpretable black-boxes: it is not clear what a model does not know
- Easily fooled (AI safety)
- Lacks solid mathematical foundations (mostly ad hoc)
- How to best regularize/train them?

Bayesian methods are good candidates to address these limitations and are explored extensively in the DL community these days.

http://bayesiandeeplearning.org
Panel discussion: "Is bayesian deep learning the most brilliant thing ever?"
https://www.youtube.com/watch?v=HumFmLu3CJ8

Bayesian methods for training neural networks

Bayesian methods for training neural networks have been proposed starting in the 90's.

The approach is very similar to bayesian linear regression or bayesian logistic regression:

- Define a prior on the network weights, eg., $W \sim \mathcal{N}(0, I)$
- Assume some likelihood model, eg., $p(y|x, W) = \mathcal{N}(y|f(x; W), \sigma^2 I)$
- Compute the posterior $p(W|\mathcal{D})$ and use it to analyse the model or make predictions through averaging

Main problem is of course the evaluation of the posterior, when f is a very complex non-linear function of the parameters.

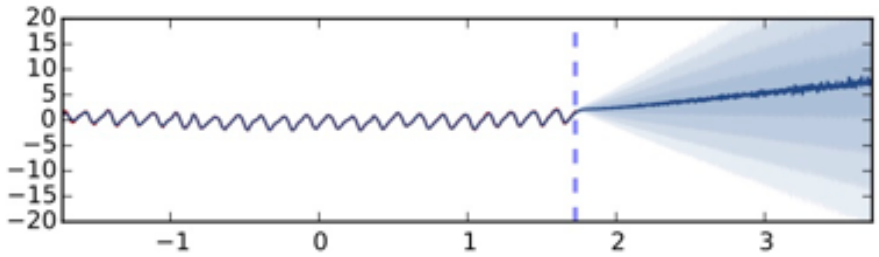
Bayesian methods for training neural networks

Several approximate inference methods have been proposed in the literature:

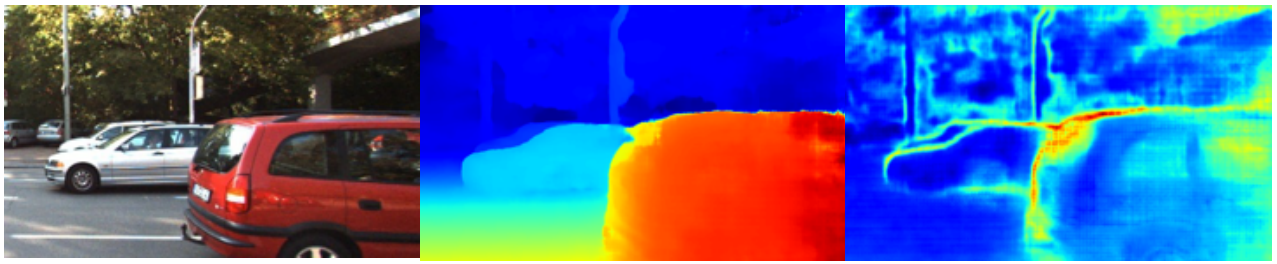
- Laplace (Gaussian) approximation (MacKay, 1991),
- MCMC techniques (HMC, Neal, 1993),
- Variational inference (Hinton and Van Camp, 1993, Barber and Bishop, 1998),
- Combination of SGD and MCMC (Welling and Whye Teh, 2011),
- Probabilistic adaptation of back-propagation (Hernandez-Lobato and Adams, 2015),
- Modified drop-out (Gal and Ghahramani, 2017)
- ...

See "Uncertainty in deep learning", Yarin Gal, Phd thesis, 2016 for a review of these works.

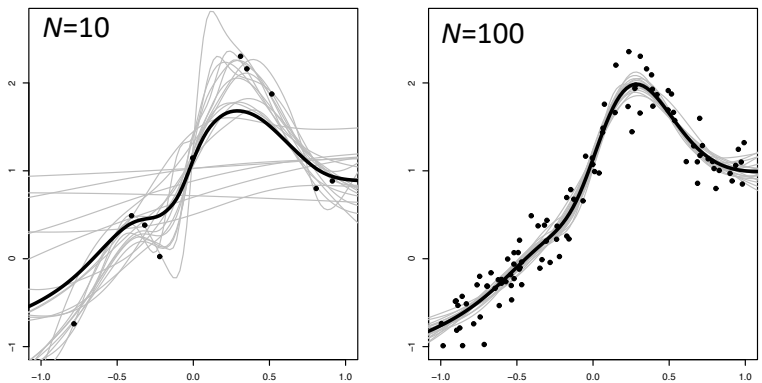
Bayesian deep learning: what do we gain?



(Gal and Ghahramani, 2016)



(Kendall et al., 2017)



Bayesian NN, 1 layer, 100 hidden units (Neal, 2004)

TABLE 11.3. *Performance of different methods. Values are average rank of test error across the five problems (low is good), and mean computation time and standard error of the mean, in minutes.*

Method	Screened Features		ARD Reduced Features	
	Average Rank	Average Time	Average Rank	Average Time
Bayesian neural networks	1.5	384(138)	1.6	600(186)
Boosted trees	3.4	3.03(2.5)	4.0	34.1(32.4)
Boosted neural networks	3.8	9.4(8.6)	2.2	35.6(33.5)
Random forests	2.7	1.9(1.7)	3.2	11.2(9.3)
Bagged neural networks	3.6	3.5(1.1)	4.0	6.4(4.4)

Comparison of Bayesian NN and other ensemble methods on a feature selection challenge (Section 11.9, Hastie et al., 2013)

Probabilistic programming

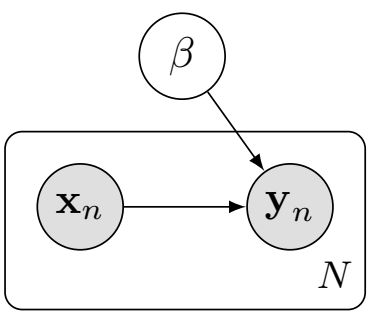
Probabilistic programming languages are programming languages designed to describe probabilistic models and then perform inference in those models. They unify standard programming language with probabilistic modeling.

These tools become more and more mature, flexible, and computationally efficient. They implement the most efficient inference tools.

By automating inference, these languages allow to focus on the most interesting part of applying Bayesian methods: the modelling.

Examples of popular languages: Stan, Pyro, Edward, PyMC3, etc.

Bayesian logistic regression in Edward (http://edwardlib.org)




```
# Model
x = tf.Variable(x_data, trainable=False)
beta = Normal(mu=tf.zeros(D), sigma=tf.ones(D))
y = Bernoulli(logits=tf.dot(x, beta))

# Inference
qbeta = Empirical(params=tf.Variable(tf.zeros([T, D])))
inference = ed.HMC({beta: qbeta}, data={y: y_data})
inference.run(step_size=0.5 / N, n_steps=10)
```

References

Bayesian approaches in textbook:

- A first course in machine learning, Rogers and Girolami, CRC, 2012
  Chapter 3 and 4
- Machine learning: a probabilistic perspective, Murphy, MIT Press, 2012
  Chapter 5, 7.6
- Pattern recognition and machine learning, Bishop, Springer, 2006
  Sections 3.3 to 3.5

Other interesting references:

- Mackay, D. (1995) Probable Networks and Plausible Predictions - A Review of Practical Bayesian Methods for Supervised Neural Networks, David Mackay, Network: Computation in Neural Systems Vol. 6, Iss. 3.
- Ghahramani, Z. (2015) Probabilistic machine learning and artificial intelligence. Nature 521:452-459.
- Course notes of Roman Garnett:
  http://www.cse.wustl.edu/~garnett/cse515t/spring_2017/

Quizz (1)

- Explain the main idea of the Bayesian approach and contrast it with maximum likelihood and maximum a posteriori learning.
- Explain the quote in slide 14.
- Explain what is shown and observed in the plots of slides 22 to 24.
- Let us consider Bayesian linear regression in the same one-dimensional setting as in Slide 22. Assuming $\sigma^2 = s^2 = 1$ and a training sample composed of a single training point $(x, y) = (1, 0.2)$, express the mean and the variance of the predictive distribution as a function of x_{new}. Where is the variance the smaller? Explain what you find.
- Show that using the Laplace priors, the MAP estimate is equivalent to LASSO.
- What is the Laplace approximation of the Bayesian linear regression posterior (using a Gaussian prior)?

Quizz (2)

- (Ex 3.11, Bishop) For Bayesian linear regression, show that the uncertainty of the prediction at any point x_{new} decreases as the learning sample size increases.
  Suggestion: Let us denote by $\sigma_N(x_{new})$ the variance of the prediction for a learning sample of size N (see Slide 25 for its expression). Show that $\sigma_{N+1}(x) \leq \sigma_N(x)$, by making use of the following matrix identity (with $M \in \mathcal{R}^{m \times m}$ and $v \in \mathcal{R}^m$):

$$(M + vv^T)^{-1} = M^{-1} - \frac{(M^{-1}v)(v^T M^{-1})}{1 + v^T M^{-1} v}.$$