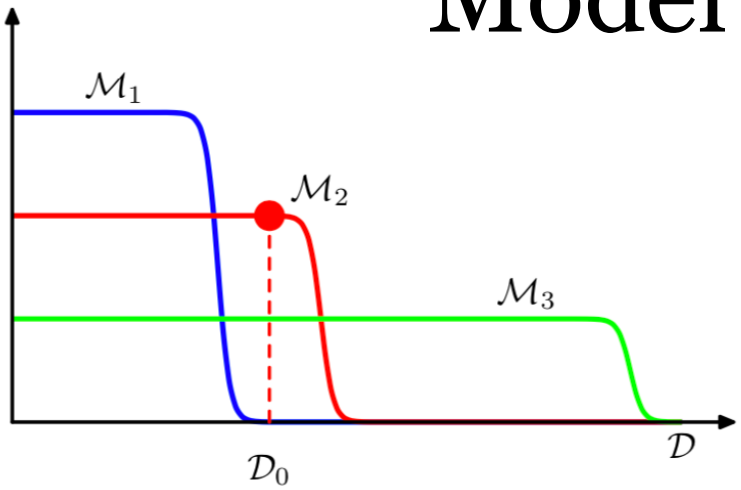
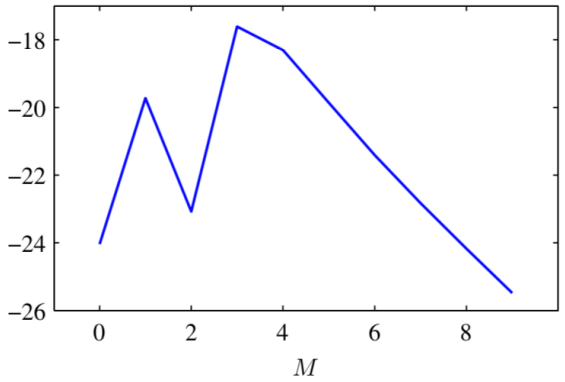
Bayesian Linear Models:
Model Selection





SPR Day 09, Spring 2020

Prof. Mike Hughes

https://www.cs.tufts.edu/comp/136/2020s/

Recap: Bayesian Linear Regression

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

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

Recap: Bayesian Linear Regression

General prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Simpler prior: assume zero mean, Just need to define precision

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

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

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}\right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}. \end{aligned}$$

Problem: Hyperparameter Selection

How do we pick the prior hyperparameters?

alpha

How do we pick the likelihood hyperparameters?

beta

Hyperparameter Selection

	Fixed valid. set (fraction f)	K-fold cross-validation	Bayesian evidence	
Fraction data used for training run	(1.0 – f)	(K-1) / K	**1.0**	**Higher is better** Better use of training data
Total runs/ examples seen for training	**1 run** **(1 – f) N**	K runs (K-1) * N	**1 run** **N**	**Lower is better** Faster training
Total runs/ examples seen for evaluation of fitness	1 run fN	K runs N	**1 run** **N**	**Lower is better** Faster evaluation
Fitness function	Heldout likelihood	Heldout likelihood	Evidence	

Why use Model Evidence?

3.4. Bayesian Model Comparison

As we shall see, the over-fitting associated with maximum likelihood can be avoided by marginalizing (summing or integrating) over the model parameters instead of making point estimates of their values. Models can then be compared directly on the training data, without the need for a validation set. This allows all available data to be used for training and avoids the multiple training runs for each model associated with cross-validation.

Related Problem: Model Selection

How do we pick the feature transform?

$$\phi(x_n) = [\phi_1(x_n) \quad \phi_2(x_n) \quad \ldots \quad \phi_M(x_N)]$$

What size M?
Which functions to include?

Challenge: Could have unbounded number of choices!

Need to enumerate **small set** of possible models (size L) if want to average over them properly

Model Selection for Regression

$\mathcal{M}_i$ Model family (size M, feature functions, hyperparameters)

$\mathbf{w}$ Specific parameters for chosen model

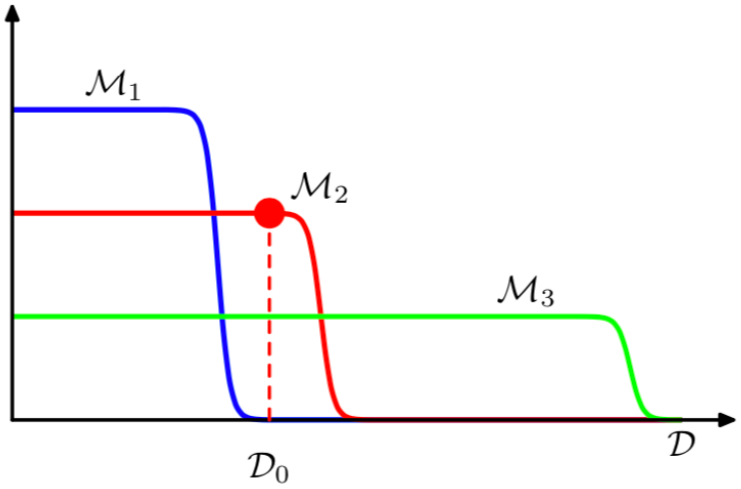
$t_n|x_n$ Observed outputs (and corresponding inputs)

$$t_n|x_n \sim \mathcal{N}(w^T\phi(x_n), \beta^{-1})$$

Model Selection via Evidence

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i) p(\mathbf{w}|\mathcal{M}_i)\, \mathrm{d}\mathbf{w}.$$

Figure 3.13 Schematic illustration of the distribution of data sets for three models of different complexity, in which $\mathcal{M}_1$ is the simplest and $\mathcal{M}_3$ is the most complex. Note that the distributions are normalized. In this example, for the particular observed data set $\mathcal{D}_0$, the model $\mathcal{M}_2$ with intermediate complexity has the largest evidence.



Key idea: "Goldilocks" principle

If model too simple, it puts high mass only few datasets

If model too complex, it puts mass on too many datasets

Predictive distribution

Average over predictive distribution for each of
L possible models, weighted by posterior probability

$$p(t|\mathbf{x}, \mathcal{D}) = \sum_{i=1}^{L} p(t|\mathbf{x}, \mathcal{M}_i, \mathcal{D}) p(\mathcal{M}_i|\mathcal{D})$$

Predictive distrib. for t given model i

Posterior of model i

Key idea: We can use all L models, don't need to pick one

Ideal predictive posterior

If we want to predict new data given old data, ideally we would **average over** all parameters w, alpha, beta, weighted by posterior prob.

$$p(t|\mathbf{t}) = \iiint p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) p(\alpha, \beta|\mathbf{t}) \, \mathrm{d}\mathbf{w} \, \mathrm{d}\alpha \, \mathrm{d}\beta$$

But, this integral is hard!

Tractable predictive posterior

Assume:

- We have enough data that the posterior p(alpha, beta | data) is peaked at MAP point estimate

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} \quad p(\alpha, \beta|\mathbf{t})$$

- Prior on alpha, beta is relatively uniform ("flat"), so these estimates might as well be ML estimates

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} \quad p(\mathbf{t}|\alpha, \beta)\text{flat}(\alpha, \beta)$$

constant

Then the **tractable estimate** of predictive posterior becomes:

$$p(t|\mathbf{t}) \simeq p(t|\mathbf{t}, \widehat{\alpha}, \widehat{\beta}) = \int p(t|\mathbf{w}, \widehat{\beta}) p(\mathbf{w}|\mathbf{t}, \widehat{\alpha}, \widehat{\beta}) \, \mathrm{d}\mathbf{w}$$

Now, we wish to solve this hyperparameter estimation

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} \quad p(\mathbf{t}|\alpha, \beta)$$

"Evidence"

But first, what is this "evidence" anyway?

Evidence for Linear Regression

- Probability of training data ***t*** given alpha, beta
- Marginalizes over the weights ***w***

$$p(\mathbf{t}|\alpha,\beta) = \int p(\mathbf{t}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)\,\mathrm{d}\mathbf{w}.$$

Simplifying the Evidence

$$p(\mathbf{t}|\alpha,\beta) = \int p(\mathbf{t}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)\,\mathrm{d}\mathbf{w}.$$

$$p(\mathbf{t}|\alpha,\beta) = \left(\frac{\beta}{2\pi}\right)^{N/2}\left(\frac{\alpha}{2\pi}\right)^{M/2}\int \exp\{-E(\mathbf{w})\}\,\mathrm{d}\mathbf{w}$$

$$E(\mathbf{w}) = \frac{\beta}{2}\|\mathbf{t}-\mathbf{\Phi}\mathbf{w}\|^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}.$$

Key ideas:

- Bring constants outside integral
- Recognize inside integral as Gaussian, “complete the square”

Closed-form Log Evidence

$$p(\mathbf{t}|\alpha,\beta) = \int p(\mathbf{t}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)\,\mathrm{d}\mathbf{w}.$$

$$\ln p(\mathbf{t}|\alpha,\beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) - \frac{1}{2}\ln|\mathbf{A}| - \frac{N}{2}\ln(2\pi)$$

$$\mathbf{A} = \alpha\mathbf{I} + \beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$$

Precision matrix for posterior p(w | data)

$$\mathbf{m}_N = \beta\mathbf{A}^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}.$$

Mean vector for posterior p(w | data)

$$E(\mathbf{m}_N) = \frac{\beta}{2}\|\mathbf{t} - \mathbf{\Phi}\mathbf{m}_N\|^2 + \frac{\alpha}{2}\mathbf{m}_N^{\mathrm{T}}\mathbf{m}_N$$

How to estimate?

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} \quad p(\mathbf{t}|\alpha, \beta)$$

- Can do gradient descent
- Can do coordinate descent (EM, later in course)
- Can get estimates analytically
  - See textbook!

$$\alpha = \frac{\gamma}{\mathbf{m}_N^{\mathrm{T}}\mathbf{m}_N} \qquad \frac{1}{\beta} = \frac{1}{N-\gamma}\sum_{n=1}^{N}\left\{t_n - \mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2$$

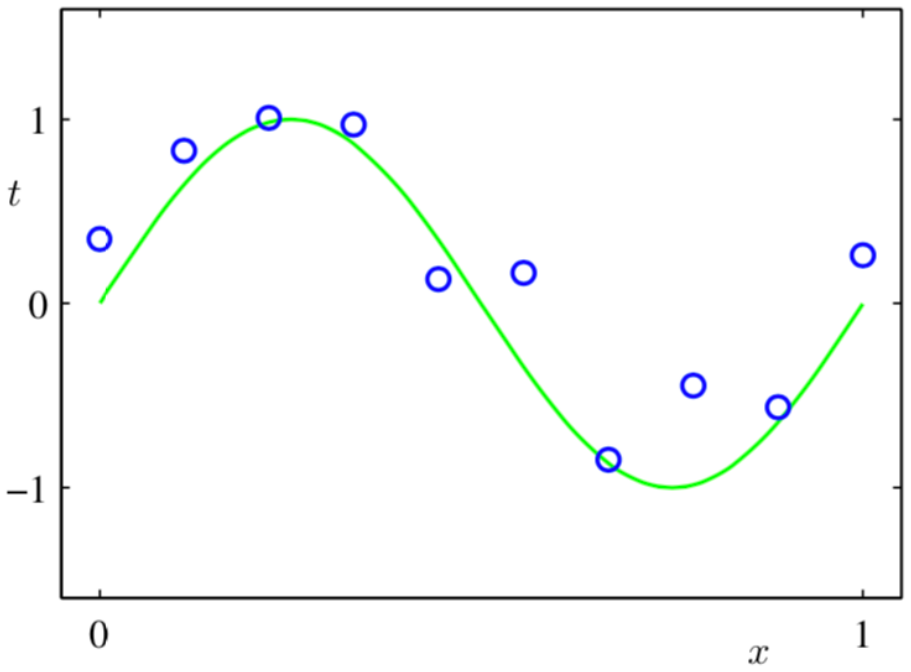
Cycle these until convergence!

$$\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i}. \qquad \left(\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)\mathbf{u}_i = \lambda_i\mathbf{u}_i$$

Eigendecomposition!

Example: 1D sinusoid data

Figure 1.2 Plot of a training data set of $N = 10$ points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable t. The green curve shows the function $\sin(2\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x, without knowledge of the green curve.



Model Selection for Linear Regr.
(using polynomial features of order M)

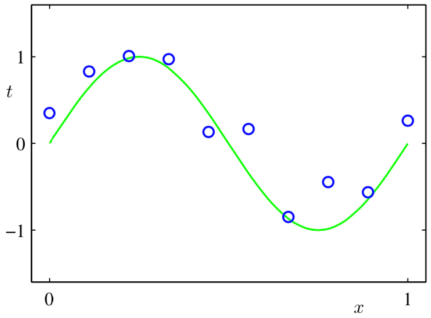
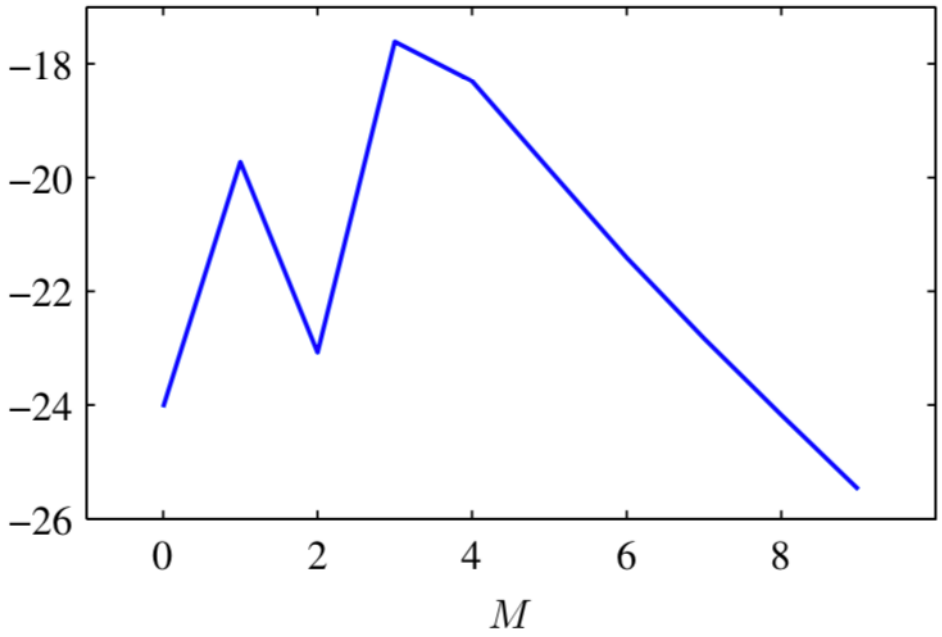


Figure 3.14 Plot of the model evidence versus the order M, for the polynomial regression model, showing that the evidence favours the model with $M = 3$.



Why does M=2 have low evidence?

Can set quadratic term to 0, but then we have a model that is "too complex"

Can set quadratic term non zero, but sinusoid (odd function) not a good fit by quadratic