Fundamentals of Machine Learning:

Bayesian Linear Regression

Prof. Andrew D. Bagdanov (andrew.bagdanov AT unifi.it)

UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

Outline

Introduction

Frequentist versus Bayesian

- In the previous lecture we saw how geometric interpretation of model error can be used to derive an intuitive linear regressor.
- We also saw how to ”beef up” our representation using an explicit embedding of inputs into a nonlinear space.
- We then added a probabilistic veneer to our model to derive a Maximum Likelihood Estimate of the ”best” model parameters.
- This frequentist inference model has many advantages and disadvantages.
- Today we will derive a fully Bayesian interpretation of inference and look at its advantages.
- First, we take a look at an important conceptual tool for understanding the relationship between bias and variance in models.

Lecture objectives

At the end of this lecture you will:

- Understand the relationship between bias and variance in models.
- Understand the tradeoff between bias and variance.
- Recognize the difference between point estimates like Maximum Likelihood and the full Bayesian treatment of linear regression.
- Understand how Bayesian inference allows us to incorporate (requires, actually) prior information about model parameters.
- Understand how Bayesian inference allows us to both quantify and update our belief in model predictions.

The Bias-Variance Decomposition

Regularization. What is it good for?

- This discussion of regularizing solutions to least-squares problems leads us naturally to an important conceptual tool in Machine Learning.
- Up to now we have, somewhat implicitly, assumed that the basis functions are somehow fixed in form and number.
- We have already seen hints of the problem of overfitting: if we use a model that is somehow too complex, we can drive down training error as much as we like.
- Regularization can control complexity, but that still leaves many questions:
  - What should λ be?
  - What should my base model (before regularization) be?
- Let's (loosely) develop some theory to help us analyze this issue.

A theoretical optimal

- For squared loss functions we can show that the optimal predictor is given by:

$$h(\mathbf{x}) = \mathbb{E}[t \mid \mathbf{x}] = \int tp(t \mid \mathbf{x})dt$$

- Which is just the conditional expectation of t given $\mathbf{x}$.
- Note that we haven't done anything with this result: the point of Machine Learning, in some sense, is to estimate this $p(t \mid \mathbf{x})$.
- That is, we want to do something like find a y that minimizes this error:

$$\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2$$

- This expresses the loss incurred for a single input $\mathbf{x}$ when using estimate $y(\mathbf{x}; \mathcal{D})$.

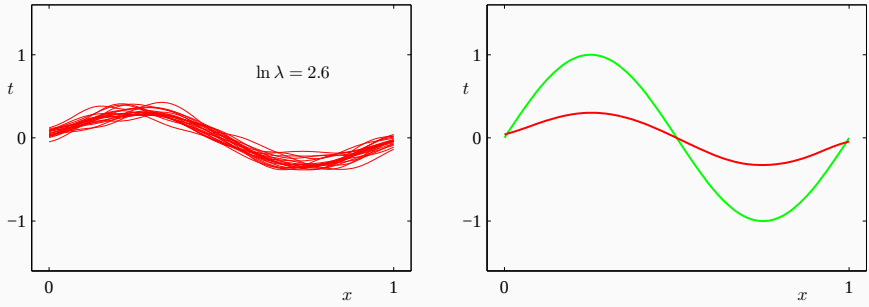
Bias, variance, and irreducible noise

- Taking expectation wrt $\mathcal{D}$ and considering all possible inputs, we (eventually) arrive at something that looks like:

$$
\begin{aligned}
\text{expected loss} &= (\text{bias})^2 + \text{variance} + \text{noise, where} \\
(\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x}) d\mathbf{x} \\
\text{variance} &= \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x}) d\mathbf{x} \\
\text{noise} &= \int \int \{h(\mathbf{x}) - t\}^2 d\mathbf{x} dt
\end{aligned}
$$

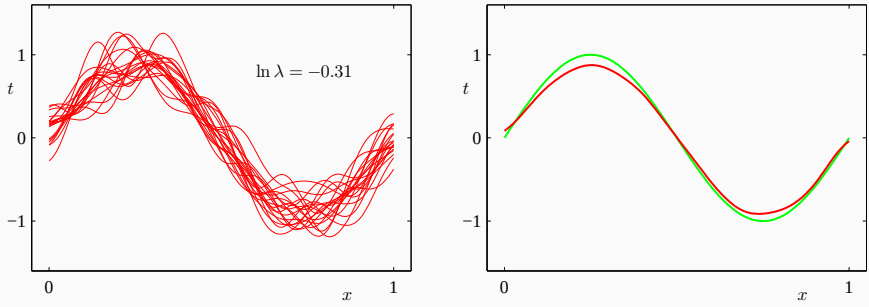
Regularization, bias, and variance

- Bias and variance depend on model complexity.
- Low complexity, implies high bias and low variance:



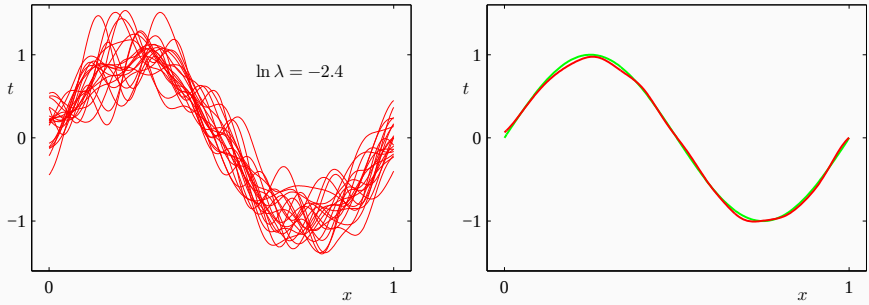
Regularization, bias, and variance

- Bias and variance depend on model complexity.
- Relaxing the regularization coefficient, reduces bias and increases variance:



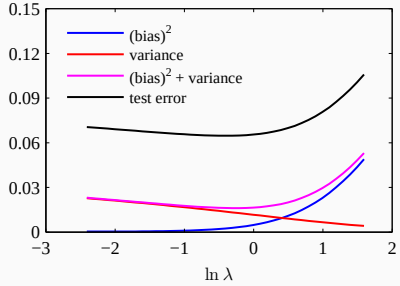
Regularization, bias, and variance

- Bias and variance depend on model complexity.
- Relaxing the regularization coefficient, reduces bias and increases variance:



Regularization, bias, and variance

- This is all very nice and all, but these integrals are completely intractable.
- These integrals are average over datasets, so we should have an ensemble of independent datasets...
- And it is nearly impossible to robustly estimate bias and variance.
- We will see more practical methods to estimate optimal empirical tradeoffs.



Bayesian Linear Regression

I’m not happy

- We might look at these regression results and, although pretty, conclude: I’m not happy.
- Why might we not be happy? We have developed a set of sophisticated mathematical tools to estimate functions from data. What more do you want?

It's all about belief

- All of this sophisticated mathematical maximum likelihood machinery is great, but it doesn't really help us understand how much we should believe in a particular solution.
- In this case, belief takes on a whole host of useful meanings:
  - My regression barfs out a $\mathbf{w}_{\mathrm{ml}}$ from data $\mathcal{D}$. Great, but how reliable is that $\mathbf{w}_{\mathrm{ml}}$, really? How much do I believe it is close to the true $\mathbf{w}^*$ that we assume generated the $\mathcal{D}$.
  - I predict a t on some new input $\mathbf{x}'$ using $t' = y(\mathbf{x}', \mathbf{x}_{\mathrm{ml}})$. Great, but how much do I believe in this t'? Is this belief constant across the whole input space?
  - What if I have prior knowledge (i.e. a belief about my parameter distribution $p(\mathbf{w})$ can I incorporate this into my estimate of $\mathbf{w}_{\mathrm{ml}}$?
- The broad class of Bayesian techniques give us exactly these tools by exploiting likelihood, prior, and evidence.

Sometimes it's also about sequential learning...

- What if we train a model using data $\mathcal{D}_1$.
- Then, tomorrow, someone dumps new data $\mathcal{D}_2$ on us.
- What can we do? Do we have to train the whole model from scratch using $\mathcal{D} = \bigcup_i \mathcal{D}$?

The parameter distribution

- We want to quantify our belief in a specific model $\mathbf{w}^*$ estimated from $\mathcal{D}$.
- Always remember your Bayes rule:

$$p(\mathbf{w} \mid \mathbf{t}) = \frac{\text{data likelihood} \times \text{prior}}{\text{evidence}}$$

- We have already derived a likelihood for data given model:

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

- We need a prior distribution $p(\mathbf{w})$ that expresses our prior belief in likely values $\mathbf{w}$ might take.
- Note the form of the data likelihood and that we will multiply it with this prior.

The parameter distribution

- Let's pick our prior first of all so that it is a reasonable expectation.
- For example, we might expect our weights to be *close* to zero, on average, with some expected variance around zero.
- Let's also pick the form of our prior so it "plays nice" with the likelihood:

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

- This is a Gaussian Conjugate Prior, which just simply means that when we multiply it with a Gaussian likelihood, the resulting posterior is also Gaussian:

$$\begin{aligned} p(\mathbf{w} \mid \mathbf{t}) &= p(\mathbf{t} \mid \mathbf{w}, \beta^{-1}) p(\mathbf{w} \mid \alpha) \\ &= \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N), \\ \text{where } \mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^T \mathbf{t} \\ \text{and } \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi} \end{aligned}$$

The parameter distribution

- Keeping everything nice and Gaussian has many advantages – the log posterior is:

$$\ln p(\mathbf{w} \mid \mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^2\mathbf{w} + \text{const}$$

- Look familiar?

The parameter distribution

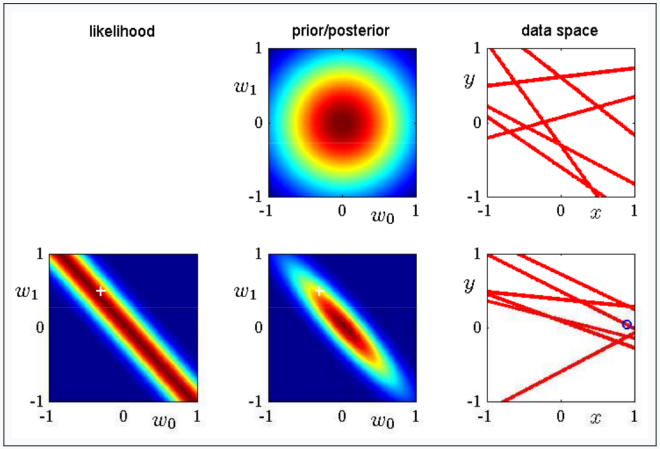
- Keeping everything nice and Gaussian has many advantages – the log posterior is:

$$\ln p(\mathbf{w} \mid \mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^2\mathbf{w} + \text{const}$$

- Maximizing this posterior yields the same solution as our “normal” regularized least squares with $\lambda = \alpha/\beta$!
- But note that we have something intrinsically more powerful here.
- We have achieved some of our goals:
  - We can quantify belief in a solution $\mathbf{w}^*$ (should be clear).
  - We can also learn incrementally when new data arrives (probably not so clear).
- Let’s look at a simple line fitting example...

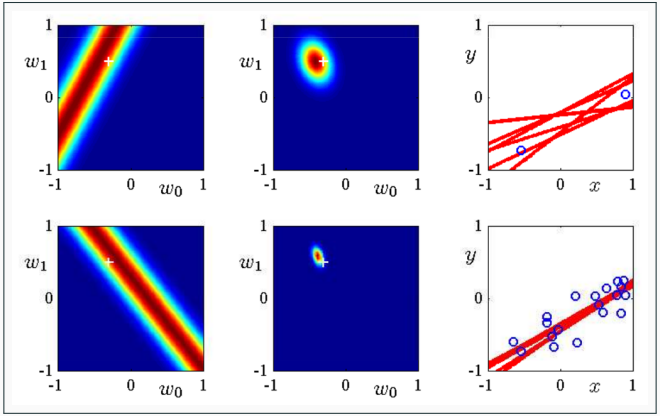
The parameter distribution

- We start with no data... So, we are left with the posterior equal to the prior:
- When we start observing data, we use Bayes rule to update belief.



The parameter distribution

- As we keep observing data, we keep using Bayes rule to update belief.
- Eventually, variance reduces and we stabilize to a posterior estimate around the ML solution.



Predictive distribution

- Note the we haven't said anything about predictions from our model.
- If I produce an output $y(\mathbf{t}, \mathbf{w}^*)$, how happy am I with it? Does it depend on $\mathbf{x}$ in any way?
- In practice, we couldn't care less about the actual value of $\mathbf{w}$, we just want predictions!
- Define the predictive distribution then as:

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

- The first way to look at this is as a average (i.e. expectation) of conditional likelihoods, where the expectation is with respect to the posterior (parameter distribution).

Predictive distribution

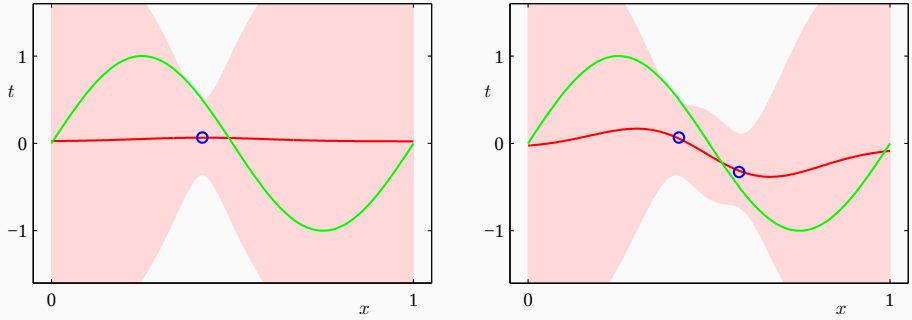
- If we dig down into the Gaussian nature of both of these, and note that the predictive distribution is a convolution of two Gaussian, we can derive an analytic form:

$$\begin{aligned} p(t \mid \mathbf{t}, \alpha, \beta) &= \mathcal{N}(t \mid \mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})), \\ \text{where } \sigma_N^2(\mathbf{x}) &= \frac{1}{\beta} + \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}) \end{aligned}$$

- $1/\beta$ represents noise in our data, and σ_N^2 represents uncertainty in our parameter estimation.
- Also note that $\sigma_{N+1}^2(\mathbf{x}) < \sigma_N^2(\mathbf{x})$, so more data is always a good thing.

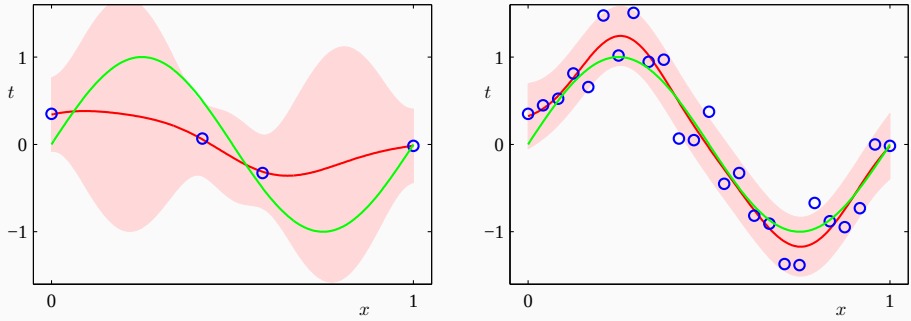
Predictive distribution

- Now we're really saying something useful about model outputs:



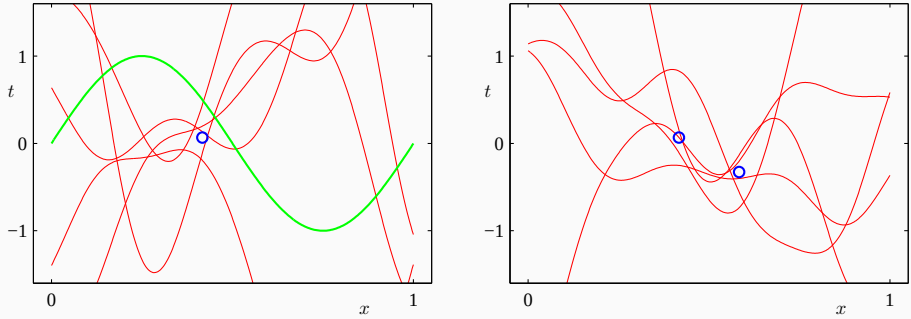
Predictive distribution

- Now we're really saying something useful about model outputs:



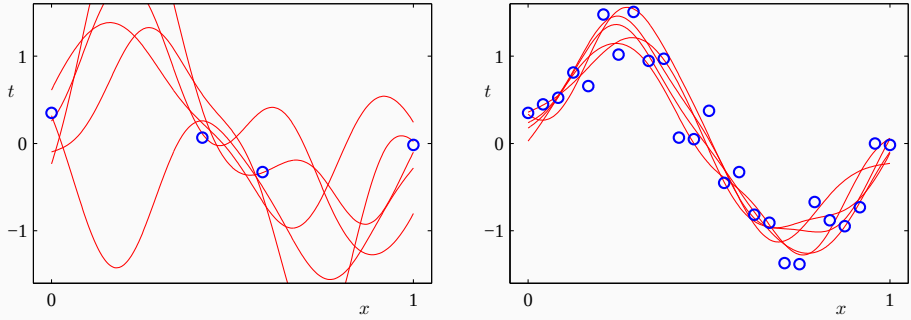
Predictive distribution

- And about model variance (by sampling from posterior over $\mathbf{w}$)::



Predictive distribution

- And about model variance (by sampling from posterior over $\mathbf{w}$)::



"The best regularizer is always more annotated data."

– Geoffrey Hinton (probably).

Concluding remarks

Linear regression in three acts

- We have seen (at least) three views of linear regression:
  1. The purely geometric view;
  2. The Maximum Likelihood view; and
  3. The Bayesian view.
- This barely scratches the surface of what is possible, but it is a good foundation.
- Interestingly, the solutions for all three views are identical.
- Each has, however, different advantages and disadvantages: a point estimate is efficient, while full Bayesian inference has more features.
- Important: the full Bayesian treatment is possible in analytic form only because of the choices we made about the prior over weights and observation noise.

The way forward

- Next we will turn our attention to linear models for classification.
- Again, we will begin with a geometric model of discriminant functions.
- Then we will proceed to apply probabilistic and Bayesian reasoning on top of our intuition.
- Though linear models are simple, they are also often very effective and their simple formulation admits simple and robust inference.
- This inferential simplicity we will have to leave behind when we move on to more complex models.

Reading and Homework Assignments

Reading Assignment:

- Bishop: Chapter 3 (3.1, 3.2, 3.3) – these are the same as in the last lecture!

Homework:

- See accompanying Jupyter Notebook in the Moodle (when I upload it).