Data-Driven Design & Analyses of Structures & Materials (3dasm)

Lecture 12

Miguel A. Bessa | **miguel_bessa@brown.edu** | Associate Professor

Outline for today

- Derivation of a few more Linear Regression models:
    - Ridge regression
    - Lasso regression
    - Bayesian linear regression

Reading material: This notebook + Chapter 11 of the book.

Summary of important Linear Regression Models

Recall our view of Linear regression models from a Bayesian perspective: it's all about the choice of **likelihood** and **prior**!

Likelihood	Prior (on the weights)	Posterior	Name of the model	Book section
Gaussian	Uniform	Point estimate	Least Squares regression	11.2.2
Gaussian	Gaussian	Point estimate	Ridge regression	11.3
Gaussian	Laplace	Point estimate	Lasso regression	11.4
Student-t	Uniform	Point estimate	Robust regression	11.6.1
Laplace	Uniform	Point estimate	Robust regression	11.6.2
Gaussian	Gaussian	Gaussian	Bayesian linear regression	11.7

Last Lecture we covered Least Squares regression. Today we will cover Ridge regression, Lasso regression and Bayesian linear regression.

Ridge regression: Linear regression with Gaussian likelihood, Gaussian prior and posterior via Point estimate

Likelihood	Prior (on the weights)	Posterior	Name of the model	Book section
Gaussian	Gaussian	Point estimate	Ridge regression	11.3

1. Gaussian observation distribution: $p(y|\mathbf{x}, \mathbf{z}) = \mathcal{N}(y|\mu_{y|z} = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \sigma^2_{y|z} = \sigma^2)$
   where $\mathbf{z} = \mathbf{w}$ are all the unknown model parameters (hidden rv's), but where σ is a **specified** value (i.e. σ is chosen by us, unlike in the last Lecture where we had considered it an unknown).

1. But using a Gaussian prior for the weights $\mathbf{w}$: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \hat{\sigma}^2_w \mathbf{I})$ where $\hat{\sigma}_w$ is **specified** by us.

1. MAP point estimate for posterior: $\hat{\mathbf{z}}_{\text{map}} = \underset{z}{\text{argmin}} \left[-\sum_{i=1}^N \log p(y = y_i|\mathbf{z}) - \log p(\mathbf{w}) \right]$,

Final prediction is given by the PPD: $p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \mathbf{z}) \delta(\mathbf{z} - \hat{\mathbf{z}}_{\text{map}}) dz = p(y^*|\mathbf{x}^*, \mathbf{z} = \hat{\mathbf{z}}_{\text{map}}, \mathcal{D})$

Important concepts for all ML models: parameters vs hyperparameters

At this point we have to highlight something very important that we have referred to multiple times, but never used the formal nomenclature:

- The difference between **parameters** and **hyperparameters**

Parameters of an ML model: these are the unknown variables $\mathbf{z}$, i.e. the rv's that are hidden (not explicitly visible in our data).Our goal when making a prediction from the PPD is to:

- Marginalize these unknown parameters $\mathbf{z}$, if using Bayesian inference (i.e. integrating them out)
- **OR** estimate the value of these parameters $\mathbf{z}$ using a Point estimate $\hat{\mathbf{z}}$ (if doing deterministic inference, i.e. the common machine learning models without uncertainty estimation)

Hyperparameters of an ML model: these are the variables that we are specifying or assuming for our model and that are not going to be updated after training, i.e. they do not change when we determine the PPD.

- For example, for the above-mentioned Ridge regression model we defined **3 hyperparameters**:
    - The prior parameters $\check{\boldsymbol{\mu}}_w$ (that we set to $\mathbf{0}$) and $\check{\sigma}_w$ (that we can set to any constant value cte)
    - **AND** the variance of the observation distribution σ (that you set to whatever particular value you want)

A note about the prior

Ususally the Gaussian prior is imposed only on the weights (not on the bias term, i.e. the w_0 term):

- This is actually very common. The bias term in the observation distribution usually has a Uniform prior because it does not contribute to overfitting.

You can also see this in the previous Lecture from the expressions for $\hat{w}_{0,\text{mle}}$ and σ^2_{mle}, as they act on the global mean and MSE (mean squared error) of the residuals, respectively.

Coming back to our Ridge regression model, computing the MAP estimate is very similar to what we did in the last Lecture, leading to:

$$\hat{\mathbf{w}}_{\text{map}} = \underset{w}{\text{argmin}} \left[\frac{1}{2\sigma^2} (\mathbf{\Phi w} - \mathbf{y})^T (\mathbf{\Phi w} - \mathbf{y}) + \frac{1}{2\tilde{\sigma}_w^2} \mathbf{w}^T \mathbf{w} \right] \tag{1}$$

$$= \underset{w}{\text{argmin}} \left[\text{RSS}(\mathbf{w}) + \alpha ||\mathbf{w}||_2^2 \right] \tag{2}$$

where $\alpha = \frac{\sigma^2}{\tilde{\sigma}_w^2}$ is proportional to the strength of the prior (depends on two hyperparameters), and $||\mathbf{w}||_2 = \sqrt{\sum_{m=1}^{M-1} |w_m|^2} = \sqrt{\mathbf{w}^T \mathbf{w}}$
is called the l_2 norm of the vector $\mathbf{w}$. Thus, we are penalizing weights that become too large in magnitude. In ML literature this is usually called l_2 **regularization** or **weight decay**, and is very widely used.
In Homework 4, you explore the difference between Linear Least Squares and Ridge regression, reporting on the influence of the prior strength for the latter.

Then, solving the MAP first for the weights $\mathbf{w}$, as we did for the MLE:

$$\nabla_{\mathbf{w}} \left[\frac{1}{2\sigma^2} (\mathbf{\Phi w} - \mathbf{y})^T (\mathbf{\Phi w} - \mathbf{y}) + \frac{1}{2\hat{\sigma}_w^2} \mathbf{w}^T \mathbf{w} \right] = \mathbf{0} \tag{3}$$

$$\nabla_{\mathbf{w}} \left[(\mathbf{\Phi w} - \mathbf{y})^T (\mathbf{\Phi w} - \mathbf{y}) + \alpha \mathbf{w}^T \mathbf{w} \right] = \mathbf{0} \tag{4}$$

from which we determine the MAP estimate for the the weights $\mathbf{w}$ as:

$$\hat{\mathbf{w}}_{\text{map}} = \left(\mathbf{\Phi}^T \mathbf{\Phi} + \alpha \mathbf{I}_M \right)^{-1} \mathbf{\Phi}^T \mathbf{y} = \left(\sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) \boldsymbol{\phi}(\mathbf{x}_n)^T + \alpha \mathbf{I}_M \right)^{-1} \left(\sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) y_n \right)$$

Once again, this can be solved using SVD or other methods to ensure that the Moore-Penrose pseudo-inverse is calculated properly.

Lasso regression: Linear regression with Gaussian likelihood, Laplace prior and posterior via Point estimate

Likelihood	Prior (on the weights)	Posterior	Name of the model	Book section
Gaussian	Laplace	Point estimate	Lasso regression	11.4

1. Gaussian observation distribution: $p(y|\mathbf{x}, \mathbf{z}) = \mathcal{N}(y|\mu_{y|z} = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \sigma^2_{y|z} = \sigma^2)$

where $\mathbf{z} = \mathbf{w}$ are all the unknown model parameters (hidden rv's), and σ is a hyperparameter.

1. But using a **Laplace** prior for the weights $\mathbf{w}$: $p(\mathbf{w}) = \prod_{m=1}^{M-1} \text{Lap}\left(w_m|0, 1/\overset{<}{\lambda}_w\right) \propto \prod_{m=1}^{M-1} \exp\left[-\overset{<}{\lambda}_w|w_m|\right]$
2. MAP point estimate for posterior: $\hat{\mathbf{z}}_{\text{map}} = \underset{z}{\text{argmin}}\left[-\sum_{i=1}^{N} \log p(y = y_i|\mathbf{z}) - \log p(\mathbf{w})\right]$

Final prediction is given by the PPD: $p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \mathbf{z})\delta(\mathbf{z} - \hat{\mathbf{z}}_{\text{map}})dz = p(y^*|\mathbf{x}^*, \mathbf{z} = \hat{\mathbf{z}}_{\text{map}}, \mathcal{D})$

Computing the MAP estimate:

$$\hat{\mathbf{w}}_{\text{map}} = \underset{w}{\text{argmin}} \left[(\mathbf{\Phi w} - \mathbf{y})^T (\mathbf{\Phi w} - \mathbf{y}) + \overset{<}{\lambda}_w ||\mathbf{w}||_1 \right] \tag{5}$$

where $||\mathbf{w}||_1 = \sum_{m=1}^{M-1} |w_m|$ is called the l_1 norm of $\mathbf{w}$. In ML literature this is called l_1 regularization.

- Calculating the MAP for Lasso is not done the same way as for Ridge because the term $||\mathbf{w}||_1$ is not differentiable whenever $w_m = 0$. In this case, the solution is found using hard- or soft-thresholding (See Section 11.4.3 in the book) because the gradient becomes a branch function.

- A more important point is to see that **different types of prior distributions** introduce **different regularizations** on the weights, aleviating overfitting in a different manner.
    - For example, in the case of Lasso, since the Laplace prior puts more density around the mean (which is zero here) than the Gaussian prior, then it has a tendency to lead the weights to zero, i.e. it introduces **sparsity** when estimating the weights via MAP. The book has a beautiful discussion about this.

Bayesian linear regression: Linear regression with Gaussian likelihood, Gaussian prior and Gaussian posterior (Bayesian solution)

As we saw in the beginning of the Lecture, there are many more models we can define! The book covers quite a few!

Likelihood	Prior (on the weights)	Posterior	Name of the model	Book section
Gaussian	Gaussian	Gaussian	Bayesian linear regression	11.7

1. Gaussian observation distribution (with **known** variance): $p(y|\mathbf{x}, \mathbf{z}) = \mathcal{N}(y|\mu_{y|z} = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \sigma^2_{y|z} = \sigma^2)$ where $\mathbf{z} = \mathbf{w}$ are all the unknown model parameters (hidden rv's), and where σ is **not** treated as an unknown, i.e. it is a **hyperparameter**.
2. Gaussian prior for the weights $\mathbf{w}$: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\overset{<}{\boldsymbol{\mu}}_w, \overset{<}{\boldsymbol{\Sigma}}_w)$
3. Gaussian posterior (obtained from Bayes rule)

Final prediction is given by the PPD: $p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \mathbf{z}) p(\mathbf{z}|\mathcal{D}) dz$

At this point you may notice that we have already derived this model in Lecture 7.
The only differences are that now we have multiple weights $\mathbf{z}$, multidimensional inputs $\mathbf{x}$ and that we allow them to have different values.
Yet, the derivation is the same! We just need to bold the letters. Let's do it:
The likelihood is a (multivariate) Gaussian distribution (product of MVN evaluated at each data point):

$$p(\mathcal{D}|\mathbf{w}, \sigma^2) = \prod_{n=1}^{N} p(y_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \sigma^2) = \mathcal{N}(\mathbf{y}|\boldsymbol{\Phi}\mathbf{w}, \sigma^2 \mathbf{I}_N)$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix, as defined previously.

To calculate the posterior, we also use the product of Gaussians rule (in Lecture 5 we also defined this rule for multivariate Gaussians!):

$$p(\mathbf{w}|\mathbf{\Phi}, \mathbf{y}, \sigma^2) \propto \mathcal{N}(\mathbf{y}|\mathbf{\Phi}\mathbf{w}, \sigma^2\mathbf{I}_N)\mathcal{N}(\mathbf{w}|\hat{<}{\boldsymbol{\mu}}_w, \hat{<}{\boldsymbol{\Sigma}}_w) = \mathcal{N}(\mathbf{w}|\hat{>}{\boldsymbol{\mu}}_w, \hat{>}{\boldsymbol{\Sigma}}_w)$$

where the mean and covariance of the posterior are given by:

$$\hat{>}{\boldsymbol{\mu}}_w = \hat{>}{\boldsymbol{\Sigma}}_w \left(\hat{<}{\boldsymbol{\Sigma}}_w^{-1} \hat{<}{\boldsymbol{\mu}}_w + \frac{1}{\sigma^2}\mathbf{\Phi}^T\mathbf{y} \right)$$

$$\hat{>}{\boldsymbol{\Sigma}}_w = \left(\hat{<}{\boldsymbol{\Sigma}}_w^{-1} + \frac{1}{\sigma^2}\mathbf{\Phi}^T\mathbf{\Phi} \right)^{-1}$$

Two additional notes about this result (it's recommended to go through these notes):

1. The above result for the posterior is similar to what we did before in Lecture 5, but it involves a little more algebra. If you are curious, I derived it for you (see **NOTE 1** below).
2. It's easy to reduce the posterior mean for Bayesian linear regression to the MAP estimate of Ridge regression (see **NOTE 2** below)

Having determined the posterior, we can determine what we really want: the PPD.

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \mathbf{z}) p(\mathbf{z}|\mathcal{D}) d\mathbf{z} \tag{8}$$

$$p(y^*|\mathbf{x}^*, \mathcal{D}, \sigma^2) = \int p(y^*|\mathbf{x}^*, \mathbf{w}, \sigma^2) p(\mathbf{w}|\mathcal{D}) d\mathbf{w} \tag{9}$$

$$= \int \mathcal{N}(y^*|\boldsymbol{\phi}(\mathbf{x}^*)^T \mathbf{w}, \sigma^2) \mathcal{N}(\mathbf{w}|\hat{\boldsymbol{\mu}}_w, \hat{\boldsymbol{\Sigma}}_w) d\mathbf{w} \tag{10}$$

We calculate this integral like we did in previous lectures, i.e. again by making explicit the observation distribution as a function of $\mathbf{w}$ and then do the product of the two MVNs. This time, I really don't think we need to show these steps again... So, I will just skip the same procedure and give the final result:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}\left(y^* \mid \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\mu}}_w, \sigma^2 + \boldsymbol{\phi}(\mathbf{x})^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\phi}(\mathbf{x})\right)$$

REVIEW THE NOTATION ABOUT INTEGRATING THE PPD FOR MULTIPLE UNKNOWNS

The book uses the following notation:

$$p(y^*|x^*, \mathcal{D}, \sigma^2) = \int p(y^*|x^*, \mathbf{w}, \sigma^2) p(\mathbf{w}|\mathcal{D}) d\mathbf{w} \tag{11}$$

We could be more explicit and use the following notation:

$$p(y^*|x^*, \mathcal{D}, \sigma^2) = \int p(y^*|x^*, \mathbf{w}, \sigma^2) p(\mathbf{w}|\mathcal{D}) d^M\mathbf{w} \tag{12}$$

$$= \int \int \cdots \int p(y^*|x^*, \mathbf{w}, \sigma^2) p(\mathbf{w}|\mathcal{D}) dw_0 dw_1 \cdots dw_{M-1} \tag{13}$$

where it is clear that we are integrating for all variables (not integrating to get a vector).
However, despite this notation being more precise, it is also less appealing. Just make sure you realize the type of integral that we are calculating when we are finding the PPD.

We conclude that the PPD of the Bayesian linear regression model is:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}\left(y^* \mid \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\mu}}_w,\, \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\phi}(\mathbf{x}^*)\right)$$

where the mean and covariance of the posterior are given by:

$$\hat{\boldsymbol{\mu}}_w = \hat{\boldsymbol{\Sigma}}_w \left(\check{\boldsymbol{\Sigma}}_w^{-1} \check{\boldsymbol{\mu}}_w + \frac{1}{\sigma^2} \boldsymbol{\Phi}^T \mathbf{y}\right)$$

$$\hat{\boldsymbol{\Sigma}}_w = \left(\check{\boldsymbol{\Sigma}}_w^{-1} + \frac{1}{\sigma^2} \boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1}$$

And where we recall the meaning of each term:

- $(\mathbf{x}, y)$ is the point were we want to make a prediction
- $\boldsymbol{\phi}(\mathbf{x})$ is an $M \times 1$ vector of basis functions
- $\hat{\boldsymbol{\mu}}_w$ is the $M \times 1$ vector with the mean of the posterior for the weights
- and $\hat{\boldsymbol{\Sigma}}_w$ is the $M \times M$ covariance matrix of the posterior for the weights.

$$p(y^*|\mathbf{x}^*, \mathcal{D}, \sigma^2) = \mathcal{N}\left(y^* \mid \hat{\boldsymbol{\mu}}_w^T \boldsymbol{\phi}(\mathbf{x}^*)\,,\, \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\phi}(\mathbf{x}^*)\right) \tag{14}$$

The above PPD for Bayesian linear regression is quite interesting because:

1. The PPD does not depend on the weights $\mathbf{w}$. This is in contrast with the other models we saw for linear regression that depended on point estimates, instead of following the full Bayesian treatment.

1. The variance of the PPD at a point $\mathbf{x}^*$ after seeing N data points depends on two terms: (1) the variance of the observation noise, σ^2 that we defined to be constant; and (2) the variance in the parameters obtained by the posterior $\hat{\boldsymbol{\Sigma}}_w$.

This means that the predicted uncertainty increases when $\mathbf{x}^*$ is located far from the training data $\mathcal{D}$, just like we want it to be! We are less certain about points **away** from our observations (training data).

You will see that more advanced methods such as Gaussian processes are not too different from Bayesian linear regression...

Finally, for convenience, let's write the PPD of Bayesian linear regression including all the terms to get the complete (and rather long!) expression:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}\left(y^* \mid \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\Sigma}}_w \left(\check{\boldsymbol{\Sigma}}_w^{-1} \check{\boldsymbol{\mu}}_w + \frac{1}{\sigma^2}\boldsymbol{\Phi}^T \mathbf{y}\right),\ \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\phi}(\mathbf{x}^*)\right)$$

with the previously determined covariance: $\hat{\boldsymbol{\Sigma}}_w = \left(\check{\boldsymbol{\Sigma}}_w^{-1} + \frac{1}{\sigma^2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}$

This long expression of the PPD is (very!) **often simplified** by considering a prior with zero mean, i.e. $\check{\boldsymbol{\mu}}_w = \mathbf{0}$:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}\left(y^* \mid \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\Sigma}}_w \frac{1}{\sigma^2}\boldsymbol{\Phi}^T \mathbf{y},\ \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^*)^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\phi}(\mathbf{x}^*)\right)$$

Other linear regression models

As we saw in the beginning of the Lecture, there are many more linear regression models we can define! The book covers quite a few!

Likelihood	Prior (on the weights)	Posterior	Name of the model	Book section
Gaussian	Uniform	Point estimate	Least Squares regression	11.2.2
Gaussian	Gaussian	Point estimate	Ridge regression	11.3
Gaussian	Laplace	Point estimate	Lasso regression	11.4
Gaussian	Gaussian$\times$Laplace	Point estimate	Elastic net	11.4.8
Student-t	Uniform	Point estimate	Robust regression	11.6.1
Laplace	Uniform	Point estimate	Robust regression	11.6.2
Gaussian	Gaussian	Gaussian	Bayesian linear regression	11.7

For example, the "Elastic net" is literally the combination of Ridge regression and Lasso by defining a prior that depends on both a Gaussian and a Laplace distribution. This model was proposed in 2005. Five years later the "Bayesian Elastic Net" was also proposed, where the posterior is calculated in a Bayesian way (just like what we did for Bayesian linear regression).

Summary

Although we only considered Linear Regression Models until now, we are starting to realize something very important:

ML models can be derived from the Bayes' rule by following the same 4 steps!

1. Define the Observation distribution and compute the likelihood.
2. Define the prior and its parameters.
3. Compute the posterior to estimate the unknown parameters (whether via a Bayesian approach or via a Point estimate).
4. Compute the PPD.

Done.

If you only care about the mean of the PPD (the mean of your prediction), then you can compute it directly without even thinking about uncertainty... But then, make sure you characterize the quality of your predictions using an appropriate **error metric** and use strategies like **cross-validation**, as you will explore in Homework 4!

See you next class

Have fun!