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

Lecture 8

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

Outline for today

- Parameter estimation from training with data (model fitting)
    - Posterior approximation by Dirac delta "distribution"
    - Point estimates for the Dirac delta "distribution"
        - MAP: Maximum A Posterior estimate
        - MLE: Maximum Likelihood Estimation
    - Negative log likelihood (NLL)
- Why some people do not adopt a Bayesian (probabilistic) perspective of ML

Reading material: This notebook + Chapter 4

Summary of past lectures

Bayesian inference:

- predicts a **quantity of interest** (e.g. y) while treating **unknown** information as rv's (e.g. z)
- it is based on establishing a model (observation distribution + prior) and evaluating it on data (joint likelihood normalized by marginal likelihood) to update our belief about the unknown (posterior)
- from the posterior, we can then predict a distribution for the quantity of interest (the PPD) that results from marginalizing (integrating out) the unknown

The good and the bad

In short: Bayesian inference results from interpreting the unknown as rv's of a model, and then evaluating the impact of all possible values of the rv's (within the constraints imposed by the model!) by marginalizing them (integrating them out).

- **The good**: This is powerful because even if our assumptions are wrong, we can at least consider different values for the rv's and know their respective impact on the predictions. This alleviates problems such as overfitting and overconfidence, that we will encounter in the remaining of the course.

- **The bad**: Bayesian inference can be difficult. We solved one of the simplest problems in the last lectures, and we saw that those integrals are a bit ugly...
    - In most cases, the integrals (to compute the marginal likelihood, and the PPD) cannot even be solved analytically.
    - Numerical strategies exist to approximate the integration, but they tend to be **slow when accurate** or **fast but innacurate** (a dangerous generalization: forgive me Bayesians!)

Very Important Question (VIQ): What if we don't calculate these integrals at all?

Machine Learning without going fully Bayesian

Avoiding integration is possible by noting that:

1. Computing the PPD is trivial if the **posterior distribution becomes the Dirac delta**
1. The marginal likelihood is just a **constant**

Let's explore these two remarks.

1. PPD when the posterior is a Dirac delta

$$p(y^*|\mathcal{D}_y) = \int \underbrace{p(y^*|z)}_{\substack{\text{observation} \\ \text{distribution}}} \overbrace{p(z|y=\mathcal{D}_y)}^{\text{posterior}} dz$$

where y^* signals that this is a prediction of y (to help distinguish from training data).
What happens if the posterior is the Dirac delta "distribution"?

$$p(z|y=\mathcal{D}_y) = \delta(z-\hat{z})$$

where $\hat{z}$ is our best estimate for the value that z should have.

$$p(y^*|\mathcal{D}_y) = \int \underbrace{p(y^*|z)}_{\substack{\text{observation} \\ \text{distribution}}} \overbrace{p(z|y=\mathcal{D}_y)}^{\text{posterior}} dz \tag{1}$$

$$= \int p(y^*|z)\delta(z-\hat{z})dz \tag{2}$$

$$= p(y^*|z=\hat{z}) \tag{3}$$

Conclusion: The PPD becomes the **observation distribution** where the unknown z becomes our **best estimate** $\hat{z}$ (in other words: $z = \hat{z} =$ const)

- But what is our "**best estimate**" $\hat{z}$?
    - There are different estimates and different strategies to get there!

2. Finding the "best estimate" $\hat{z}$ without computing the marginal likelihood

Remember: the Bayes' rule determines the posterior,

$$p(z|y=\mathcal{D}_y) = \frac{p(y=\mathcal{D}_y|z)p(z)}{p(y=\mathcal{D}_y)}$$

and the marginal likelihood $p(y=\mathcal{D}_y)$ is just a constant.

If we want to reduce the posterior to the Dirac delta "distribution",

$$p(z|y=\mathcal{D}_y) = \delta(z-\hat{z})$$

what is the only parameter that we need to find?

- We just need to find $\hat{z}$ to completely characterize $\delta(z - \hat{z})$

Note that this is not the case if the posterior is a different distribution!

For example, we saw in the previous lectures that the posterior for the car stopping distance problem was a **Gaussian**.

- How many parameters do you need to characterize the Gaussian distribution?

Indeed... Two!
And if the posterior distribution is more complicated, you may need a lot more parameters! In some cases, the posterior does not even have an analytical description!

Anyway, the question still remains: what should be the value $\hat{z}$?
Let's go back to the two problems we have seen in Lecture 6 and Lecture 7.

Recall our reflection on the differences between the posterior for the two priors we used.

- When using the noninformative Uniform prior $p(z) = \frac{1}{C_z}$ (Lecture 6):

$$p(z|y = \mathcal{D}_y) = \mathcal{N}(z|\mu, \sigma^2) \tag{4}$$

- When using a Gaussian prior $p(z) = \mathcal{N}\left(z|\overset{<}{\mu}_z, \overset{<}{\sigma}_z^2\right)$ (Lecture 7):

$$p(z|y = \mathcal{D}_y) = \mathcal{N}\left(z|\overset{>}{\mu}_z, \overset{>}{\sigma}_z^2\right) = \mathcal{N}\left(z \middle| \frac{1}{\frac{1}{\sigma^2} + \frac{1}{\overset{<}{\sigma}_z^2}}\left(\frac{\mu}{\sigma^2} + \frac{\overset{<}{\mu}_z}{\overset{<}{\sigma}_z^2}\right), \frac{1}{\frac{1}{\sigma^2} + \frac{1}{\overset{<}{\sigma}_z^2}}\right) \tag{5}$$

The posterior is still a Gaussian but its mean and variance have been updated by the influence of the prior!

Let's play a simple game:

- Choose where to place the Dirac delta "distribution" for those two posteriors we found before.

In [5]:
```
# Showing posteriors and Dirac delta with interactive plot. Code is hidden in presentation.
       from ipywidgets import interactive # so that we can interact with the plot
interactive_plot = interactive(Posteriors_and_Dirac_delta,
                      z_hat_case1=(min(zrange_case1), max(zrange_case1), 6/10*sigma_posterior_UniformPrior),
                      z_hat_case2=(min(zrange_case2), max(zrange_case2), 6/10*sigma_posterior_GaussianPrior) )
interactive_plot
```

```
interactive(children=(FloatSlider(value=0.9197345124021497, description='z_hat_case1', max=
3.9989359480801534,…
```

Probably you didn't hesitate to place the Dirac delta "distribution" at the mean or mode (they are the same for a Gaussian distribution)!

What if the Posterior distribution is something else? For example, a Gamma distribution

```
In [8]: # Showing posteriors and Dirac delta with interactive plot. Code is hidden in presentation.
        interactive_plot = interactive(Gamma_Posterior_and_Dirac_delta,z_hat=(0.5, 6.5, 0.5 ) )
interactive_plot
```

```
interactive(children=(FloatSlider(value=0.5, description='z_hat', max=6.5, min=0.5, step=0.
5), Output()), _dom…
```

Maybe now you are hesitating where to place it?

Both are used in practice! And there are other estimates...
These are called **point estimates**.

- They reduce each unknown rv z to a point $\hat{z}$ (transforming the posterior distribution into the Dirac delta "distribution").

Of course, as everything in life, some choices are better than others...

Common point estimates for determining $\hat{z}$:

- Maximum Likelihood Estimation (MLE):
    - You choose the mode (the maximum) of the posterior but you used a Uniform prior
- Maximum A Posterior (MAP) estimate:
    - You choose the mode (the maximum) of the posterior (and your prior is **not** Uniform)
- Posterior mean estimate (no accronym!):
    - You choose the mean of the posterior.
- ... and so on

Calculating the **Posterior mean estimate** is not new to us (see Lecture 1):

$$\mathbb{E}[z|\mathcal{D}] = \int_{\mathcal{Z}} zp(z|\mathcal{D})dz$$

But I told you that today we are all about avoiding integrals!

So, let's focus on two very common point estimates: **MAP** and **MLE**.
Both are obtained by finding the **mode** of the posterior (i.e. maximum location in the posterior):

$$\hat{\mathbf{z}} = \underset{z}{\operatorname{argmax}}\, p(z|\mathcal{D})$$

In other words, we need to solve an optimization problem.

But finding the mode of the posterior involves a few simple "tricks"...

$$p(z|y=\mathcal{D}_y) = \frac{p(y=\mathcal{D}_y|z)p(z)}{p(y=\mathcal{D}_y)}$$

Calculating the mode of posterior: Trick 1 (taking the log)

We can separate the three terms of the posterior if we work with its $\log$:

$$\log p(z|y=\mathcal{D}_y) = \log p(y=\mathcal{D}_y|z) + \log p(z) - \log p(y=\mathcal{D}_y)$$

- Note: $\log$ is a monotone function, so the argmax of a function is the same as the argmax of the $\log$ of the function! Mathematically:

$$\hat{\mathbf{z}} = \operatorname*{argmax}_z p(z|\mathcal{D}) = \operatorname*{argmax}_z \log p(z|\mathcal{D})$$

CALCULATING THE MODE: TRICK 2 (MAXIMIZING BY MINIMIZING THE NEGATIVE LOG)

Maximizing a function is the same as **minimizing the negative of a function** (flipping the sign in the end).
Mathematically:

$$\hat{\mathbf{z}} = \underset{z}{\operatorname{argmax}} \ \log p(z|\mathcal{D}) = \underset{z}{\operatorname{argmin}} \left[-\log p(z|\mathcal{D}) \right]$$

- In numerical optimization, this is very common practice!
    - Most optimization algorithms are designed to *minimize* functions.
    - In general, when we are optimizing (whether maximizing or minimizing) functions we call them "**objective**" functions. Yet, in particular:
        - when we are *minimizing* functions we call them "**loss**" or "cost" functions.
        - when we are *maximizing* functions we call them "**reward**" or "score" functions.

CALCULATING THE MODE: FOCUSING ON EACH LOG TERM

$$\hat{\mathbf{z}} = \underset{z}{\operatorname{argmax}} \left[\log p(z|y = \mathcal{D}_y)\right] \tag{6}$$
$$= \underset{z}{\operatorname{argmin}} \left[-\log p(z|y = \mathcal{D}_y)\right] \tag{7}$$
$$= \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y|z) - \log p(z) + \log p(y = \mathcal{D}_y)\right] \tag{8}$$
$$= \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y|z) - \log p(z) + \text{constant}\right] \tag{9}$$

- The last line can be further simplified because a constant does not change the location of the minimum.

So, we get: $\hat{\mathbf{z}} = \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y|z) - \log p(z)\right]$

At this point, recall that the likelihood is usually calculated assuming the training examples (observations) are sampled independently from the observation distribution $p(y|z)$:

$$p(y = \mathcal{D}_y|z) = \prod_{i=1}^{N} p(y = y_i|z)$$

which is known as the **i.i.d.** assumption (independent and identically distributed).

This means that the $\log$ likelihood usually has a very convenient form:

$$\mathrm{LL}(z) = \log p(y = \mathcal{D}_y|z) = \sum_{i=1}^{N} \log p(y = y_i|z)$$

which decomposed into a sum of terms, one per example (observation).

In summary, the mode of the posterior is calculated as:

$\hat{\mathbf{z}} = \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y | z) - \log p(z) \right]$

where the first term is called **negative log likelihood**:

$\operatorname{NLL}(z) = -\operatorname{LL}(z) = -\log p(y = \mathcal{D}_y | z) = -\sum_{i=1}^{N} \log p(y = y_i | z)$

Maximum A Posterior (MAP) estimate

If we choose any prior distribution **except** the Uniform distribution, then the estimate is called MAP:

$\hat{\mathbf{z}}_{\text{map}} = \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y | z) - \log p(z) \right]$

where $p(z)$ is **not** the Uniform distribution.

Maximum Likelihood Estimation (MLE)

In the special case of choosing the prior to be a **Uniform distribution**, $p(z) \propto 1$, then the mode of the posterior becomes the same as the mode of the (log) likelihood:

$$\hat{\mathbf{z}} = \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y | z) - \log p(z) \right] = \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y | z) \right]$$

and we say that we are using the Maximum Likelihood Estimation (MLE) for the unknown z:

$$\hat{\mathbf{z}}_{\text{mle}} = \underset{z}{\operatorname{argmin}} \left[-\log p(y = \mathcal{D}_y | z) \right] = \underset{z}{\operatorname{argmin}} \left[-\sum_{i=1}^{N} \log p(y = y_i | z) \right]$$

where, again, the argument of this expression is called the **negative log likelihood** $\text{NLL}(z)$.

Summary of Machine Learning without going fully Bayesian

1. Approximate posterior by a **Dirac delta** "distribution" $\delta(z - \hat{z})$ where $\hat{z}$ is a chosen **Point estimate**:
    - MLE: $\hat{\mathbf{z}}_{\text{mle}} = \underset{z}{\text{argmin}} \left[-\sum_{i=1}^{N} \log p(y = y_i | z) \right]$
    - MAP: $\hat{\mathbf{z}}_{\text{map}} = \underset{z}{\text{argmin}} \left[-\sum_{i=1}^{N} \log p(y = y_i | z) - \log p(z) \right]$
    - etc.
1. Compute the PPD using the Point estimate $\hat{z}$ and without calculating any integrals:

$$p(y^* | \mathcal{D}_y) = \int p(y^* | z) \delta(z - \hat{z}) dz = p(y^* | z = \hat{z})$$

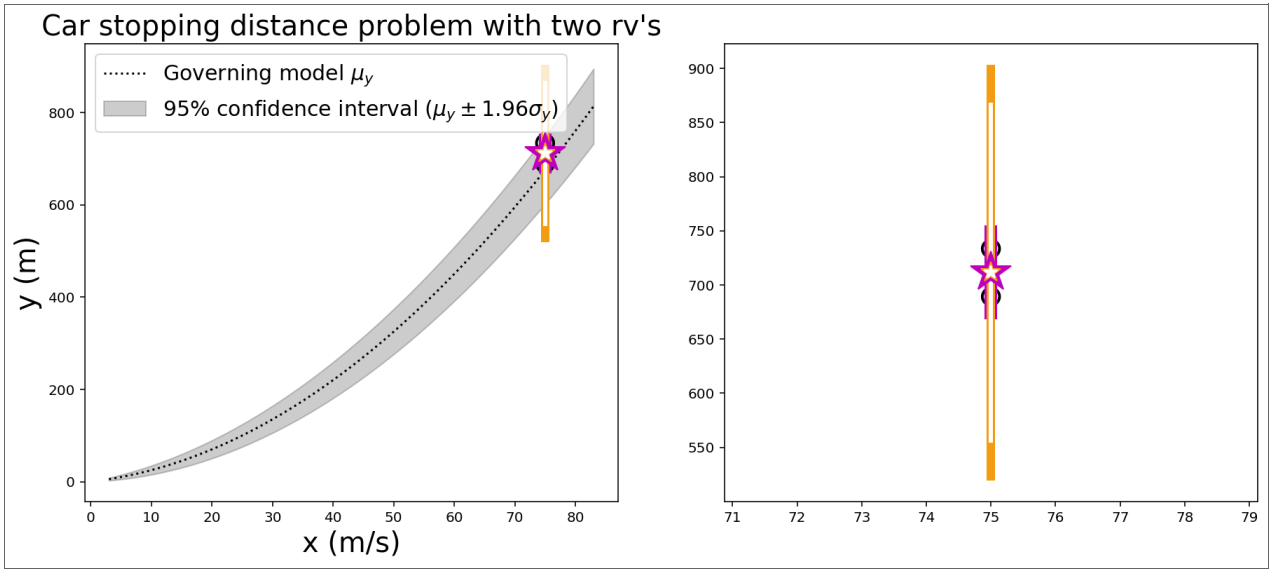
In Homework 3...

1. Using the MLE point estimate, predict the PPD for the car stopping distance problem (Lecture 6).
1. Using the MAP estimate, predict the PPD for the car stopping distance problem considering the Gaussian prior of Lecture 7.
1. Create a plot of the two PPD's and compare them with the PPD's obtained in Lecture 6 and Lecture 7.
    - Note: create these plots of the PPD's such that the abscissa (horizontal) axis is the y rv and the ordinate (vertical axis) is the probability density.

"Teaser": PPD obtained with the MLE **versus** PPD obtained in Lecture 6 (Uniform prior)

```
In [11]: MLE_versus_Bayesian_PPD_for_UniformPrior(N_samples=2)
```

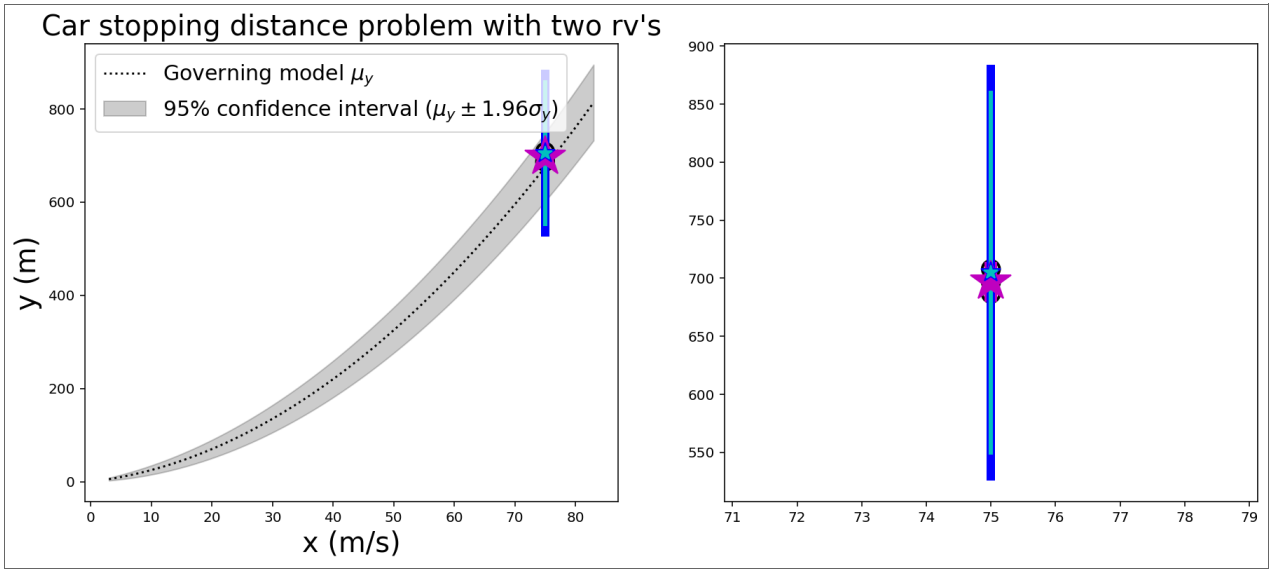
```
Ground truth                    : mean[y] = 675    & std[y] = 37.5
Empirical values (purple)       : mean[y] = 711.38 & std[y] = 22.13
PPD with Uniform Prior (orange): mean[y] = 711.38 & std[y] = 97.98
PPD from MLE (white)            : mean[y] = 711.38 & std[y] = 80.00
```



"Teaser": PPD obtained with the MLE **versus** PPD obtained in Lecture 7 (Gaussian prior)

```
In [13]: MAP_versus_Bayesian_PPD_for_GaussianPrior(N_samples=3)
```

```
Ground truth                  : mean[y] = 675    & std[y] = 37.5
Empirical values (purple)     : mean[y] = 696.92 & std[y] = 8.72
PPD with Gaussian Prior (blue): mean[y] = 704.76 & std[y] = 91.37
PPD from MAP (cyan)           : mean[y] = 704.76 & std[y] = 80.00
```



Final reflection: what strategy should we choose?

Approximating the PPD using a Point estimate is usually much simpler and faster than marginalizing unknown rv's such as z (integrals!).
This is true analytically as well as numerically.

This explains why many ML practitioners choose Point estimates like MLE or MAP.

But in general the predictions of the PPD have different robustness:

- PPD calculated from Posterior distribution > PPD from Point estimates
    - We can also say that within the Point estimates: Posterior mean estimate > MAP > MLE

We will see evidence in favor of this in the remaining of the course.

Final reflection: Bayesian versus non-Bayesian perspective on ML

We can do one last simplification (but it can **mislead** us into believing that ML is not probabilistic!) When the PPD is approximated by the observation distribution for a Point estimate $\hat{z}$, allowing us to predict for new points y^*:

$$p(y^*|\mathcal{D}_y) = p(y^*|z = \hat{z})$$

we can decide to focus on only making a prediction for the **mean** of the PPD and even forget that it is a distribution (we forget uncertainties!).
This is very common in ML literature! But, I think it's advantageous not to think about it that way...

See you next class

Have fun!