CS 189 Introduction to Machine Learning
Fall 2017

Note 4

1 Overview

So far, we've explored a number of ideas under the least squares framework:

- Ordinary Least Least Squares

$$\min_{\vec{w}} \sum_{i=1}^{n} (y_i - \phi(x_i)^T \vec{w})^2 = \|\vec{y} - A\vec{w}\|_2^2 \implies \vec{w}^*_{OLS} = (A^T A)^{-1} A^T \vec{y}$$

- Ridge Regression

$$\min_{\vec{w}} \sum_{i=1}^{n} (y_i - \phi(x_i)^T \vec{w})^2 + \lambda^2 \sum_{j=1}^{d} w_j^2 = \|\vec{y} - A\vec{w}\|_2^2 + \lambda^2 \|\vec{w}\|_2^2 \implies \vec{w}^*_{Ridge} = (A^T A + \lambda^2 I)^{-1} A^T \vec{y}$$

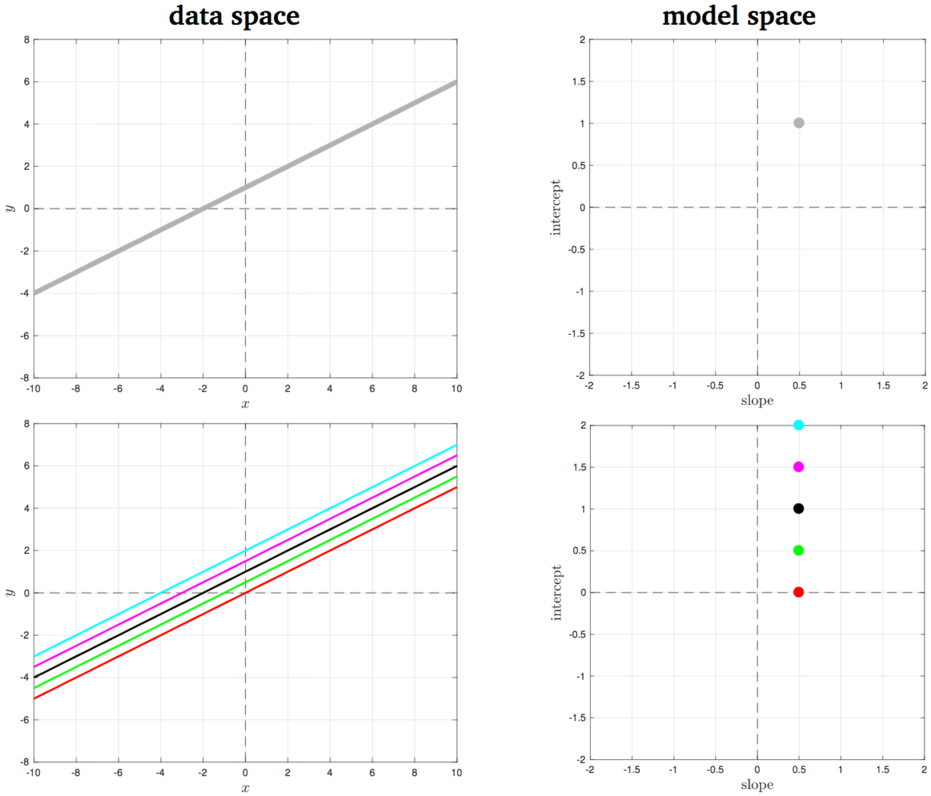
One question that you may have been asking yourself is why we are using the squared error to measure the effectiveness of our model, and why we use the $\ell 2$ norm for the model weights (and not some other norm). We will justify all of these design choices by exploring the statistical interpretations of supervised learning methods and in particular, regression methods. In more concrete terms, we will use a variety of concepts from probability, such as Gaussians, MLE and MAP, in order to validate what we've done so far through a different lens.

2 Optimization in Model Space

Let's look at a simple regression model:

$$f(x) = \text{slope} \cdot x + \text{intercept}$$

Our goal is to find the optimal slope and intercept values that we believe best describe the data. Each arbitrary (slope, intercept) pair forms a line in **data space**. However, it can also be viewed as a single point in **model space**. Learning the optimal model amounts to fitting a line to the data points in the data space; it's equivalent to locating the optimal parameter point in the model space:



3 Data as Samples from Distributions

The data that we observe are random samples with different distributions, coverages, and densities. There are many different distributions that we will encounter, such as **Unifrom**, **Gaussian**, and **Laplacian**.

However, it is arguable that Gaussians distributions are by far the most prevalent. For the purposes of this section we will assume that you already have had exposure to Gaussian distributions before. Let's review their properties for convenience.

random variable: $X \sim p(x)$

probability distribution: $p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

mean: $E[X] = \int_{-\infty}^{+\infty} x p(x) dx = \mu$

variance: $V[X] = E[(X-\mu)^2] = \int_{-\infty}^{+\infty} (x-\mu)^2 p(x) dx = \sigma^2$

parameters: $X \sim \mathcal{N}(\mu, \sigma^2)$

log-likelihood: $\log P(x) = -\frac{(x-\mu)^2}{2\sigma^2} - \log(\sqrt{2\pi}\sigma)$

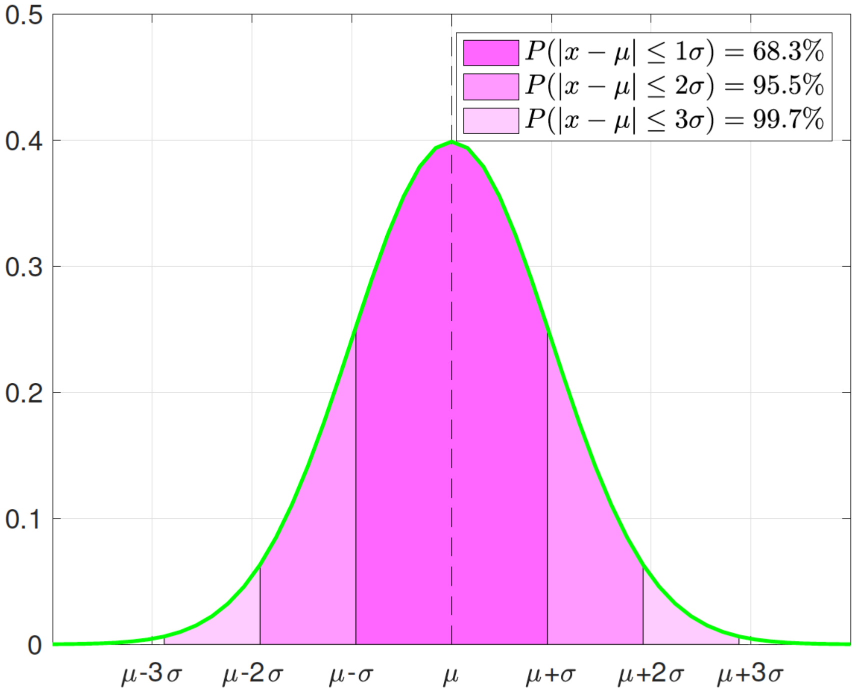
$X \sim \mathcal{N}(\mu_X, \sigma_X^2), \qquad Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$

linear combinations: $E[aX + bY] = a\mu_X + b\mu_Y$

if X, Y are independent: $V[aX + bY] = a^2\sigma_X^2 + b^2\sigma_Y^2$

$aX + bY \sim \mathcal{N}(a\mu_x + b\mu_Y, a^2\sigma_X^2 + b^2\sigma_Y^2)$

Gaussians are particularly appealing because they occur quite frequently in natural settings. Noise distributions in particular are often assumed to be Gaussian distributions, partly because most of noise is captured within 1 or 2 standard deviations of the mean:



4 MLE and MAP for Model Selection

In the context of regression (and all of supervised learning for that matter), we assume a **true underlying model** that maps inputs to outputs. Our goal as machine learning experts is to find a **hypothesis model** that best represents the true underlying model, by using the data we are given.

Let's define more concretely our definition of the data and the model. Our data takes consists of n (x,y) pairs, just as we have seen before:

$$\mathscr{D} = \{(x_i, y_i)\}_{i=1}^n$$

The true underlying model f is a function that maps the inputs x_i to the true outputs $f(x_i)$. Each **observation** y_i is simply a noisy version of $f(x_i)$:

$$y_i = f(x_i) + N_i$$

Note that $f(x_i)$ is a constant, while N_i is a random variable. We always assume that N_i has zero mean, because otherwise there would be systematic bias in our observations. The N_i's could be gaussian, uniform, laplacian, etc.. Here, let us assume that they are **i.i.d** and gaussian: $N_i \overset{\text{i.i.d}}{\sim} \mathscr{N}(0, \sigma^2)$. We can therefore say that $y_i | x_i \sim \mathscr{N}(f(x_i), \sigma^2)$.

Now that we have introduced the data and model, we wish to find a hypothesis model that best describes the data, while possibly taking into account prior beliefs that we have about the true model. We can represent this as a probability problem, where the goal is to find the optimal model that maximizes our probability.

4.1 Maximum Likelihood Estimation

In **Maximum Likelihood Estimation** (MLE), the goal is to find the model that maximizes the probability of the data. If we denote the set of hypothesis models as $\mathscr{H}$, we can represent the problem as:

$$h^*_{MLE} = \underset{h \in \mathscr{H}}{\arg\max}\, P(\text{data} = \mathscr{D} | \text{ true model} = h)$$

More concretely:

$$h^*_{MLE} = \underset{h \in \mathscr{H}}{\arg\max}\, P(y_1, \ldots, y_n | x_1, \ldots, x_n, h)$$

Note that we actually conditioned on the x_i's, because we treat them as *fixed* values of the data. The only randomness in our data comes from the y_i's (since they are noisy versions of the true values $f(x_i)$).

From the chain rule of probability, we can expand the probability statement:

$$P(y_1, \ldots, y_n | x_1, \ldots, x_n, h) = P(y_1 | x_1, \ldots, x_n, h) \cdot P(y_2 | y_1, x_1, x_2 \ldots, x_n, h) \cdot \ldots \cdot P(y_n | y_1, \ldots, y_{n-1}, x_1, \ldots, x_n, h)$$

We can simplify this expression by viewing the problem as a graphical model. Note that y_i only depends on its parent in the graphical model, x_i. It does not depend on the other y_j's, since all y's have independent noise terms. We can therefore simplify the objective:

$$h^*_{MLE} = \underset{h \in \mathscr{H}}{\arg\max} P(y_1, \ldots y_n | x_1, \ldots x_n, h) = \prod_{i=1}^{n} P(y_i | x_i, h)$$

Now let's focus on each individual term $P(y_i|x_i,h)$. We know that $y_i|x_i,h \sim \mathscr{N}(h(x_i), \sigma^2)$, which is cumbersome to work with because gaussians have exponential terms. So instead we wish to work with logs, which eliminate the exponential terms:

$$h^*_{MLE} = \underset{h \in \mathscr{H}}{\arg\max}\, log[P(y_1, \ldots y_n | x_1, \ldots x_n, h)] = \sum_{i=1}^{n} log[P(y_i | x_i, h)]$$

Note that with logs we are still working with the same problem, because logarithms are monotonic functions. Let's try to understand this mathematically through calculus:

$$\frac{d}{dx} log(f(x)) = 0 \iff \frac{1}{f(x)} \frac{df}{dx} = 0 \iff \frac{df}{dx} = 0$$

The last statement is true in this case since $f(x)$ is a gaussian function and thus can never equal 0. Continuing with logs:

$$\begin{aligned}
h^*_{MLE} &= \underset{h \in \mathscr{H}}{\arg\max} \sum_{i=1}^{n} log[P(y_i | x_i, h)] && (1)\\
&= \underset{h \in \mathscr{H}}{\arg\max} -\Big(\sum_{i=1}^{n} \frac{(y_i - h(x_i))^2}{2\sigma^2}\Big) - n \log \sqrt{2\pi}\sigma && (2)\\
&= \underset{h \in \mathscr{H}}{\arg\min} \Big(\sum_{i=1}^{n} \frac{(y_i - h(x_i))^2}{2\sigma^2}\Big) + n \log \sqrt{2\pi}\sigma && (3)\\
&= \underset{h \in \mathscr{H}}{\arg\min} \sum_{i=1}^{n} (y_i - h(x_i))^2 && (4)
\end{aligned}$$

Note that in step (3) we turned the problem from a maximization problem to a minimization problem by negating the objective. In step (4) we eliminated the second term and the denominator in the first term, because they do not depend on the variables we are trying to optimize over.

Now let's look at the case of regression. In that case our hypothesis has the form $h(x_i) = \phi(x_i)^T \vec{w}$, where $\vec{w} \in \mathbb{R}^d$, where d is the number of dimensions of our featurized datapoints. For this specific setting, the problem becomes:

$$\vec{w}^*_{MLE} = \underset{\vec{w} \in \mathbb{R}^d}{\arg\min} \Big(\sum_{i=1}^{n} \frac{(y_i - \phi(x_i)^T \vec{w})^2}{2\sigma^2}\Big)$$

This is just the Ordinary Least Squares (OLS) problem! We just proved that OLS and MLE for regression lead to the same answer! We conclude that MLE is a probabilistic justification for why using squared error (which is the basis of OLS) is a good metric for evaluating a regression model.

4.2 Maximum a Posteriori

In **Maximum a Posteriori** (MAP) Estimation, the goal is to find the model, for which the data maximizes the probability of the model:

$$\begin{aligned}
h^*_{MAP} &= \underset{h\in\mathscr{H}}{\arg\max}\, P(\text{true model} = h|\text{ data} = \mathscr{D}) && (1)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, \frac{P(\text{true model} = h,\text{ data} = \mathscr{D})}{P(\text{ data} = \mathscr{D})} && (2)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, c\cdot P(\text{true model} = h,\text{ data} = \mathscr{D}) && (3)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, P(\text{true model} = h,\text{ data} = \mathscr{D}) && (4)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, P(\text{ data} = \mathscr{D}|\text{true model} = h)\cdot P(\text{true model} = h) && (5)
\end{aligned}$$

Here, we used **Bayes' Rule** to reexpress the objective. In step (3) we represent $P(\text{ data} = \mathscr{D})$ as a constant value because it does not depend on the variables we are optimizing over. Notice that MAP is just like MLE, except we add a term $P(h)$ to our objective. This term is the **prior** over our true model.

More concretely, we have (just as we did with MLE):

$$\begin{aligned}
h^*_{MAP} &= \underset{h\in\mathscr{H}}{\arg\max}\, P(h|y_1,\dots,y_n,x_1,\dots,x_n) && (1)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, \frac{P(h,y_1,\dots,y_n|x_1,\dots,x_n)}{P(y_1,\dots,y_n|x_1,\dots,x_n)} && (2)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, c\cdot P(h,y_1,\dots,y_n|x_1,\dots,x_n) && (3)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, P(h,y_1,\dots,y_n|x_1,\dots,x_n) && (4)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, P(y_1,\dots,y_n|h,x_1,\dots,x_n)\cdot P(h) && (5)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, log[P(y_1,\dots,y_n|h,x_1,\dots,x_n)\cdot P(h)] && (6)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, log[P(y_1,\dots,y_n|h,x_1,\dots,x_n)] + log[P(h)] && (7)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\, \left(\sum_{i=1}^{n} log[P(y_i|x_i,h)]\right) + log[P(h)] && (8)
\end{aligned}$$

Again, just as in MLE, notice that we condition on the x_i's in the whole process because we treat them as constants. Also, let us assume as before that the noise terms are i.i.d and gaussian: $N_i \overset{\text{i.i.d}}{\sim} \mathscr{N}(0,\sigma^2)$. For the prior term $P(h)$, we assume that it follows a shifted and scaled version of the standard Multivariate Gaussian distribution: $h \sim \mathscr{N}(h_0,\sigma_h^2 I)$. Using this specific information, we now have:

$$\begin{aligned}
h^*_{MAP} &= \underset{h\in\mathscr{H}}{\arg\max}\left(\sum_{i=1}^{n} log[P(y_i|x_i,h)]\right) + log[P(h)] &(1)\\
&= \underset{h\in\mathscr{H}}{\arg\max}\left(-\frac{\sum_{i=1}^{n}(y_i-h(x_i))^2}{2\sigma^2}\right)+\left(\frac{-\|h-h_0\|^2}{2\sigma_h^2}\right) &(2)\\
&= \underset{h\in\mathscr{H}}{\arg\min}\left(\frac{\sum_{i=1}^{n}(y_i-h(x_i))^2}{2\sigma^2}\right)+\left(\frac{\|h-h_0\|^2}{2\sigma_h^2}\right) &(3)\\
&= \underset{h\in\mathscr{H}}{\arg\min}\left(\sum_{i=1}^{n}(y_i-h(x_i))^2\right)+\frac{\sigma^2}{\sigma_h^2}\left(\|h-h_0\|^2\right) &(4)
\end{aligned}$$

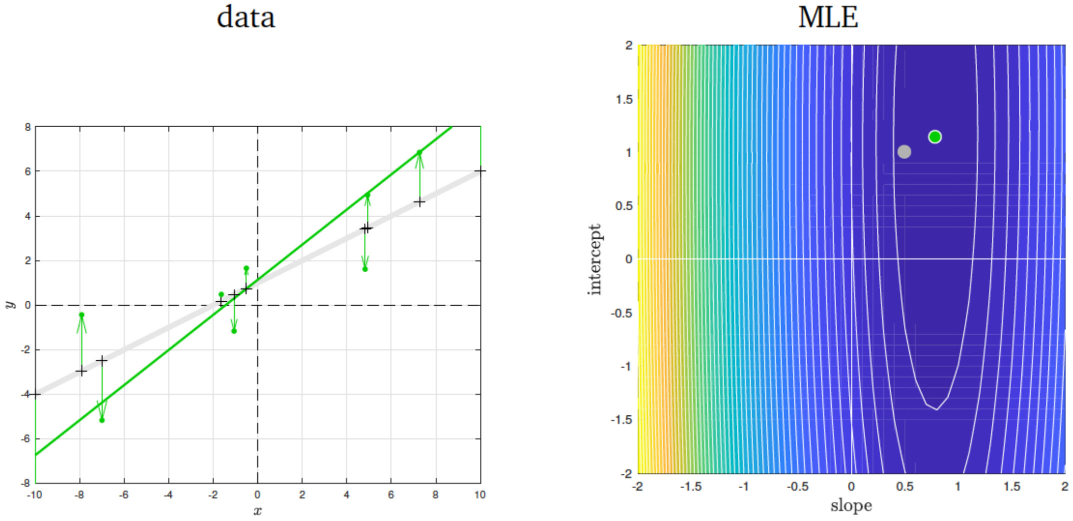
Let's look again at the case for linear regression to illustrate the effect of the prior term when $h_0 = 0$:

$$\vec{w}^*_{MAP} = \underset{\vec{w}\in\mathbb{R}^d}{\arg\min}\left(\sum_{i=1}^{n}(y_i-\phi(x_i)^T\vec{w}))^2\right)+\frac{\sigma^2}{\sigma_h^2}\|\vec{w}\|^2$$

This is just the Ridge Regression problem! We just proved that Ridge Regression and MAP for regression lead to the same answer! We can simply set $\lambda = \frac{\sigma}{\sigma_h}$. We conclude that MAP is a probabilistic justification for adding the penalized ridge term in Ridge Regression.

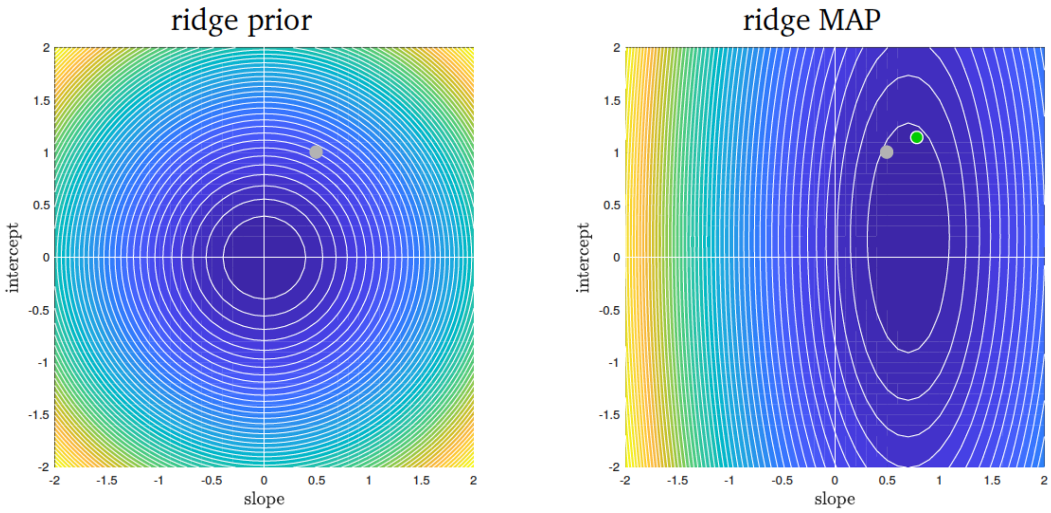
4.3 MLE v. MAP

Based on our analysis of Ordinary Least Squares Regression and Ridge Regression, we should expect to see MAP perform better than MLE. But is that always the case? Let us revisit at the (slope, intercept) example from earlier. We already know the true underlying model parameters, and we will compare them to the values that MLE and MAP select. Let's start with MLE:



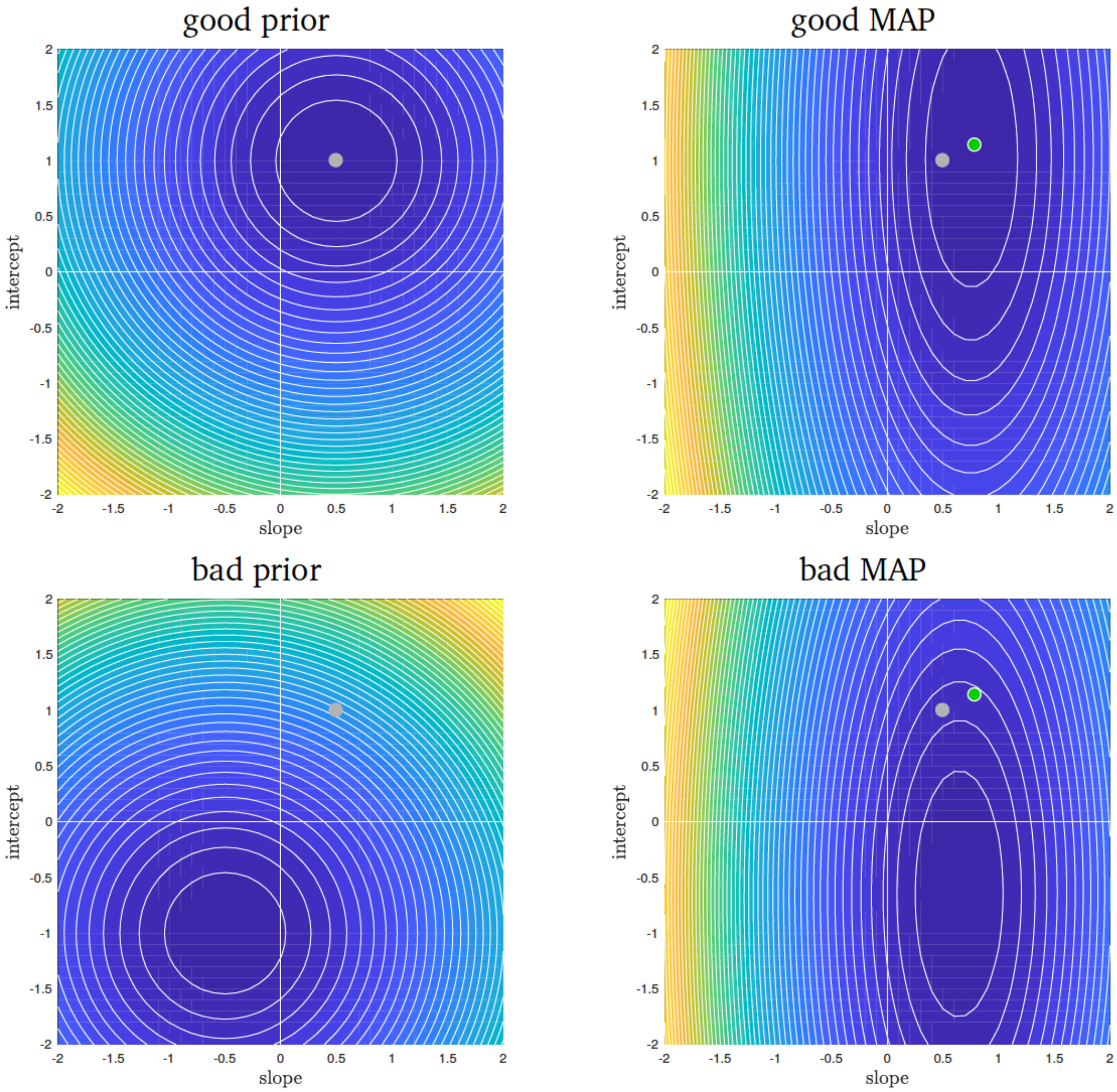
The diagram on the left shows the data space representation of the problem, and the diagram on the right shows the model space representation. The gray line in the left diagram and the gray dot in the right diagram are the true underlying model. Using noisy data samples, MLE predicts a reasonable hypothesis model (as indicated by the green line in the left diagram and the green dot in the right diagram).

Now, let's take a look at the hypothesis model from MAP. One question that arises is where the prior should be centered and what its variance should be. This depends on your belief of what the true underlying model is. If you have reason to believe that the model weights should all be small, then the prior should be centered at zero. Let's look at MAP for a prior that is centered at zero:



For reference, we have marked the MLE estimation from before as a green point and the true model as a gray point. As we can see from the right diagram, using a prior centered at zero leads us to skew our prediction of the model weights toward the origin, leading to a less accurate model than MLE.

Let's say in our case that we have reason to believe that both model weights should be centered around the 0.5 to 1 range. As the first set of diagrams below show, our prediction would be better than MLE. However, if we believe the model weights should be centered around the -0.5 to -1 range, we would make a much poorer prediction than MLE.



As always, in order to compare our beliefs to see which prior works best in practice, we should use cross validation!