Fundamentals of Machine Learning:

Linear Regression, from Geometry to Maximum Likelihood

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

UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Outline

Introduction

Lecture objectives

At the end of this lecture you will:

- Understand the geometric approach to linear regression problems as an application of curve fitting.
- Understand the probabilistic formulation of linear regression and know how the Maximum Likelihood Estimate of model parameters is derived.
- Recognize when underfitting and overfitting of models takes place.
- Understand how regularization can be used to smoothly limit the capacity of our models and mitigate overfitting.

Motivations

- Keep in mind and we work through the lecture today that we are heading towards a probabilistic model.
- We will start, however, with a straightforward geometric model that we will base out intuitions on.
- Interestingly, we will keep coming back to this ”simple” model and verify that our intuitions are solid even from a probabilistic perspective.

Linear Regression as Geometric Curve Fitting

Linear Regression from the ground up

- We begin our study of linear models for regression from a ground-up, geometric perspective.
- We will define the learning problem as one of finding the optimal parameters $\mathbf{w}$ of a parametrized hypothesis space $\mathcal{H}$.
- Optimal will be determined (for now) in terms of a simple, geometric error on a finite sample of observations.

A parametric, linear hypothesis space

- Consider first the class of univariate regression problems where we observe pairs of observations (x_i, t_i), where:
  - $x_i \in \mathbb{R}$ are observations of the independent variable.
  - $t_i \in \mathbb{R}$ are observations of the dependent variable.
- We assume the dependent variable are related to the independent variable via a function f that unknown but linear in its parameters $\mathbf{w}$:

$$
\begin{aligned}
y(x \mid \mathbf{w}) &= w_0 + w_1 x \\
&= \mathbf{w}^T \begin{bmatrix} 1 \\ x \end{bmatrix}
\end{aligned}
$$

Quantifying the goodness of a solution

- OK, we have some data (i.e. observations of the independent and dependent variable).
- And we have a hypothesis space conveniently parameterized by $\mathbf{w}$.
- To actually learn something (i.e. fit the model to the data) we next need some sort of loss $\mathcal{L}$.
- This loss should measure the badness of an arbitrary hypothesis $\mathbf{w}'$ – that is, high loss indicates a bad fit and low loss a good one.
- It will be a function of the candidate parameters and the observed data $\mathcal{D} = \{ (x_i, y_i) \mid i = 1, \ldots N \}$:

$$E(\mathbf{w} \mid \mathcal{D}) = \frac{1}{2} \sum_{i=1}^{N} [y(x_i|\mathbf{w}) - t_i]^2$$

Linear Regression: a Critique of Pure Reason

- We will look in more detail at this choice of this loss.
- But, for now let's consider the properties, good and bad, of this formulation.

$$E(\mathbf{w} \mid \mathcal{D}) = \sum_{i=1}^{N}[(w_0 + w_1 x_i) - t_i]^2$$

Linear Regression and Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) and Least Squares

- Our geometric interpretation is an example of a *point estimate* of model parameters: it gives us a solution $\mathbf{w}^*$ which is a single point in parameter space.
- We would like to move towards a probabilistic model of linear regression.
- This should, ideally:
  - Allow us to reason probabilistically about the quality of our solution.
  - Allow us to quantify belief in the quality of individual predictions.
  - Allow us to ”bake in” prior knowledge about likely solutions.
- Out first step is the Maximum Likelihood Estimate (MLE) of the optimal parameters $\mathbf{w}$ given the observed data $\mathcal{D}$.
- First, let’s beef up our model a bit to allow polynomial functions of the input.

Linear basis function models

- The simplest linear model for regression just uses linear combinations of the input variables $\mathbf{x} = (x_1, \ldots, x_D)^T$:

$$y(\mathbf{x} \mid \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$$

- This is a linear function in both $\mathbf{w}$ (good) and $\mathbf{x}$ (very limiting!).
- So, we can allow linear combinations fixed nonlinear functions of the input:

$$y(\mathbf{x} \mid \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

Linear basis function models

- w_0 is known as a bias parameter and allows for any fixed-offset in the output.
- It is often convenient to define a dummy basis function $\phi_0(\mathbf{x}) = 1$ so we can compactly write:

$$\begin{aligned} y(\mathbf{x} \mid \mathbf{w}) &= \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) \\ &= \mathbf{w}^T \phi(\mathbf{x}) \end{aligned}$$

- Where $\mathbf{w} = (w_0, \ldots, w_{M-1})^T$ and $\phi = (\phi_0, \ldots, \phi_{M-1})^T$
- Common basis functions:

$\phi_i(x) = x^i$	$\phi_i = \exp\{-\frac{(x-\mu_j)^2}{2s^2}\}$	$\phi_i(x) = \tanh(x)$
Polynomial	**Gaussian**	**Sigmoid**

Maximum likelihood and least squares

- Neat, but where are our probabilities in all of this?!
- Let's go back to our original assumption that our target variable is the realization of a deterministic function $y(\mathbf{x}, \mathbf{w})$ with additive Gaussian noise:

$$t = y(\mathbf{x} \mid \mathbf{w}) + \varepsilon$$

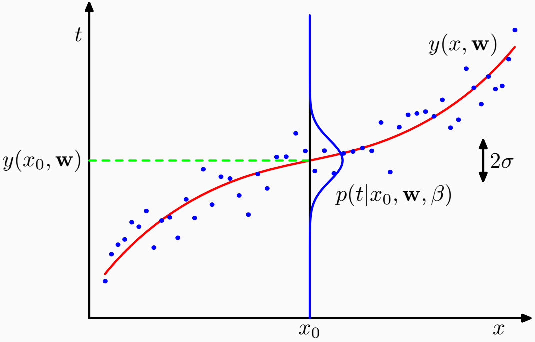
- Where ε is a zero-mean, Gaussian random variable with precision β.
- Thus, we can write:

$$p(t \mid \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(\mathbf{x} \mid \mathbf{w}), \beta^{-1})$$

- Great! A probability distribution! Now let's learn something from data...

Maximum likelihood and least squares

- If you strip away all of the fancy mumbo jumbo, what we are doing is pretty literal and intuitive.
- At any point x_0, our predictor qualifies its prediction by placing a Gaussian around it.



Maximum likelihood and least squares

- Now consider a dataset of inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
- Along with corresponding target values $\mathbf{t} = (t_1, \ldots, t_N)^T$.
- Assuming that these samples are all identically and independently drawn from $p(t \mid \mathbf{x}, \mathbf{w}, \beta)$, we can write:

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

- And we have a (very inconvenient) likelihood expression for our data under a specific model.

Maximum likelihood and least squares

- For most supervised learning problem we are not interested in modeling the distribution of inputs.
- Thus $\mathbf{X}$ will always appear in the conditioning variables and we will omit it for now to unclutter notation.
- Taking logarithms helps simplify the likelihood (thanks in part to the exponential form of the Gaussian):

$$
\begin{aligned}
\ln p(\mathbf{t} \mid \mathbf{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n \mid \mathbf{w}^T \phi(\mathbf{x}_n), \beta^{-1}) \\
&= \frac{N}{2}(\ln \beta - \ln(2\pi)) - \frac{\beta}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2
\end{aligned}
$$

Maximum likelihood and least squares

- Let's maximize this likelihood in $\mathbf{w}$. First the gradient:

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

- And set it to zero:

$$0 = \sum_{n=1}^{N} t_n \phi(\mathbf{x}_n)^T - \mathbf{w}^T \left(\sum_{n=1}^{N} \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right)$$

- Which after solving for $\mathbf{w}$ gives us:

$$\mathbf{w}_{\text{ml}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{t}$$

Maximum likelihood and least squares

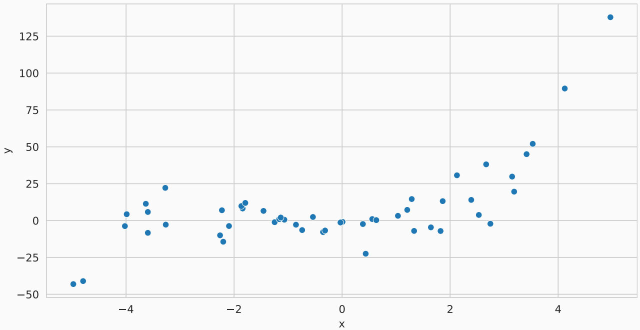
- This solution $(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{t}$ uses the design matrix:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

- An easy way to think of $\boldsymbol{\Phi}$:
    - A row is evaluation of all basis functions on the corresponding training sample (dimensionality M).
    - A column is the corresponding basis function evaluated on all training samples (dimensionality N).
- $\boldsymbol{\Phi}^\dagger \equiv (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T$ is called the Moore-Penrose Psuedo-Inverse.

Regularized least squares

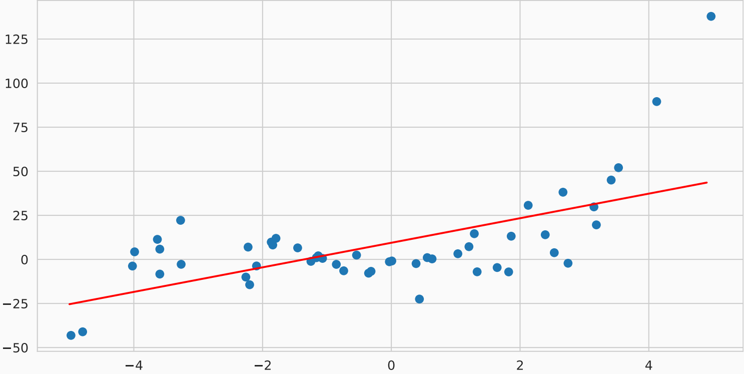
- So, what if we have data distributed like this:



- What might a good basis be? Can we recoup our curve-fitting approach?
- Short answer: Yes! We just saw an example of a polynomial basis!

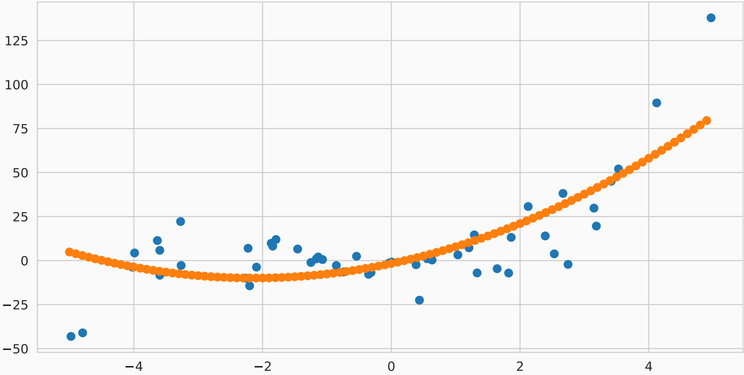
Regularized least squares

- OK, let's go to town:



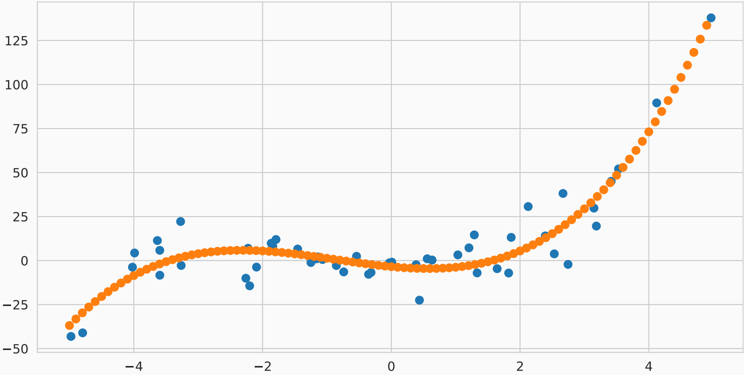
Regularized least squares

- OK, let's go to (quadratic) town:



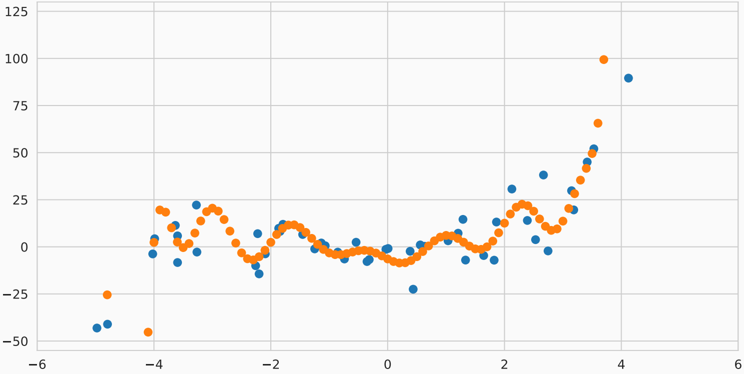
Regularized least squares

- OK, let's go to (cubic) town:



Regularized least squares

- OK, let's really go to (20-degree) town:

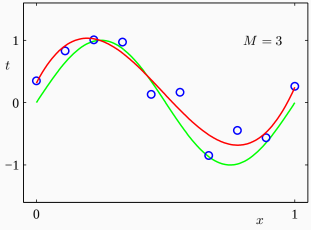


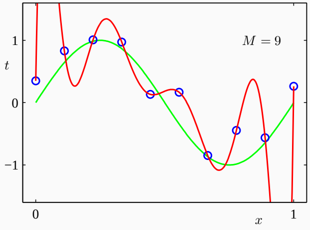
Regularized least squares: over- and under-fitting

- Which is the best result? What does best mean here?
- We see in these plots examples of both overfitting (when M is too large) and underfitting (when M is too small).
- These results might seem counter intuitive: after all, a degree-10 polynomial hypothesis space also contains all polynomials of smaller degrees.
- Why can't doesn't least squares find the right solution if we set $M = 100$ (for example)?
- The answer has to do with model capacity and the fact that the noise in our observations is easily captured by a powerful model.

Regularized least squares

- We can gain some insight by inspecting the magnitude of the parameters estimated by maximizing the likelihood.





	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Regularized least squares

- We will consider regression objectives of this form:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

- Where E_D will continue to be our least squares error with basis functions ϕ.
- The term E_W is called a weight regularizer (or just regularizer), and a very common one is the squared norm of the model weights:

$$E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

- So, the total error function to minimize becomes:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{w})\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

Regularized least squares

- This regularized error function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{w})\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- Is still a quadratic function in $\mathbf{w}$, and the optimal $\mathbf{w}$ is:

$$\mathbf{w}_{\text{ML}} = (\lambda\mathbf{I} + \mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}$$

- The Maximum Likelihood estimate of the β imprecision parameter can also be derived by maximizing $p(t \mid \mathbf{w}, \beta)$ wrt β:

$$\frac{1}{\beta_{\text{ML}}} = \frac{1}{N}\sum_{i=1}^{N}[t_n - \mathbf{w}_{\text{ML}}^T\phi(\mathbf{x}_i)]^2$$

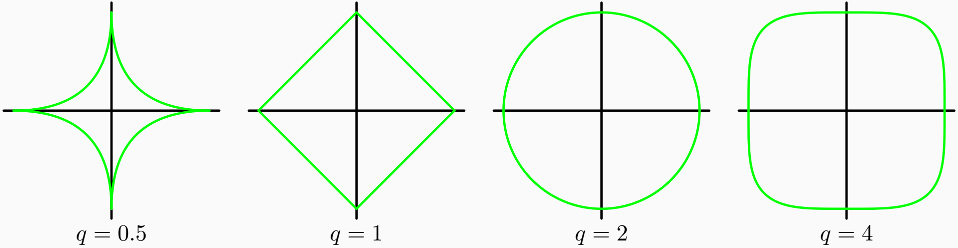
Live Demo!

- Let's take a look at how regularized least squares works in practice.
- [MAKE DEMO GO NOW]

Regularized least squares

- A more general regularized error is:

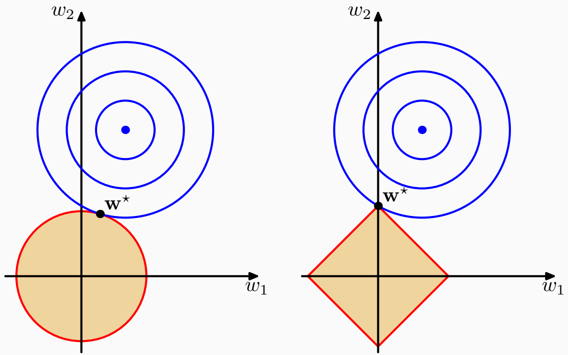
$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Regularized least squares

- A more general regularized error is:

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Concluding remarks

Good Old Linear Regression (tm)

- Today we took a fairly deep look at linear regression.
- Remember that linear is somewhat of misnomer: anything that is linear in the model parameters, is a linear regression.
- In fact, we saw how flexible linear regression with basis functions is.
- So we can fit nonlinear models to data using pure linear regression.
- The Maximum Likelihood solution also has an analytical form, which is just extra nice.
- With this curve fitting model (splines are actually an instance of this) we can fit a lot of phenomena.

The way forward

- The purely geometric and MLE estimators we developed are examples of point estimates.
- They result in a single point in parameter space that is an optimal estimate (for some appropriate definition of ”optimal”).
- But, we cannot bring all of our statistical tools to bear and quantify various types of belief.
- In the next lecture we will develop a Bayesian theory of linear regression which addresses this issue.

Reading and Homework Assignments

Reading Assignment:

- Bishop: Chapter 3 (3.1, 3.2, 3.3)

Homework:

1. How does the scale of our input features affect the linear regression formulations we saw in this lecture? That is, if the scale of our input features and target is $\mathcal{O}(1)$ or $\mathcal{O}(1e6)$, does this make any difference?
2. Does the scale of our input features matter more (or less) if we use regularized least squares?
3. Why would it be problematic if, when using a K-degree polynomial basis for regression, the design matrix $\mathbf{\Phi}$ is rank deficient (i.e. $\text{rank}(\mathbf{\Phi}) < K$)?