Lecture 6
Linear Regression

Luigi Freda

ALCOR Lab
DIAG
University of Rome "La Sapienza"

January 28, 2018
Outline

1 Intro
   • Linear Regression

2 MLE - Least Squares
   • Basic Idea
   • MLE Derivation
   • Geometric Interpretation

3 Convexity
   • Idea
   • Convex Set
   • Convex Function
   • Convexity and Optimization

4 Ridge Regression
   • MLE Issues
   • Basic Idea

5 Regularization Effects of Big Data
   • Idea and Learning Curves
Linear Regression

Model Specification

Linear Regression

\[ y(x) = w^T x + \epsilon = \sum_{j=1}^{D} w_j x_j + \epsilon \]

- \( y \in \mathbb{R} \) and \( x \in \mathbb{R}^D \)
- \( w \in \mathbb{R}^D \) is the weight vector
- \( \epsilon \sim \mathcal{N}(0, \sigma^2) \) is the residual error

This entails

\[ p(y|x, \theta) = \mathcal{N}(\mu(x), \sigma^2) = \mathcal{N}(w^T x, \sigma^2) \]

- \( \mu(x) = w^T x = [w_0, \tilde{w}^T \tilde{x}]^T \) where \( x = [1, \tilde{x}]^T \)
- \( \theta = (w, \sigma^2) \) are the model parameters
Polynomial Regression

if we replace $x$ by a non-linear function $\phi(x)$

$$y(x) = w^T \phi(x) + \epsilon$$

we now have

$$p(y|x, \theta) = \mathcal{N}(w^T \phi(x), \sigma^2)$$

- $\mu(x) = w^T \phi(x)$ (basis function expansion)
- if $x \in \mathbb{R}$ we can use $\phi(x) = [1, x, x^2, \ldots, x^d]$ which is the vector of polynomial basis functions
- in principle, if $x \in \mathbb{R}^D$ we could use a multivariate polynomial expansion
  $$w^T \phi(x) = \sum w_{i_1i_2\ldots i_D} \prod_{j=1}^D x_j^{i_j}$$
  up to a certain degree $d$
- $\theta = (w, \sigma^2)$ are the model parameters

N.B.: note that the model is still linear in the parameters $w$
Linear Regression

2D Data

vertical axis is the temperature, horizontal axes are location within a room

- **left**: fitted function is a plane: \( \hat{f}(x) = w_0 + w_1 x_1 + w_2 x_2 \)
- **right**: fitted function is quadratic: \( \hat{f}(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 \)
Outline

1. Intro
   - Linear Regression

2. MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3. Convexity
   - Idea
   - Convex Set
   - Convex Function
   - Convexity and Optimization

4. Ridge Regression
   - MLE Issues
   - Basic Idea

5. Regularization Effects of Big Data
   - Idea and Learning Curves
the MLE of a parameter $\theta$ is computed as

$$\hat{\theta}_{\text{MLE}} \triangleq \arg \max_\theta p(\mathcal{D}|\theta)$$

given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of iid samples, the log-likelihood can be computed as

$$l(\theta) \triangleq \log p(\mathcal{D}|\theta) = \sum_{i=1}^N \log p(y_i|x_i, \theta)$$

maximizing the log-likelihood is equivalent to minimize the Negative Log-Likelihood (NLL)

$$\text{NLL}(\theta) \triangleq -\sum_{i=1}^N \log p(y_i|x_i, \theta)$$
MLE - Least Squares

- inserting the linear regression model into the log-likelihood returns

\[ l(\theta) = \sum_{i=1}^{N} \log \left[ \left( \frac{1}{2\pi\sigma^2} \right)^{\frac{1}{2}} \exp \left( - \frac{1}{2\sigma^2} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \right) \right] = \]

\[ = - \frac{\text{RSS}(\mathbf{w})}{2\sigma^2} - \frac{N}{2} \log(2\pi\sigma^2) \]

where RSS stands for **Residual Sum of Squares**

\[ \text{RSS}(\mathbf{w}) \triangleq \sum_{i=1}^{N} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \]

aka **Sum of Squared Errors** (SSE), i.e. \( \text{SSE} = \text{RSS} \)

- the Mean Squared Error (MSE) is \( \text{MSE} \triangleq \text{SSE}/N \)
the log-likelihood is
\[ l(\theta) = -\frac{\text{RSS}(w)}{2\sigma^2} - \frac{N}{2} \log(2\pi\sigma^2) \]

if we define the residual errors \( \epsilon_i \triangleq y_i - w^T x_i \), one has
\[ \text{RSS}(w) = \|\epsilon\|_2^2 = \sum_{i=1}^{N} \epsilon_i^2 \]

the MLE for \( w \) minimizes the RSS, and for this reason the method is called least squares
MLE - Least Squares

- **Left**: red circles are training points; blue crosses are approximations.
- In least squares we try to minimize the sum of squared distances from each training point to its approximation (i.e. the sum of the lengths of the little vertical blue lines).
- **Right**: contours of the RSS error; the surface is a bowl with a unique minimum.

Luigi Freda ("La Sapienza" University)
Lecture 6
January 28, 2018 11 / 47
Outline

1. Intro
   - Linear Regression

2. MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3. Convexity
   - Idea
   - Convex Set
   - Convex Function
   - Convexity and Optimization

4. Ridge Regression
   - MLE Issues
   - Basic Idea

5. Regularization Effects of Big Data
   - Idea and Learning Curves
Derivation of the MLE

- The NLL can be rewritten as

\[
NLL(w) = \frac{1}{2} \|y - Xw\|^2 = \frac{1}{2} (y - Xw)^T (y - Xw) = \frac{1}{2} w^T (X^T X) w - w^T (X^T y) + \frac{1}{2} y^T y
\]

where \( y = [y_1, ..., y_N]^T \in \mathbb{R}^N, X \in \mathbb{R}^{N \times D} \) is the design matrix.

- In order to minimize the NLL, we have to compute its gradient

\[
g(w) = (X^T X) w - (X^T y)
\]

and equating it to zero, we get the normal equation

\[
X^T (Xw - y) = 0
\]

- The corresponding solution is called the **Ordinary Least Squares** (OLS)

\[
\hat{w}_{OLS} = (X^T X)^{-1} X^T y
\]

- Considering that \( X^T X = \sum_{i=1}^{N} x_i x_i^T \) and \( X^T y = \sum_i x_i y_i \), one has

\[
\hat{w}_{OLS} = (\sum_{i=1}^{N} x_i x_i^T)^{-1} \sum_i x_i y_i
\]
Outline

1 Intro
   - Linear Regression

2 MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3 Convexity
   - Idea
   - Convex Set
   - Convex Function
   - Convexity and Optimization

4 Ridge Regression
   - MLE Issues
   - Basic Idea

5 Regularization Effects of Big Data
   - Idea and Learning Curves
Geometric Interpretation

Least Squares

- Let's consider the optimization problem we solve

\[ \hat{w}_{OLS} = \underset{w}{\text{argmin}} \frac{1}{2} \| y - Xw \|^2 \]

- This optimization problem is equivalent to solve the normal equation

\[ X^T (Xw - y) = 0 \]

- These equations have an elegant geometric interpretation

- The columns of \( X = [\tilde{x}_1, \ldots, \tilde{x}_D] \in \mathbb{R}^{N \times D} \) define a linear subspace in \( \mathbb{R}^N \)

- N.B.: the columns \( \tilde{x}_j \in \mathbb{R}^N \) of \( X \) are different from the rows \( x_i \in \mathbb{R}^D \) of \( X \)

- The vector \( y \) lives in \( \mathbb{R}^N \)

- The least squares problem above is equivalent to

\[ \underset{\hat{y} \in \text{span}(\{\tilde{x}_1, \ldots, \tilde{x}_D\})}{\text{argmin}} \frac{1}{2} \| y - \hat{y} \|^2 \quad (\hat{y} = Xw = w_1 \tilde{x}_1 + \ldots + w_D \tilde{x}_D) \]
Geometric Interpretation

Least Squares

• let's assume $N = 3$ and $D = 2$

$$X = \begin{bmatrix} 1 & 2 \\ 1 & -2 \\ 1 & 2 \end{bmatrix}, \quad y = \begin{bmatrix} 8.89 \\ 0.61 \\ 1.77 \end{bmatrix}$$
Geometric Interpretation
Least Squares

- we seek a vector $\hat{y} \in \mathbb{R}^N$ that lies in the linear subspace defined by the columns of $X$ and is close as possible to $y \in \mathbb{R}^N$

$$\argmin_{\hat{y} \in \text{span}\{\tilde{x}_1, ..., \tilde{x}_D\}} \frac{1}{2} \|y - \hat{y}\|^2$$

$(\hat{y} = Xw = w_1\tilde{x}_1 + ... + w_D\tilde{x}_D)$
Geometric Interpretation

Least Squares

- if we want to minimize the residual \( \argmin_{\hat{y} \in \text{span}(\tilde{x}_1, ..., \tilde{x}_D)} \frac{1}{2} ||y - \hat{y}||^2 \) then we also want the residual vector \( y - \hat{y} \) to be orthogonal to the linear space (hyperplane) defined by the columns of \( X \)

\[
\tilde{x}_j^T(\hat{y} - y) = 0 \quad \text{for} \quad j \in \{1, ..., D\}
\]
hence, solving \[ \argmin_{\hat{y} \in \text{span}(\tilde{x}_1, \ldots, \tilde{x}_D)} \frac{1}{2} \| y - \hat{y} \|^2 \] is equivalent to solve the normal equation

\[ \tilde{x}_j^T (\hat{y} - y) = 0 \text{ for } j \in \{1, \ldots, D\} \implies X^T (\hat{y} - y) = X^T (Xw - y) = 0 \]
solving $X^T(Xw - y) = 0$ returns $\hat{w} = (X^TX)^{-1}X^Ty$

we can get the **orthogonal projection** $\hat{y} = X\hat{w} = X(X^TX)^{-1}X^Ty$

the projection matrix $P \triangleq X(X^TX)^{-1}X^T$ is called the **hat matrix**, since it "puts the hat" on $y$
Outline

1 Intro
   ■ Linear Regression

2 MLE - Least Squares
   ■ Basic Idea
   ■ MLE Derivation
   ■ Geometric Interpretation

3 Convexity
   ■ Idea
   ■ Convex Set
   ■ Convex Function
   ■ Convexity and Optimization

4 Ridge Regression
   ■ MLE Issues
   ■ Basic Idea

5 Regularization Effects of Big Data
   ■ Idea and Learning Curves
when solving least squares, we noted that the NLL had a bowl shape with a unique minimum

- the technical term for functions like this is convex
- convex functions play a very important role in machine learning
Outline

1. Intro
   - Linear Regression

2. MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3. Convexity
   - Idea
   - Convex Set
     - Convex Function
     - Convexity and Optimization

4. Ridge Regression
   - MLE Issues
   - Basic Idea

5. Regularization Effects of Big Data
   - Idea and Learning Curves
a set $S$ is **convex** if for any $\theta, \theta' \in S$ we have

$$\lambda \theta + (1 - \lambda) \theta' \in S \quad \forall \lambda \in [0, 1]$$

that is, if we draw a line from $\theta$ to $\theta'$, all points on the line lie inside the set $S$.

for instance in this figure we have a convex set on the left and a nonconvex set on the right.
Outline

1 Intro
   - Linear Regression

2 MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3 Convexity
   - Idea
   - Convex Set
   - Convex Function
   - Convexity and Optimization

4 Ridge Regression
   - MLE Issues
   - Basic Idea

5 Regularization Effects of Big Data
   - Idea and Learning Curves
Convexity
Convex Function Definition

- A function $f(\theta)$ is **convex** if it is defined on a convex set and if, for any $\theta, \theta' \in S$, and for any $\lambda \in [0, 1]$, we have

  $$f(\lambda \theta + (1 - \lambda) \theta') \leq \lambda f(\theta) + (1 - \lambda) f(\theta')$$

- A function $f(\theta)$ is **strictly convex** if the inequality is strict.

- A function $f(\theta)$ is **concave** if $-f(\theta)$ is convex.

- For instance, in the following figure:

  - The function on the left is convex and the function on the right is neither concave nor convex.
Convexity
Convex Function Definition

- examples of scalar convex functions include $\theta^2$, $e^\theta$ and $\theta \log \theta$ (for $\theta > 0$)
- examples of concave functions include $\log \theta$ and $\sqrt{\theta}$

Intuitively, a (strictly) convex function has a *bowl shape*, and hence has a **unique global minimum** $\theta^*$ corresponding to the bottom of the bowl.

In the scalar case, a convex function has its second derivative positive everywhere, i.e. $\frac{d^2}{d\theta^2} f(\theta) > 0$
Convex (concave) functions have a unique global minimum (maximum).
Outline

1. Intro
   - Linear Regression

2. MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3. Convexity
   - Idea
   - Convex Set
   - Convex Function
   - Convexity and Optimization

4. Ridge Regression
   - MLE Issues
   - Basic Idea

5. Regularization Effects of Big Data
   - Idea and Learning Curves
Convexity
Convex Function

Theorem 1

A twice-continuously differentiable, multivariate function $f(\theta) \in \mathbb{R}$ is convex if and only if its Hessian is positive definite for all $\theta$.

- The Hessian matrix $H = \frac{\partial^2 f(\theta)}{\partial \theta^2}$ of a function $f(\theta) \in \mathbb{R}$ is defined as follows (element-wise):
  \[ H_{jk} = \frac{\partial^2 f(\theta)}{\partial \theta_j \partial \theta_k} \]

- The Hessian is symmetric since $\frac{\partial^2 f(\theta)}{\partial \theta_j \partial \theta_k} = \frac{\partial^2 f(\theta)}{\partial \theta_k \partial \theta_j}$ (Schwartz’s theorem).

- Recall that a matrix $H$ is positive definite if $v^T H v > 0$ for any $v \neq 0$.

- A convex function has a bowl shape.
a convex function can be approximated about its unique global minimum $\theta^*$ with a bowl shaped quadratic function (paraboloid in 3D)

$$f(\theta) \approx f(\theta^*) + \left. \frac{\partial f}{\partial \theta} \right|_{\theta^*} (\theta - \theta^*) + \frac{1}{2} (\theta - \theta^*)^T H(\theta^*)(\theta - \theta^*)$$

$$= f(\theta^*) + \frac{1}{2} (\theta - \theta^*)^T H(\theta^*)(\theta - \theta^*)$$

where we used the fact that at the minimum $\theta^*$ one has $\frac{\partial f}{\partial \theta} |_{\theta^*} = 0$ and we know that $H(\theta^*) > 0$
one problem with ML estimation is that it can result in **overfitting**
the reason that the MLE can overfit is that it is picking the parameter values that are the best for modeling the **training data**
but if the data is **noisy**, such parameters often result in complex functions
as a simple example, suppose we fit a degree 14 polynomial to $N = 21$ data points using least squares. The resulting curve is very ”wiggly”
in this case, overfitting equals interpolating noise
if we change the data a little bit our estimate of $w$ will change a lot (unstable estimation)
the corresponding least squares coefficients $\mathbf{w}$ (excluding $w_0$) are as follows

\begin{align*}
8524.540, & 6607.897, -12640.058, -5530.188, 9479.730, 1774.639, -2821.526
\end{align*}

- there are many large positive and negative numbers
- these variations balance out exactly to make the curve “wiggle” in just the right way so that it almost perfectly interpolates the data
1. Intro
   - Linear Regression

2. MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3. Convexity
   - Idea
   - Convex Set
   - Convex Function
   - Convexity and Optimization

4. Ridge Regression
   - MLE Issues
   - Basic Idea

5. Regularization Effects of Big Data
   - Idea and Learning Curves
Ridge Regression

- we can encourage the parameters to be small, thus resulting in a smoother curve, by using a zero-mean Gaussian prior

\[ p(w) = \prod_j \mathcal{N}(w_j | 0, \tau^2) \]

where the precision \(1/\tau^2\) controls the strength of the prior

- the corresponding MAP estimation problem becomes

\[ \hat{w}_{MAP} = \arg\max_w p(w|D) = \arg\max_w p(D|w)p(w) = \arg\max_w \log p(D|w) + \log p(w) = \]

\[ \arg\max_w \sum_{i=1}^N \log \mathcal{N}(y_i | w_0 + w^T x_i, \sigma^2) + \sum_{i=1}^D \log \mathcal{N}(w_j | 0, \tau^2) \]

- it is simple to show that this is equivalent to minimize

\[ J(w) = \frac{1}{N} \sum_{i=1}^N (y_i - (w_0 + w^T x_i)^2) + \lambda \|w\|_2^2 = \|y - Xw\|_2^2 + \lambda \|w\|_2^2 \]

where \(\lambda \triangleq \sigma^2 / \tau^2\) and \(\|w\|_2^2 = w^T w\)

- the first term is the MSE/NLL as usual, and the second term with \(\lambda > 0\) is a complexity penalty
Ridge Regression

- minimization problem

\[ J(w) = \frac{1}{N} \sum_{i=1}^{N} (y_i - (w_0 + w^T x_i)^2) + \lambda \|w\|_2^2 = \|y - Xw\|_2^2 + \lambda \|w\|_2^2 \]

- the solution is

\[ \hat{w}_{ridge} = (\lambda I_D + X^T X)^{-1} X^T y \]

- this technique is known as ridge regression
Outline

1. Intro
   - Linear Regression

2. MLE - Least Squares
   - Basic Idea
   - MLE Derivation
   - Geometric Interpretation

3. Convexity
   - Idea
   - Convex Set
   - Convex Function
   - Convexity and Optimization

4. Ridge Regression
   - MLE Issues
   - Basic Idea

5. Regularization Effects of Big Data
   - Idea and Learning Curves
Regularization Effects of Big Data

- **regularization** is the most common way to avoid **overfitting**
- another effective approach — which is not always available — is to use **lots of data**
- intuitively, the more training data we have, the better we will be able to learn
- let’s consider a **polynomial regression** and let’s plot the Mean Squared Error (MSE) incurred on the test set achieved by models of different degrees vs $N = \mid \mathcal{D} \mid$
- a plot of error vs training set size is known as a **learning curve**
Regularization Effects of Big Data

(a) truth=degree 2, model = degree 1

(b) truth=degree 2, model = degree 2

(c) truth=degree 2, model = degree 10

(d) truth=degree 2, model = degree 25
- the truth is a degree 2 polynomial
- we try fitting polynomials of degrees 1, 2, 10 and 25 to this data (respectively models $M_1, M_2, M_{10}$ and $M_{25}$)
the level of the plateau for the test error consists of two terms

1. **noise floor**: an irreducible component that all models incur, due to the intrinsic variability of the generating process

2. **structural error**: a component that depends on the discrepancy between the generating process (the “truth”) and the model
Regularization Effects of Big Data

- the structural error for models $M_2$, $M_{10}$ and $M_{25}$ is zero, since they are all able to capture the true generating process.
- the structural error for $M_1$ is substantial, which is evident from the fact that the plateau occurs high above the noise floor.
for any model that is expressive enough to capture the truth (i.e., one with small structural error), the **test error** will go to the **noise floor** as $N \to \infty$

- the error will typically go to zero faster for simpler models (fewer parameters to estimate)
for finite training sets, there will be some discrepancy between the parameters that we estimate and the best parameters that we could estimate given the particular model class.

- this discrepancy is called **approximation error**, and goes to zero as \( N \to \infty \), but it goes to zero faster for simpler models.
in domains with lots of data, **simple methods** can work surprisingly well

however, there are still reasons to study more **sophisticated learning methods**, because there will always be problems for which we have **little data**

for example, even in such a data-rich domain as web search, as soon as we want to start **personalizing the results**, the amount of data available for any given user starts to look small again (relative to the complexity of the problem)
Kevin Murphy’s book