Foundation of Intelligent Systems, Part I

Regression 2

mcuturi@i.kyoto-u.ac.jp
Some Words on the Survey, 15 Answers

• 7 better theoretical understanding
• 6 applications
• 5 introduction
• 3 advanced

How often do you program?

- very rarely
- I programmed a few times, only to complete assignments for other courses
- I spend about 2-3 hours programming every month
- I spend about 2-3 hours programming every week
- I program every day

• 7/15 every day, 4/15 once in a while, 2 little, 2 very rarely.
Some Words on the Survey, 15 Answers

- **AVG < 2**: VC dimension, Markov Inequality, QP, CG, Empirical Risk
- **2 < AVG < 3**: Jensen, KL div, LP, Simplex, Lagrangean, Psd-ness
- **AVG > 3**: Eigendecomposition, Matrix Inv, CLT, Probability Space, Expectation, Gaussian density

For any questions on derivatives/gradient/convexity check
- Convex Optimization, Boyd Vandenberghe

*free online*. you can also check CVX the matlab optimization package.
Last Week

- **Regression**: relationship between **predictors** and **predicted variables**.

- **in 2D**: Least-Squares Criterion \( L(b, a_1, \cdots, a_p) \) to fit **lines**, polynomials.
  - results in solving a linear system.

\[
\frac{\partial \text{2nd order}(b, a_1, \cdots, a_p)}{\partial a_p} = \text{linear in } (b, a_1, \cdots, a_p)
\]

  - \( p + 1 \) equations, \( p + 1 \) variables.

- **in \( \mathbb{R}^d \)**, find best fit \( \alpha \in \mathbb{R}^n \) such that \( (\alpha^T x + \alpha_0) \approx y \)
  - The Least-Squares criterion also applies:

\[
L(\alpha) = \|Y - \alpha^T X\|^2 = (\alpha^T X X^T \alpha - 2Y X^T \alpha + \|Y\|^2).
\]

\[
\nabla_\alpha L = 0 \implies \alpha^* = (X X^T)^{-1} X Y^T
\]

- This works if \( X X^T \in \mathbb{R}^{d+1} \) is **invertible**.
Last Week

\[
\begin{align*}
\text{ans} = & -0.049332605603095 \times \text{age} \\
& + 0.163122792160298 \times \text{surface} \\
& - 0.004411580036614 \times \text{distance} \\
& + 2.731204399433800 \text{ JPY}
\end{align*}
\]
Today

- A few words on the statistical / probabilistic perspective on LS-regression
- A few words on polynomials in higher dimensions
- A geometric perspective
- A practical perspective
- Some solutions: advanced regression techniques
  - Subset selection
  - Ridge Regression
  - Lasso... next time
A (very few) words on the statistical/probabilistic interpretation of LS
The Statistical Perspective on Regression

- **Assume that** the values of $y$ are stochastically linked to observations $x$ as

$$y - (\alpha^T x + \beta) \sim \mathcal{N}(0, \sigma).$$

- This difference is a random variable called $\varepsilon$ and is called a residue.

- This can be rewritten as,

$$y = (\alpha^T x + \beta) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma),$$

- We **assume** that the difference between $y$ and $(\alpha^T x + b)$ behaves like a Gaussian (normally distributed) random variable.

- **Objective:** Identify good candidates for $\alpha$ and $\beta$.

**Estimate** $\alpha$ and $\beta$ given observations.
Identically Independently Distributed (i.i.d) Observations

- Classic statistical methodology is to compute the **probability** of different observations, **assuming that the parameters are** $\alpha = a, \beta = b$:
  
  - For each couple $(x_j, y_j), j = 1, \ldots, N$,
    
    $$P(x_j, y_j | \alpha = a, \beta = b) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{\|y_j - (a^T x_j + b)\|^2}{2\sigma^2} \right)$$

  - Since each measurement $(x_j, y_j)$ has been **independently sampled**,
    
    $$P \left( \{ (x_j, y_j) \}_{j=1}^{N} | \alpha = a, \beta = b \right) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{\|y_j - (a^T x_j + b)\|^2}{2\sigma^2} \right)$$

  - A.K.A **likelihood** of the dataset $\{(x_j, y_j)_{j=1,\ldots,N}\}$ as a function of $a$ and $b$,
    
    $$\mathcal{L}_{\{(x_j, y_j)\}}(a, b) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{\|y_j - (a^T x_j + b)\|^2}{2\sigma^2} \right)$$
Maximum Likelihood Estimation (MLE) of Parameters

Given the **likelihood** function on the dataset \( \{(x_j, y_j)_{j=1,\ldots,N}\} \):  

\[
L(a, b) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{\|y_j - (a^T x_j + b)\|^2}{2\sigma^2} \right)
\]

...the **MLE** approach selects the values of \((a, b)\) which **maximize** \(L(a, b)\)

- **However**, \(\max_{(a,b)} L(a, b) \iff \max_{(a,b)} \log L(a, b)\)

\[
\log L(a, b) = C - \frac{1}{2\sigma^2} \sum_{j=1}^{N} \|y_j - (a^T x_j + b)\|^2.
\]

- **Hence** \(\max_{(a,b)} L(a, b) \iff \min_{(a,b)} \sum_{j=1}^{N} \|y_j - (a^T x_j + b)\|^2\)...
Statistical Approach to Linear Regression

• Properties of the MLE estimator: convergence of $\|\alpha - a\|$ for instance?

• Confidence intervals for coefficients,

• Tests procedures to assess if model “fits” the data,

• Bayesian approaches: instead of looking for one optimal fit $(a, b)$ juggle with a whole density on $(a, b)$ to make decisions

• etc.
Very few words on polynomials in higher dimensions
Very few words on polynomials in higher dimensions

- For $d$ variables, that is for points $\mathbf{x} \in \mathbb{R}^d$,
  - the space of polynomials on these variables up to degree $p$ is generated by
    \[
    \{ \mathbf{x}^\mathbf{u} | \mathbf{u} \in \mathbb{N}^d, \mathbf{u} = (u_1, \cdots, u_d), \sum_{i=1}^{d} u_i \leq p \}
    \]
    where the monomial $\mathbf{x}^\mathbf{u}$ is defined as $x_1^{u_1} x_2^{u_2} \cdots x_d^{u_d}$
  - Recurrence for dimension of that space: $\dim_{p+1} = \dim_p + \binom{p+1}{d+p}$

- For $d = 20$ and $p = 5$, $1 + 20 + 210 + 1540 + 8855 + 42504 > 50,000$

Problem with polynomial interpolation in high-dimensions is the explosion of relevant variables (one for each monomial)
Geometric Perspective
Recall the problem:

\[ X = \begin{bmatrix}
    1 & 1 & \cdots & 1 \\
    \vdots & \vdots & \ddots & \vdots \\
    x_1 & x_2 & \cdots & x_N \\
    \vdots & \vdots & \ddots & \vdots 
\end{bmatrix} \in \mathbb{R}^{d+1 \times N} \]

and

\[ Y = [y_1 \cdots y_N] \in \mathbb{R}^N. \]

We look for \( \alpha \) such that \( \alpha^T X \approx Y \).
Back to Basics

- If we transpose this expression we get $X^T \alpha \approx Y^T$,

$$
\begin{bmatrix}
1 & x_{1,1} & \cdots & x_{d,1} \\
1 & x_{1,2} & \cdots & x_{d,2} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{1,k} & \cdots & x_{d,k} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{1,N} & \cdots & x_{d,N}
\end{bmatrix} \times
\begin{bmatrix}
\alpha_0 \\
\vdots \\
\alpha_d
\end{bmatrix} =
\begin{bmatrix}
y_1 \\
\vdots \\
y_N
\end{bmatrix}
$$

- Using the notation $Y = Y^T$, $X = X^T$ and $X_k$ for the $(k + 1)^{th}$ column of $X$,

$$
\sum_{k=0}^{d} \alpha_k X_k \approx Y
$$

- Note how the $X_k$ corresponds to all values taken by the $k^{th}$ variable.

- **Problem**: approximate/reconstruct Reconstructing $Y \in \mathbb{R}^N$ using $X_0, X_1, \cdots, X_d \in \mathbb{R}^N$?
Linear System

Reconstructing $\mathbf{Y} \in \mathbb{R}^N$ using $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $\mathbf{Y}$ depends implicitly on the space spanned by $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$.

Consider the observed vector in $\mathbb{R}^N$ of predicted values.
Linear System

Reconstructing $\mathbf{Y} \in \mathbb{R}^N$ using $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $\mathbf{Y}$ depends implicitly on the space spanned by $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$

Plot the first regressor $\mathbf{X}_0$...
Reconstructing $\mathbf{Y} \in \mathbb{R}^N$ using $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $\mathbf{Y}$ depends implicitly on the space spanned by $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$

Assume the next regressor $\mathbf{X}_1$ is colinear to $\mathbf{X}_0$...
Linear System

Reconstructing $Y \in \mathbb{R}^N$ using $X_0, X_1, \cdots, X_d$ vectors of $\mathbb{R}^N$.

• Our ability to approximate $Y$ depends implicitly on the space spanned by $X_0, X_1, \cdots, X_d$.

and so is $X_2$...
Linear System

Reconstructing $\mathbf{Y} \in \mathbb{R}^N$ using $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $\mathbf{Y}$ depends implicitly on the space spanned by $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_d$

Very little choices to approximate $\mathbf{Y}$...
Reconstructing $Y \in \mathbb{R}^N$ using $X_0, X_1, \cdots, X_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $Y$ depends implicitly on the space spanned by $X_0, X_1, \cdots, X_d$.

Suppose $X_2$ is actually not colinear to $X_0$. 
Reconstructing $Y \in \mathbb{R}^N$ using $X_0, X_1, \cdots, X_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $Y$ depends implicitly on the space spanned by $X_0, X_1, \cdots, X_d$.

This opens new ways to reconstruct $Y$. 
Reconstructing $\mathbf{Y} \in \mathbb{R}^N$ using $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $\mathbf{Y}$ depends implicitly on the space spanned by $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$

When $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2$ are linearly independent,
Reconstructing $\mathbf{Y} \in \mathbb{R}^N$ using $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $\mathbf{Y}$ depends implicitly on the space spanned by $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$

$\mathbf{Y}$ is in their span since the space is of dimension 3
Linear System

Reconstructing $\mathbf{Y} \in \mathbb{R}^N$ using $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$ vectors of $\mathbb{R}^N$.

- Our ability to approximate $\mathbf{Y}$ depends implicitly on the space spanned by $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d$

The dimension of that space is $\text{Rank}(\mathbf{X})$, the rank of $\mathbf{X}$

$\text{Rank}(\mathbf{X}) \leq \min(d + 1, N)$. 
Linear System

Three cases depending on $\text{Rank } X$ and $d, N$

1. $\text{Rank } X < N$. $d + 1$ column vectors do not span $\mathbb{R}^N$
   - For arbitrary $Y$, there is no solution to $\alpha^T X = Y$

2. $\text{Rank } X = N$ and $d + 1 > N$, too many variables span the whole of $\mathbb{R}^N$
   - infinite number of solutions to $\alpha^T X = Y$

3. $\text{Rank } X = N$ and $d + 1 = N$, # variables = # observations
   - Exact and unique solution: $\alpha = X^{-1}Y$ we have $\alpha^T X = Y$

In most applications, $d + 1 \neq N$ so we are either in case 1 or 2
Case 1: Rank $X < N$

- **no solution** to $\alpha^T X = Y$ (equivalently $X\alpha = Y$) in general case.
- What about the **orthogonal projection** of $Y$ on the image of $X$

- Namely the point $\hat{Y}$ such that

$$\hat{Y} = \arg\min_{u \in \text{span } \{X_0, X_1, \ldots, X_d\}} \|Y - u\|.$$
Case 1: Rank $X < N$

**Lemma 1.** \( \{X_0, X_1, \cdots, X_d\} \) is a l.i. family \( \iff X^T X \) is invertible
Case 1: Rank $X < N$

- Computing the projection $\hat{w}$ of a point $w$ on a subspace $V$ is well understood.
- In particular, if $(X_0, X_1, \cdots, X_d)$ is a basis of $\text{span}\{X_0, X_1, \cdots, X_d\}$...

  (that is $\{X_0, X_1, \cdots, X_d\}$ is a linearly independent family)

  ... then $(X^TX)$ is invertible and ...

  $$\hat{Y} = X(X^TX)^{-1}X^TY$$

- This gives us the $\alpha$ vector of weights we are looking for:

  $$\hat{Y} = X \underbrace{(X^TX)^{-1}X^T}_\alpha Y = X\hat{\alpha} \approx Y \text{ or } \hat{\alpha}^TX = Y$$

- What can go wrong?
Case 1: Rank $X < N$

- If $X^T X$ is invertible,
  \[
  \hat{Y} = X(X^T X)^{-1}X^T Y
  \]

- If $X^T X$ is not invertible... we have a problem.

- If $X^T X$'s condition number
  \[
  \frac{\lambda_{\text{max}}(X^T X)}{\lambda_{\text{min}}(X^T X)},
  \]
  is very large, a small change in $Y$ can cause dramatic changes in $\alpha$.

- In this case the linear system is said to be **badly conditioned**...

- Using the formula
  \[
  \hat{Y} = X(X^T X)^{-1}X^T Y
  \]
  might return garbage as can be seen in the following Matlab example.
Case 2: \textbf{Rank }\textbf{X} = N \textbf{ and } d + 1 > N

\textbf{high-dimensional low-sample setting}

- \textbf{Ill-posed inverse problem}, the set

\[ \{ \alpha \in \mathbb{R}^d \mid \mathbf{X}\alpha = \mathbf{Y} \} \]

is a whole \textbf{vector space}. We need to choose one from many admissible points.

- When does this happen?
  - High-dimensional low-sample case (DNA chips, multimedia \textit{etc.})

- How to solve for this?
  - Use something called regularization.
A practical perspective: Overfitting and Interpretability
A Few High-dimensions Low sample settings

- DNA chips are very long vectors of measurements, one for each gene

- Task: regress a Cancer related variable w.r.t these genes
A Few High-dimensions Low sample settings

- Emails represented as histograms of words $\text{email}_j = \left[ \begin{array}{c} \#\{\text{please}\} \\ \vdots \end{array} \right]$

- Task: regress a spam related variable (e.g. how many users classified this as spam) w.r.t these variables

Source of this image
Correlated Variables

- Suppose you run a real-estate.
  - For each apartment you have compiled a few hundred variables, e.g.
    - distances to conv. store, pharmacy, supermarket, parking lot, etc.
    - distances to all main locations in Kansai
    - socio-economic variables of the neighborhood
  - Some are obviously correlated (correlated = “almost” colinear)
  - We will run into some issues (Matlab example)
Overfitting

- Given \( d \) variables (including constant variable), consider the least squares criterion

\[
L_d (\alpha_1, \cdots, \alpha_d) = \sum_{i=1}^{j} \left\| y_j - \sum_{i=1}^{d} \alpha_i x_{i,j} \right\|^2
\]

- Add any variable vector \( x_{d+1,j} \), \( j = 1, \cdots, N \), and define

\[
L_{d+1} (\alpha_1, \cdots, \alpha_d, \alpha_{d+1}) = \sum_{i=1}^{j} \left\| y_j - \sum_{i=1}^{d} \alpha_i x_{i,j} - \alpha_{d+1} x_{d+1,j} \right\|^2
\]

Then \( \min_{\mathbb{R}^{d+1}} L_{d+1} \leq \min_{\mathbb{R}^d} L_d \)

- Focusing exclusively on the RSS is a poor choice. (Matlab example)
Occam's razor formalization of overfitting

- **Occam’s razor**: *lex parsimoniae*

- **Law of parsimony**: principle that recommends selecting the hypothesis that makes the fewest assumptions.
Advanced Regression Techniques
Quick Reminder on Vector Norms

- For a vector $\mathbf{a} \in \mathbb{R}^d$, the Euclidian norm is the quantity

$$\|\mathbf{a}\| = \|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^{d} a_i^2}.$$

- More generally, the $q$-norm is for $q > 0$,

$$\|\mathbf{a}\|_q = \left(\sum_{i=1}^{d} |a_i|^q\right)^{\frac{1}{q}}.$$

- In particular for $q = 1$,

$$\|\mathbf{a}\|_1 = \sum_{i=1}^{d} |a_i|.$$

- In the limit $q \to \infty$ and $q \to 0$,

$$\|\mathbf{a}\|_\infty = \max_{i=1,\ldots,d} |a_i|, \quad \|\mathbf{a}\|_0 = \#\{i | a_i \neq 0\}.$$
Tikhonov Regularization ’43 - Ridge Regression ’62

• Tikhonov’s motivation: solve **ill-posed inverse problems** by **regularization**

• If $\min_\alpha L(\alpha)$ is achieved on many points... consider

$$\min_\alpha L(\alpha) + \lambda \| \alpha \|^2$$

• We can show that this leads to selecting

$$\hat{\alpha} = (X^T X + \lambda I_{d+1})^{-1}XY$$

• The condition number has changed to

$$\frac{\lambda_{\text{max}}(X^T X) + \lambda}{\lambda_{\text{min}}(X^T X) + \lambda}.$$
Subset selection: Exhaustive Search

- Following Ockham’s razor, ideally we would like to know for any value $p$
  \[
  \min_{\alpha, \|\alpha\|_0=p} L(\alpha)
  \]

- That is, select the vector $\alpha$ which only considers $p$ variables which has the best fit.

- This is akin to doing selecting the **best** combination of variables.

### Practical Implementation

- For $p \leq n$, $\binom{n}{p}$ possible combinations of $p$ variables.

- Brute force approach: generate $\binom{n}{p}$ regression problems and select the one that achieves the best RSS.

**Impossible in practice** with moderately large $n$ and $p...\binom{30}{5} = 150,000$
Subset selection: Forward Search

- In Forward search:
  - define $I_1 = \{0\}$.
  - given a set $I_k$ of $k$ variables already selected in $0, \cdots, d$:
    - Compute for each variable $i$ in $0, \cdots, d \setminus I_k$
      $$t_i = \min_{(\alpha_k)_{k \in I_k}, \alpha} \left\| y_j - \left( \sum_{k \in I_k} \alpha_k x_{k,j} + \alpha x_{i,j} \right) \right\|^2$$
    - Set $I_{k+1} = I_k \cup \{i^*\}$ for any $i^*$ such that $i^* = \min t_i$.
    - $k = k + 1$ until desired number of variables.
In Backward search:

- define \( I_d = \{0, 1, \cdots, n\} \).
- given a set \( I_k \) of \( k \) variables already selected in \( 0, \cdots, d \):
  - Compute for each variable \( i \) in \( I_k \)
    \[
    t_i = \min_{(\alpha_k)_{k \in I_k, k \neq i}, \alpha} \sum_{j=1}^{N} \left\| y_j - \left( \sum_{k \in I_k} \alpha_k x_{k,j} + \alpha x_{i,j} \right) \right\|^2
    \]
  - Set \( I_{k-1} = I_k \setminus \{i^*\} \) for any \( i^* \) such that \( i^* = \max t_i \).
  - \( k = k - 1 \) until desired number of variables