CSI 436/536
Introduction to Machine Learning

Regression and LLSE

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York
Regression problem

• Use input to estimate a target variable that takes continuous values

• It is an example of **supervised machine learning** problem: in training, the target variables together with the inputs are given

• In testing, we only have input and need to estimate the target
Regression problem

- robotic control/automatic driving
  - input: internal parameters of robotic arm (force at angle)
  - output: end effector location
- treat input-output as going through a black box transform
- use training data to figure out best control function

\[ x \xrightarrow{f_\theta(x)} y' \]

\[ L \]

\[ y \]
Regression problem

- High-frequency stock trading (algorithmic trading)
  - input: historic stock prices & trading records
  - output: new trading action
  - treat input-output as going through a black box transform
  - use training data to figure out best control function

\[ x \xrightarrow{f_\theta(x)} y' \xrightarrow{L} y \]
Notations

- Data matrix can include processed data, i.e.,
  \[ g(x) \text{ is a function on raw } x \]

- Mean and centering

  - introduce N-dim all one vectors \( 1_N \), the (arithmetic) mean of data is computed as
  \[ m = \frac{1}{N} X 1_N \]

  - The (column) centering operation is expressed as
  \[ \tilde{X} = X - m 1_N^T = X - \frac{1}{N} X 1_N 1_N^T = X \left( I - \frac{1}{N} 1_N 1_N^T \right) \]
  the final matrix is the column centering operation

- **Correlation** and **covariance** matrices are defined as \( XX^T \) and \( \tilde{X} \tilde{X}^T \), respectively
Kernel matrix

- Definition: $G = X^T X \succeq 0$, $G_{ij} = x_i^T x_j$, element is the pairwise inner product of two points
- This matrix is known as the inner product matrix, the Gram matrix, or the *kernel* matrix
- It is in a sense the *dual* of the correlation matrix $XX^T$, when $X$ is full ranked, then at least one of them is invertible
- Kernel matrix plays a central role in the subsequent nonlinear extension of linear machine learning algorithms
General regression

- **Training**
  - Training data matrix
    - data points are column vectors
  - Training targets, assuming scalar
  - parametric function \( f_w(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R} \)
  - **loss function** \( L(y - f_w(x)) \geq 0 \)
  - Numerical procedure to find optimal \( w \) to minimize the learning objective \( \sum_{i=1}^{n} L(y_i - f_w(x_i)) \)

- In testing, for input \( x \) and generate prediction \( f_w(x) \)
  - **metric function** \( m(y - f_w(x)) \geq 0 \) on a validation dataset, may be different from the loss

\[
X = \begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \quad y = (y_1, y_2, \cdots, y_N)^T
\]
Linear least squares regression

- **Training**
  - Training data matrix
    - data points are column vectors
  - Training targets, assuming scalar
  - **Linear** function $f_w(x) = w^T \phi(x)$
  - **Least squares loss function**
    - $L(y, f_w(x)) = \|y - f_w(x)\|^2$
  - Optimal solution to the learning objective
    - $\sum_{i=1}^n L(y_i - f_w(x_i))$ satisfies the normal equation

- **Testing**
  - Metric function is also the least squares loss
LLSE: the Swiss army knife in ML

- Learning tasks
  - Supervised learning
    - Regression: basic LLSE and weighted LLSE
    - Classification: discriminative LLSE
  - Unsupervised learning
    - Clustering: multi-modal LLSE
    - Dimension reduction: total LLSE
- Learning paradigms
  - Batch learning: all other LLSE methods
  - Online learning: recursive LLSE
  - Dynamic programming: segmented LLSE
- Control of overfitting
  - Model selection: model selection LLSE
  - cross-validation: LOO LLSE
  - Regularization: ridge LLSE & LASSO
LLSE — linear function

- finding linear relation between input/output
  \[ f(x) = ax + b \]
- solving an optimization problem
  \[ \min_{w=(a,b)^T} \sum_{i=1}^{N} (y_i - ax_i - b)^2 \]
LLSE — quadratic function

• finding quadratic relation between input/output
  \[ f(x) = ax^2 + bx + c \]

• solving an optimization problem
  \[ \min_{w=(a,b,c)^T} \sum_{i=1}^{N} (y_i - ax_i^2 - bx_i^2 - c)^2 \]
LLSE — polynomial function

- find d-degree polynomial
  \[ f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_dx^d \]
  as
  \[ \min_{w=(a_0,\ldots,a_d)^T} \sum_{i=1}^{N} (y_i - f(x_i))^2 \]
LLSE — arbitrary basis functions

- find linear combinations of basis functions
  \[ f(x) = a_0 + a_1 g_1(x) + a_2 g_2(x) + \cdots + a_d g_d(x) \]
  to minimize
  \[ \min_{w=(a_0,\ldots,a_d)^T} \sum_{i=1}^{N} (y_i - f(x_i))^2 \]

- monomials: \( g_i(x) = x^i \) (polynomial fitting)

- Chebychev (orthogonal) polynomials

- Hermite polynomials: \( g_i(x) = e^{x^2} \frac{d^i e^{-x^2}}{dx^i} \)

- complex exponentials (Fourier transform): \( g_i(x) = e^{-i\alpha x} \)

- radial basis functions (RBFs): \( g_i(x) = e^{-a_i(x-b_i)^2} \)
LLSE — general case

- Define the general problem as fitting \( \sum_{i=1}^{m} a_i g_i(x_j) \) to target \( y \) by minimizing \( \sum_{j=1}^{n} (y_j - \sum_{i=1}^{m} a_i g_i(x_j))^2 \)

- Rewrite using linear algebra notations

\[
\begin{align*}
y &= \begin{pmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N
\end{pmatrix},
\quad w &= \begin{pmatrix}
a_1 \\
a_2 \\
\vdots \\
a_m
\end{pmatrix},
\quad \text{objective is } \min_w ||y - X^T w||^2
\end{align*}
\]

matrix \( X = \begin{pmatrix}
g_1(x_1) & g_1(x_2) & \cdots & g_1(x_N) \\
g_2(x_1) & g_2(x_2) & \cdots & g_2(x_N) \\
\vdots & \vdots & \ddots & \vdots \\
g_m(x_1) & g_m(x_2) & \cdots & g_m(x_N)
\end{pmatrix} \)
Solving LLSE

• Expand the terms
  \[ \|y - X^Tw\|^2 = y^Ty - 2y^TX^Tw + w^TXX^Tw \]

• Taking derivative on both sides w.r.t. \( w \)
  \[ \nabla_w \|y - X^Tw\|^2 = 2(XX^Tw - Xy) = 0 \]

• The solution is given by \( XX^Tw = Xy \), which is known as the normal equation

• Check Hessian matrix \( \nabla \nabla^T_w \|y - X^Tw\|^2 = 2XX^T \preceq 0 \)
  (why?)
  so the solution is a minimum

• We will assume the data matrix is full ranked (no linearly dependent rows or columns)
Weighted LLSE

• Introducing a weight matrix $W$, usually diagonal with $W_{ii} \geq 0$, and to solve
  \[ \min_w (y - X^T w)^T W (y - X^T w) \]

• This is known as weighted LLSE

• When $W = I$, WLLSE reduces to LLSE

\[
(y - X^T w)^T W (y - X^T w) = \sum_{i=1}^{n} W_{ii} \left( y_i - \sum_{j=1}^{m} a_j g_j(x_i) \right)^2
\]

• Solution
  \[
  \nabla_w (y - X^T w)^T W (y - X^T w) = 2 (XWX^T w - XW y) = 0
  \]
  so $XWX^T w = XW y \Rightarrow w = (XWX^T)^{-1} XW y$
Weighted LLSE

- How to determine the weight
  - Larger weight => error has to be small
  - Smaller weight => more relaxed error
- Relation with the variance of the error
  \[ W_{ii} = \frac{1}{\sigma_i^2} \], where \( \sigma_i^2 \) is the variance of the error in the corresponding component
  - Larger variance => less reliable estimation => smaller weight => more relaxed error
  - Smaller variance => more reliable estimation => larger weight => error has to be small
Solving normal equation

- case 1: complete problem
  \( N = m \), i.e., # of data = # of parameters
  \( \Rightarrow \) matrix \( X \) is square
  \( \Rightarrow \) correlation matrix \( XX^T \), \( X \) and \( X^T \) are all invertible

- case 2: over-complete problem
  \( N > m \), i.e., # of data > # of parameters
  \( \Rightarrow \) matrix \( X \) is short & fat
  \( \Rightarrow \) correlation matrix \( XX^T \) is \( N \times N \) and invertible

- case 3: under-complete problem
  \( N < m \), i.e., # of data < # of parameters
  \( \Rightarrow \) matrix \( X \) is tall & thin
  \( \Rightarrow \) correlation matrix \( XX^T \) is \( m \times m \) and not invertible, but the Gram matrix \( X^TX \) is invertible
Complete case

- We can solve directly by matrix inversion
  \[ XX^T w = Xy \Rightarrow X^T w = y \Rightarrow w = X^{-T}y \]

- Prediction error is zero: \[ y - X^T w = y - X^T X^{-T}y = 0 \]

- Direct matrix inversion is usually not a good option

- Solving \( Xp = y \) becomes \( LDUp = y \), then two steps \( Lx = y \) (forward elimination), \( DUp = x \) (backward elimination)

  - This is known as Gaussian elimination

- Solution time is \( O(n^2) \), and numerically it is very stable (caveat: if the pivots are chosen right)

- It is numerically stable (only divide by pivot)
over-complete problem

- Correlation matrix $XX^T$ is invertible and positive definite so LLSE objective function has unique global optimal solution, as $XX^Tw = Xy \Rightarrow w = (XX^T)^{-1}Xy$

- Interpretation: projection of $y$ in row space of $X$

- Prediction is $X^Tw = X^T(XX^T)^{-1}Xy$

- Prediction error is
  \[ y - X^Tw = y - X^T(XX^T)^{-1}Xy = (I_N - X^T(XX^T)^{-1}X)y \]

- $(XX^T)^{-1}X$ is known as the left Penrose-Moore pseudo inverse of general matrix $X^T$, as $(XX^T)^{-1}XX^T = I_N$
under-complete problem

- X is not invertible, $X^TX$ is invertible and p.d.
- Define the right Penrose-Moore pseudo inverse of general matrix $X$, $X^T(XX^T)^{-1}$, then $w = X^T(XX^T)^{-1}y$ is a solution to the normal equation
- solution is not unique
  - for any vector in the null space of $X$, $Xh = 0$, $p + h$ is also a solution
  - $p$ is a solution, we have $X(p + h) = Xp = y$
- there are infinite number of solutions that lead to zero least squares error (ill-posed problem)
under-complete problem

- Correlation matrix $XX^T$ is not invertible, but Gram matrix $X^TX$ is invertible and p.d.

- Define the right Penrose-Moore pseudo inverse of general matrix $X$, $X(X^TX)^{-1}$, then $w = X(X^TX)^{-1}y$ is a solution to the normal equation

- solution is not unique
  - for any vector in the row null space of $X$, $X^Th = 0$, $w+h$ is also a solution
  - $w$ is a solution, we have $X^T(w+h) = X^Tw = y$

- there are infinite number of solutions that lead to zero least squares error (ill-posed problem)
Solving normal equation

- **case 1:** complete problem
  \( N = m \), i.e., \# of data = \# of parameters
  \( \Rightarrow \) matrix \( X \) is square
  \( \Rightarrow \) **unique** solution with **zero prediction error**

- **case 2:** over-complete problem
  \( N > m \), i.e., \# of data > \# of parameters
  \( \Rightarrow \) matrix \( X \) is short & fat
  \( \Rightarrow \) **unique** solution with **non-zero prediction error**

- **case 3:** under-complete problem
  \( N < m \), i.e., \# of data < \# of parameters
  \( \Rightarrow \) matrix \( X \) is tall & thin
  \( \Rightarrow \) **non-unique** solution with **zero prediction error**
LLSE — general procedure

• Obtain training data $X$
• Decide number of base functions to use
• Choose a proper weight matrix $W$
• Form LSE objective function, and solve the normal equation for optimal solution
Issues

- Squared L2 loss is sensitive to outliers in training data
- Using L1 loss is more robust to outliers in training data
- Data points may not come at the same time, we need to handle the data in an online manner
- Using a high degree of polynomial may overfit the data, how do we control that from occurring
- The number of base functions (degree of polynomials) is a hyper-parameter, how do we select it