COMP5211: Machine Learning

Lecture 2

Minhao Cheng
Last lecture
Matrix calculus

- \( f = \|Xw - y\|^2 \), solve \( \frac{\partial f}{\partial w} \), where \( y \) is \( m \times 1 \) vector, \( X \) is \( m \times n \) matrix, \( w \) is \( n \times 1 \) vector

\[
df = d(\|Xw - y\|^2) = d((Xw - y)^T(Xw - y)) = d((Xw - y)^T)(Xw - y) + (Xw - y)^Td(Xw - y)
\]

\[
= (Xdw)^T(Xw - y) + (Xw - y)^T(Xdw) = 2(Xw - y)^TXdw
\]

- So \( \frac{\partial f}{\partial w} = 2X^T(Xw - y) \)
Regression

Example

• Classification:
  • Customer record $\rightarrow$ Yes/No

• Regression: predicting credit limit
  • Customer record $\rightarrow$ dollar amount
Regression

Linear regression

• Classification:
  • Customer record $\rightarrow$ Yes/No

• Regression: predicting credit limit
  • Customer record $\rightarrow$ dollar amount

• Linear Regression:

\[
  h(x) = \sum_{i=0}^{d} w_i x_i = w^T x
\]
Linear Regression
The data set

- Training data:
  - \((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\)
  - \(x_n \in \mathbb{R}^d\): feature vector for a sample
  - \(y_n \in \mathbb{R}\): observed output (real number)
Linear Regression

The data set

- Training data:
  - \((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\)
  - \(x_n \in \mathbb{R}^d\): feature vector for a sample
  - \(y_n \in \mathbb{R}\): observed output (real number)

- Linear regression: find a function \(h(x) = w^T x\) to approximate \(y\)
Linear Regression

The data set

- Training data:
  - \((x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)\)
  - \(x_n \in \mathbb{R}^d\): feature vector for a sample
  - \(y_n \in \mathbb{R}\): observed output (real number)
- Linear regression: find a function \(h(x) = w^T x\) to approximate \(y\)
- Measure the error by \((h(x) - y)^2\) (square error)

  - **Training error**: \(E_{\text{train}}(h) = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2\)
Linear Regression

Illustration

\[ \mathbf{x} = (x) \in \mathbb{R} \]

\[ \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2 \]
Linear Regression

Matrix form

\[ E_{\text{train}}(w) = \frac{1}{N} \sum_{n=1}^{N} (x_n^T w - y_n)^2 = \frac{1}{N} \| \begin{bmatrix} x_1^T w - y_1 \\ x_2^T w - y_2 \\ \vdots \\ x_N^T w - y_N \end{bmatrix} \|^2 \]

\[ = \frac{1}{N} \| \begin{bmatrix} -x_1^T \\ -x_2^T \\ \vdots \\ -x_N^T \end{bmatrix} w - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \| \]

\[ = \frac{1}{N} \| X w - \bar{y} \|^2 \]
Linear Regression

Minimize $E_{\text{train}}$

- $\min_w f(w) = \|Xw - y\|^2$

- $E_{\text{train}}$: continuous, differentiable, convex

- Necessary condition of optimal $w$:

$$\nabla f(w^*) = \begin{bmatrix} \frac{\partial f}{\partial w_0}(w^*) \\ \vdots \\ \frac{\partial f}{\partial w_d}(w^*) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$
Linear Regression

Minimizing $f$

\[
f(w) = \|Xw - y\|^2 = w^T X^T Xw - 2w^T X^T y + y^T y
\]

\[
\nabla f(w) = 2(X^T Xw - X^T y)
\]

\[
\nabla f(w^*) = 0 \Rightarrow X^T Xw^* = X^T y
\]

\[\text{normal equation}\]
Linear Regression
Minimizing \( f \)

\[
f(w) = \|Xw - y\|^2 = w^T X^T Xw - 2w^T X^T y + y^T y
\]

\[
\nabla f(w) = 2(X^T Xw - X^T y)
\]

\[
\nabla f(w^*) = 0 \Rightarrow X^T X w^* = X^T y
\]

\[
\text{normal equation}
\]

\[
\Rightarrow w^* = (X^T X)^{-1} X^T y \quad \text{How?}
\]
Linear Regression
Solutions

• Case I: $X^T X$ is invertible $\Rightarrow$ Unique solution
  • Often when $N > d$
  • Yes, $w^* = (X^T X)^{-1} X^T y$

• Case II: $X^T X$ is non-invertible $\Rightarrow$ Many solutions
  • Often when $d > N$
Linear Regression
Linear System Solver

• A “linear system”:
  • Find the minimum 2-norm solution of $\min_w \|Xw - y\|$
Linear Regression

Linear System Solver

• A “linear system”:
  - Find the **minimum 2-norm solution** of \( \min_w \|Xw - y\| \)

• Let \( X = U \Sigma V^T \) be the SVD of \( X \):
  - \( U^T U = I \)
  - \( V^T V = I \)
  - \( \Sigma = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0] \), \( (\sigma_1, \ldots, \sigma_r > 0) \)

• Solution: \( w^+ = X^+ y \)
  - where \( X^+ = V \Sigma^+ U^T \), \( \Sigma^+ = \text{diag}[1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_r, 0, \ldots, 0] \)
Linear Regression
Linear System Solver

- A “linear system”:
  - Find the minimum 2-norm solution of \( \min_w \|Xw - y\| \)

- Let \( X = U\Sigma V^T \) be the SVD of \( X \):
  - \( U^TU = I \)
  - \( V^TV = I \)
  - \( \Sigma = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0] \), \( (\sigma_1, \ldots, \sigma_r > 0) \)

- Solution: \( w^+ = X^+ y \)
  - where \( X^+ = V\Sigma^+ U^T \), \( \Sigma^+ = \text{diag}[1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_r, 0, \ldots, 0] \)
  - \( X^+ \): pseudo-inverse of \( X \)
  - Why?
Linear Regression
Linear System Solver (Cont*)

- \( \|Xw - y\|^2 = \|U\Sigma V^Tw - y\|^2 = \|\Sigma V^Tw - U^Ty\|^2 \) since?

- Let \( z = V^Tw, \)
  - The solution of \( \min \|\Sigma V^Tw - U^Ty\|^2 \) is equivalent to find the solution of \( \min \|\Sigma z - U^Ty\|^2 \)
    - Since \( \Sigma \) is diagonal, so the solution is \( z^+ = \Sigma^+ U^Ty \)
  - So \( w^+ = Vz^+ = V\Sigma^+ U^Ty \)
Computation complexity
Dense vector and sparse vector

- If $x, y \in \mathbb{R}^m$ are dense:
  - $x + y, x - y, x^T y : O(m)$ operations

- If $x, y \in \mathbb{R}^m$, $x$ is dense and $y$ is sparse:
  - $x + y, x - y, x^T y : O(\text{nnz}(y))$ operations

- If $x, y \in \mathbb{R}^m$ and both of them are sparse:
  - $x + y, x - y, x^T y : O(\text{nnz}(y) + \text{nnz}(x))$ operations
Computation complexity
Dense matrix

- Let $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{m \times n}, s \in \mathbb{R}$:
  - $A + B, sA, A^T : O(mn)$ operations
- If $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{n \times 1}$
  - $Ab : O(mn)$ operations
- If $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times n}$,
  - $AB : O(n^3)$ operations; theoretical best: $O(n^{2.xxx})$
- $A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{k \times n}$,
  - $AB : O(mnk)$ operations;
Computation complexity
Sparse matrix format

• Widely-used format: Compressed Sparse Column (CSC), Compressed Sparse Row (CSR), …

• CSR: three arrays for storing an $m \times n$ matrix with $nnz$ nonzero
  
  • $val$ ($nnz$ real numbers): the values of each nonzero elements
  
  • $row\_ind$ ($nnz$ integers): the column indices corresponding to the values
  
  • $col\_ptr$ ($m + 1$ integers): the list of value indexes where each column starts

\[
\begin{array}{ccc}
10 & 0 & 0 \\
0 & 30 & 0 \\
0 & 40 & 0 \\
20 & 50 & 0 \\
0 & 0 & 60 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c}
\hline
Val: & 10 & 30 & 40 & 20 & 50 & 60 \\
\hline
col\_idx: & 0 & 1 & 1 & 0 & 1 & 2 \\
\hline
row\_ptr: & 0 & 1 & 2 & 3 & 5 & 6 \\
\hline
\end{array}
\]
Computation complexity
Sparse matrix operations

- Let $A \in \mathbb{R}^{m \times n}$ (sparse), $B \in \mathbb{R}^{m \times n}$ (sparse or dense), $s \in \mathbb{R}$:
  - $A + B, sA, A^T : O(\text{nnz})$ operations
- If $A \in \mathbb{R}^{m \times n}$ (sparse), $b \in \mathbb{R}^{n \times 1}$
  - $Ab : O(\text{nnz})$ operations
- If $A \in \mathbb{R}^{n \times k}$ (sparse), $B \in \mathbb{R}^{k \times n}$ (dense),
  - $AB : O(\text{nnz}(A)n)$ operations (use sparse BLAS)
- $A \in \mathbb{R}^{n \times k}$ (sparse), $B \in \mathbb{R}^{k \times n}$ (sparse),
  - $AB : O(\text{nnz}(A)\text{nnz}(B)/k)$ in average
  - $AB : O(\text{nnz}(A)n)$ worst case
  - The resulting matrix will be much denser
Linear Regression
Computational complexity

- Computational cost for computing \((X^TX)^{-1}X^Ty\):
  - Computing \(X^TX\): \(O(d^2N)\) time
  - Computing matrix inversion: \(O(d^3)\) time
  - Overall complexity: \(O(d^2N + d^3)\)
- What if \(d, N \approx \) millions?
  - (Use iterative algorithms, next class)
Logistic Regression
Binary Classification

- Input: training data $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ and corresponding outputs $y_1, y_2, \ldots, y_n \in \{+1, -1\}$

- Training: compute a function $f$ such that $\text{sign}(f(x_i)) \approx y_i$ for all $i$

- Prediction: given a testing sample $\tilde{x}$, predict the output as $\text{sign}(f(x_i))$
Logistic Regression
Binary Classification

• Assume linear scoring function: \( s = f(x) = w^T x \)

• Logistic hypothesis:
  
  \[
  P(y = 1 \mid x) = \theta(w^T x),
  \]

  Where \( \theta(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}} \)

  • It is called sigmoid function
Logistic Regression
Binary Classification

• Assume linear scoring function: \( s = f(x) = w^T x \)

• Logistic hypothesis:
  - \( P(y = 1 \mid x) = \theta(w^T x) \),
  - Where \( \theta(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}} \)

• How about \( P(y = -1 \mid x) \)?
Logistic Regression
Binary Classification

• Assume linear scoring function: \( s = f(x) = w^T x \)

• Logistic hypothesis:
  
  \[
  P(y = 1 \mid x) = \theta(w^T x),
  \]
  
  Where \( \theta(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}} \)

• How about \( P(y = -1 \mid x) \)?

  \[
  P(y = -1 \mid x) = 1 - \frac{1}{1 + e^{-w^T x}} = \frac{1}{1 + e^{w^T x}} = \theta(-w^T x)
  \]
Logistic Regression
Binary Classification

• Assume linear scoring function: \( s = f(x) = w^T x \)

• Logistic hypothesis:
  
  \[ P(y = 1 \mid x) = \theta(w^T x), \]

  • Where \( \theta(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}} \)

• How about \( P(y = -1 \mid x) \)?
  
  \[ P(y = -1 \mid x) = 1 - \frac{1}{1 + e^{-w^T x}} = \frac{1}{1 + e^{w^T x}} = \theta(-w^T x) \]

• Therefore, \( P(y \mid x) = \theta(yw^T x) \)
Logistic Regression
Maximizing the likelihood

- Likelihood of $\mathcal{D} = (x_1, y_1), \ldots, (x_N, y_N)$:

$$\prod_{n=1}^{N} P(y_n | x_n) = \prod_{n=1}^{N} \theta(y_n w^T x_n)$$
Logistic Regression
Maximizing the likelihood

- Likelihood of \( \mathcal{D} = (x_1, y_1), \ldots, (x_N, y_N) \):
  \[
  \prod_{n=1}^{N} P(y_n | x_n) = \prod_{n=1}^{N} \theta(y_n w^T x_n)
  \]

- Find \( w \) to maximize the likelihood!
  \[
  \max_w \prod_{n=1}^{N} \theta(y_n w^T x_n) \\
  \equiv \max_w \log(\prod_{n=1}^{N} \theta(y_n w^T x_n)) \\
  \equiv \min_w - \sum_{n=1}^{N} \log(\theta(y_n w^T x_n)) \\
  \equiv \min_w \sum_{n=1}^{N} \log(1 + e^{-y_n w^T x_n})
  \]
Logistic Regression
Empirical Risk Minimization (linear)

• Linear classification/regression:

\[
\min_w \frac{1}{N} \sum_{n=1}^{N} \text{loss}(\hat{\gamma}_n, y_n)
\]

\[
\hat{\gamma}_n: \text{the predicted score}
\]

• Linear regression:

\[
\text{loss}(h(x_n), y_n) = (w^T x_n - y_n)^2
\]

• Logistic regression:

\[
\text{loss}(h(x_n), y_n) = \log(1 + e^{-y_n w^T x_n})
\]
Support Vector Machines

Hinge loss

- Replace the logistic loss by hinge loss:

\[
\min_w \frac{1}{N} \sum_{n=1}^{N} \max(0, 1 - y_n w^T x_n)
\]
Logistic Regression

Empirical Risk Minimization (general)

- Assume $f_W(x)$ is the decision function to be learned
  - $(W$ is the parameters of the function$)$
- General empirical risk minimization

$$\min_W \frac{1}{N} \sum_{n=1}^{N} \text{loss}(f_W(x_n), y_n)$$

- Example: Neural network ($f_W(\cdot)$ is the network)