intro to machine learning

zhangyifeng

October 21, 2018

Contents

1 Intro

1.1 Course Requirements and Grading

Lab(30%)

- Python
- Synthetic data
- 2 deliverables, distributed over moodle

Theory exercises(0/20)

close to the end(early December)

Final exam(70%)

- Theory questions(judgement-oriented)
- Simulate running algorithms by hand

Meeting hours

- Office: 104B, 68-72 Gower street
- Meeting hours: Tuesday, 14:00-15:00

Prerequisites

Linear Algebra; Calculus; Probability; Programming

1.2 Course

Machine Learning

data -> maodel ->prediction

Least squares model

- least squares solution for linear regression
- Least squares solution for generalized linear regression
- Least squares solution for ridge regression

notation

$\boldsymbol{D}$: probleim dimension, e.g. 1D, 2D(can visualize)
$\boldsymbol{N}$: training set size
Training set: input-output pairs $S = \{\boldsymbol{x}_i, y_i\}, i = 1, \ldots, N$ where,$\boldsymbol{x}_i = \{x_{i1}, \ldots, x_{iD}\}^T \in \mathbb{R}^D, y_i \in \mathbb{R}$, generally can be $\boldsymbol{x}$
$\boldsymbol{w}$: weight, $\boldsymbol{w} = \{w_1, \ldots, w_D\}^T \in \mathbb{R}^D$
ϵ_i: noise
$X = \{x_1, x_2, \ldots, x_N\}^T = \{x_1^T; x_2^T; \ldots; x_N^T\}$
Remark: ";" represent column vector
$\boldsymbol{y} = \{y_1, \ldots, y_N\}^T$
$\boldsymbol{\epsilon} = \{\epsilon_1, \ldots, \epsilon_N\}^T$: residuals vectors

linear regression model

$$\boldsymbol{y} = X\boldsymbol{w} + \epsilon \quad or \quad \boldsymbol{y}^T = \boldsymbol{w}^T X^T + \boldsymbol{\epsilon}^T$$

that is $y_i = \boldsymbol{x}_i^T \boldsymbol{w} + \epsilon_i, or\ y_i = \boldsymbol{w}^T \boldsymbol{x}_i + \epsilon_i, i = 1, \ldots, N$
Loss function: $L(w) = \sum_{i=1}^{N} (y_i - \boldsymbol{w}^T x_i)^2$
goal: $\min\limits_{w} L(w)$ Least squares solution for linear regression: $w^* = (X^T X)^{-1} X^T \boldsymbol{y}$

Generalized linear regression model

$\boldsymbol{x} \to [\boldsymbol{\phi}(\boldsymbol{x})] = [\phi_1(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x})]^T$, where $\phi_i(\boldsymbol{x}), i = 1, \ldots, M$, $\phi_i(\boldsymbol{x})$ can be other form besides x_i (if x_i, and $M = D$, it is just the linear regression model)
If $D = 1, and\ \phi_i(x) = x^{i-1}$, then it is k-th degree ploynomial fitting
If the highest order of $\phi_i(\boldsymbol{x})$ is 2, then it is second-order polynomials fitting
set $\Phi = [\boldsymbol{\phi}(\boldsymbol{x_1})^T; \boldsymbol{\phi}(\boldsymbol{x_2})^T; \ldots; \boldsymbol{\phi}(\boldsymbol{x_N})^T]$
then the model is:

$$\boldsymbol{y} = \Phi \boldsymbol{w} + \boldsymbol{\epsilon}$$

Least squares solution for generalized linear regression: $w^* = (\Phi^T \Phi)^{-1} \Phi^T \boldsymbol{y}$

approximations

If $N > D$ (e.g. 30 points, 2 dimensions): overdetermined system
If $N < D$ (e.g. 30 points, 3000 dimensions): underdetermined system (overfitting)

How to control complexity (Regularized linear regression)

1.use vector norm (L2, L1, Lp norm) to measure residual vector
Remark: different norm represent different regularized linear regression, here we use L2 norm

2.rewrite loss function: $L(\boldsymbol{w}) = ||\boldsymbol{y} - X\boldsymbol{w}||^2 + \lambda||\boldsymbol{w}||^2$
this is Ridge regression, a.k.a, L2-regularized linear regression
Remark: λ is "hyperparameter", select λ with cross-validation(use cross-validation for diff values of λ – pick value minimizes cross-validation error)

Cross-validation: least glorious, most effective of all methods (teacher said)

3.Least squares solution for ridge regression: $w^* = (X^T X + \lambda \boldsymbol{I})^{-1} X^T \boldsymbol{y}$

2 Logistic Regression

Machine Learning variants–Supervised learning

- Classification
- Regression

Gaussian (or Normal) distribution

one-dimensional case
- Mean μ
- Variance σ^2
- Pdf: $\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp\{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T}{2\sigma^2}\}$

Multi-dimensional case
- Mean $\boldsymbol{\mu}$
- Covariance $\boldsymbol{\Sigma}$
- Pdf: $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp(\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\})$

Remark: $|\Sigma|$ represent matrix norm, e.g. Frobenius norm of Σ

Parameter estimation

Given: $X = \{x_1, x_2, \ldots, x_N\}$, parametric form of distribution, parameters θ
Learning goal: estimate θ
Likelihood of θ: $L(\theta) = p(X; \theta) = \prod_{n=1}^{N} p(x_n; \theta)$

Log-likelihood: $\ln L(\theta) = \sum_{n=1}^{N} \ln p(x_n; \theta)$
$\max \ln L(\theta) \Rightarrow \frac{\partial \ln L(\theta)}{\partial \theta} = 0 \Rightarrow \hat{\theta}$