Machine Learning Review

1 Linear Regression

Assume a set of traning data is denoted by $\{x^{(i)}, y^{(i)}\}_{i=1,\cdots,m}$ where $x^{(i)} \in \mathbb{R}^n$ and $y^{(i)} \in \mathbb{R}$. Our aim is to find an optimal θ such that the hypothesis function $h_\theta : \mathbb{R}^n \to \mathbb{R}$ fits the training data best. In particular, our problem can be formulated under the Euclidean norm as follows

$$\min_\theta \quad J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

where $J(\theta)$ is so-called a cost function One choice to solve the above problem is the *Gradient Descent* (GD) algorithm (see pp. 7 in our slides).

Another soution to the lienar regression problem is to directly compute the close form solution (see pp. 13-17 in our slides). Assuming

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} (y^{(1)}) \\ \vdots \\ (y^{(m)}) \end{bmatrix}$$

$J(\theta)$ can be rewritten as

$$J(\theta) = \frac{1}{2}(\vec{y} - X\theta)^T(\vec{y} - X\theta)$$

It can be minimized by letting its derivative w.r.t. θ equal zero, i.e.,

$$\bigtriangledown_\theta J(\theta) = X^T X\theta - X^T\vec{y}$$

Therefore, we have

$$\theta = (X^T X)^{-1}X^T Y$$

2 Logistic Regression

In this section, we look at classificatin problem where $y^{(i)} \in \{0, 1\}$ (so-called the *label* for the training example). Basically, we use a logistic function (or sigmoid function) $g(z) = 1/(1 + e^{-z})$ to continuously approximate the discrete classification. In particular, we look at the probability that a given data x belongs to a category $y = 0$ (or $y = 1$). The hypothesis function is defined as

$$h_\theta(x) = g(\theta^T x) = 1/(1 + e^{-\theta^T x})$$

Assuming

$$p(y = 1 \mid x; \theta) = h_\theta(x) = 1/(1 + \exp(-\theta^T x)$$
$$p(y = 0 \mid x; \theta) = 1 - h_\theta(x) = 1/(1 + \exp(\theta^T x)$$

we have that

$$p(y \mid x; \theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

Given a set of training data $\{x^{(i)}, y^{(i)}\}_{i=1,\cdots,m}$, we define the likelihood function as

$$\begin{aligned} L(\theta) &= \prod_i p(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_i (h_\theta(x^{(i)}))^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}} \end{aligned}$$

To simply the computations, we take the logarithm of the likelihood function (i.e. so-called *log-likelihood*)

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \Big(y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})\Big)$$

We can use *gradient ascent* algorithm to maximize the above objective function (see pp. 8 in our slides). Also, we can take another choice, i.e., *Newton's method* (see. pp. 9-11 in our slides)

3 Regularization and Bayesian Statistics

We first take the linear regression problem for an example. To sovle the overfitting problem, we introduce a regularization item (or penalty) in the objective function, i.e.,

$$\min_\theta \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

A gradient descent based solution is given on pp. 7 in our slides. Also, the closed form solution can be found on pp. 9. As shown on pp. 10, similar method can be applied to logistic regression problem.

We then look at *Maximum Likelihood Estimation* (MLE). Assume data are generated via probability distribution

$$d \sim p(d \mid \theta)$$

Given a set of independent and identically distributed (i.i.d.) data samples $D = \{d_i\}_{i=1,\cdots,m}$, our goal is to estimate θ that best models the data. The log-likelihood function is defined by

$$\ell(\theta) = \log \prod_{i=1}^{m} p(d_i \mid \theta) = \sum_{i=1}^{m} p(d_i \mid \theta)$$

Our problem becomes

$$\theta_{MLE} = \arg\max_\theta \ell(\theta) = \arg\max_\theta \sum_{i=1}^{m} p(d_i \mid \theta)$$

In *Maximum-a-Posteriori Estimation* (MAP), we treat θ as a random variable. Our goal is to choose that maximizes the pos- terior probability of (i.e., probability in the light of the observed data). According to bayes rule

$$p(\theta \mid D) = \frac{p(\theta)p(D \mid \theta)}{p(D)}$$

Note that, $p(D)$ is **independent** of θ. In MAP, our problem is formulated by

$$\begin{aligned}
\theta_{MAP} &= \arg\max_{\theta} p(\theta \mid D) \\
&= \arg\max_{\theta} \frac{p(\theta)p(D \mid \theta)}{p(D)} \\
&= \arg\max_{\theta} p(\theta)p(D \mid \theta) \\
&= \arg\max_{\theta} \log p(\theta)p(D \mid \theta) \\
&= \arg\max_{\theta} (\log p(\theta) + \log p(D \mid \theta)) \\
&= \arg\max_{\theta} \left(\log p(\theta) + \sum_{i=1}^{m} \log p(d_i \mid \theta) \right)
\end{aligned}$$

The detailed application of MLE and MAP to linear regression and logistic regression can be found in pp. 15-22.

4 Naive Bayes and EM Algorithm

Given a set of training data, the features and labels are both represented by random variables $\{X_j\}_{j=1,\cdots,n}$ and Y, respectively. The key of Naive Bayes is to assume that X_j and $X_{j'}$ are conditionally independent given Y, for $\forall j \neq j'$. Hence, given a data sample $x = [x_1, x_2, \cdots, x_n]$ and its label y, we have that

$$\begin{aligned}
& p(X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n \mid Y = y) \\
= & \prod_{j=1}^{n} p(X_j = x_j \mid X_1 = x_1, X_2 = x_2, \cdots, X_{j-1} = x_{j-1}, Y = y) \\
= & \prod_{j=1}^{n} p(X_j = x_j \mid Y = y)
\end{aligned}$$

Then the Naive Bayes model can be formulated as

$$\begin{aligned}
& p(Y = y \mid X_1 = x_1, \cdots, X_n = x_n) \\
= & \frac{p(X_1 = x_1, \cdots, X_n = x_n \mid Y = y)p(Y = y)}{p(X_1 = x_1, \cdots, X_n = x_n)} \\
= & \frac{p(Y = y) \prod_{j=1}^{n} p(X_j = x_j \mid Y = y)}{p(X_1 = x_1, \cdots, X_n = x_n)}
\end{aligned}$$

where there are two sets of parameters (denoted by Ω): $p(Y = y)$ (or $p(y)$) and $p(X_j = x \mid Y = y)$ (or $p_j(x \mid y)$). Since $p(X_1 = x_1, \cdots, X_n = x_n)$ follows a predefined distribution and is independent of these parameters, we have that

$$p(y \mid x_1, \cdots, x_n) \propto p(y) \prod_{j=1}^{n} p_j(x_j \mid y)$$

Given a set of training data $\{x^{(i)}, y^{(i)}\}$, the log-likelihood function is

$$\begin{aligned}\ell(\Omega) &= \sum_{i=1}^{m} \log p(x^{(i)}, y^{(i)}) \\ &= \sum_{i=1}^{m} \log \left(p(y^{(i)}) \prod_{j=1}^{n} p_j(x_j^{(i)} \mid y^{(i)}) \right) \\ &= \sum_{i=1}^{m} \log p(y^{(i)}) + \sum_{i=1}^{m}\sum_{j=1}^{n} \log p_j(x_j^{(i)} \mid y^{(i)})\end{aligned}$$

The maximum-likelihood estimates are then the parameter values $p(y)$ and $p_j(x_j|y)$ that maximize

$$\ell(\Omega) = \sum_{i=1}^{m} \log p(y^{(i)}) + \sum_{i=1}^{m}\sum_{j=1}^{n} \log p_j(x_j^{(i)} \mid y^{(i)})$$

subject to the following constraints:

- $p(y) \geq 0$ for $\forall y \in \{1, 2, \cdots, k\}$
- $\sum_{y=1}^{k} p(y) = 1$
- For $\forall y \in \{1, \cdots, k\}, j \in \{1, \cdots, n\}, x \in \{0, 1\}$, $p_j(x \mid y) \geq 0$
- For $\forall y \in \{1, \cdots, k\}, j \in \{1, \cdots, n\}$, $\sum_{x \in \{0,1\}} p_j(x \mid y) = 1$

Solutions can be found on pp. 21 in our slides.

In the *Expectation-Maximization* (EM) algorithm, we assume there is no label for any training data. By introducing a latent variable z, the log-likelihood function can be defined as

$$\ell(\theta) = \sum_{i=1}^{m} \log p(x^{(i)}; \theta) = \sum_{i=1}^{m} \log \sum_{z} p(x^{(i)}, z; \theta)$$

The basic idea of EM algorithm (to maximizing $\ell(\theta)$) is to repeatedly construct a lower-bound on ℓ (E-step), and then optimize that lower-bound (M-step)

For each $i \in \{1, 2, \cdots, m\}$, let Q_i be some distribution over the z's

$$\sum_{z} Q_i(z) = 1, \quad Q_i(z) \geq 0$$

We have

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^{m} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta) \\ &= \sum_{i=1}^{m} \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \\ &\geq \sum_{i=1}^{m} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}\end{aligned}$$

since

$$\log\left(E_{z^{(i)}\sim Q_i}\left[\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}\right]\right) \geq E_{z^{(i)}\sim Q_i}\left[\log\left(\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}\right)\right]$$

according to the Jensen's inequality (see pp. 25 for the details of Jensen's inequality). Therefore, for any set of distributions Q_i, $\ell(\theta)$ has a lower bound

$$\sum_{i=1}^{m}\sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}$$

In order to tighten the lower bound (i.e., to let the equality hold in the Jensen's inequality), $p(x^{(i)},z^{(i)};\theta)/Q_i(z^{(i)})$ should be a constant (such that $\mathrm{E}[p(x^{(i)},z^{(i)};\theta)/Q_i(z^{(i)})] = p(x^{(i)},z^{(i)};\theta)/Q_i(z^{(i)})$). Therefore,

$$Q_i(z^{(i)}) \propto p(x,z^{(i)};\theta)$$

Since $\sum_z Q_i(z) = 1$, one natural choice is

$$\begin{aligned} Q_i(z^{(i)}) &= \frac{p(x^{(i)},z^{(i)};\theta)}{\sum_z p(x^{(i)},z^{(i)};\theta)} \\ &= \frac{p(x^{(i)},z^{(i)};\theta)}{p(x^{(i)};\theta)} \\ &= p(z^{(i)} \mid x^{(i)};\theta) \end{aligned}$$

In EM algorithm, we repeat the following step until convergence

- (E-step) For each i, set

$$Q_i(z^{(i)}) := p(z^{(i)} \mid x^{(i)};\theta)$$

- (M-step) set

$$\theta := \arg\max_{\theta} \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}$$

The convergence of the EM algorithm can be found on pp. 30 in our slides. An example of applying the EM algorithm to the Mixtures of Gaussians is given on pp. 31-37.

5 SVM

In SVM, we use a hyperplane ($w^T x + b = 0$) to separate the given training data. We first assume the training data is linearly separable. Give a training data $x^{(i)}$, the margin $\gamma^{(i)}$ is the distance between $x^{(i)}$ and the hyperplane

$$w^T\left(x^{(i)} - \gamma^{(i)}\frac{w}{\|w\|}\right) + b = 0 \Rightarrow \gamma^{(i)} = \left(\frac{w}{\|w\|}\right)^T x^{(i)} + \frac{b}{\|w\|}$$

If $y^{(i)} = 1$, $\gamma^{(i)} \geq 0$; otherwise (when $y^{(i)} = -1$), $\gamma^{(i)} < 0$. In SVM, we actually look at the geometric margin by removing the signs. Then $\gamma^{(i)}$ is redefined as

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{w}{\|w\|} \right)^T x^{(i)} + \frac{b}{\|w\|} \right) = \frac{y^{(i)}(w^T x^{(i)} + b)}{\|w\|}$$

Note that, scaling (w, b) does not change $\gamma^{(i)}$ (and thus γ).

For the whole training data, the (geometric) margin is written as

$$\gamma = \min_i \gamma^{(i)} = \frac{\min_i \{y^{(i)}(w^T x^{(i)} + b)\}}{\|w\|}$$

Then, SVM can be formulated as

$$\begin{array}{cl} \max_{\gamma,w,b} & \gamma \\ \text{s.t.} & y^{(i)}(w^T x^{(i)} + b) \geq \gamma\|w\|, \quad \forall i \end{array}$$

We scale (w, b) such that $\min_i\{y^{(i)}(w^T x^{(i)} + b)\} = 1$; therefore, $\gamma = 1/\|w\|$. Then, the above formulation becomes

$$\begin{array}{cl} \max_{w,b} & 1/\|w\| \\ \text{s.t.} & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad \forall i \end{array}$$

Maximizing $1/\|w\|$ is equivalent to minimizing $\|w\|^2 = w^T w$

$$\begin{array}{cl} \min_{w,b} & w^T w \\ \text{s.t.} & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad \forall i \end{array}$$

The above QP problem can be resolved by the state-of-art QP solvers. Unfortunately, existing generic QP solvers is of low efficiency, especially in face of a large training set. We hereby apply the Lagrange duality theory to the SVM problem. Details can be found on pp. 8-18.

When the training data is linearly non-separable, we can use kernel method (pp. 19-30) or the regularized version of SVM (pp. 31-40).

6 K-Means

In each iteration, we first (re)-assign each example $x^{(i)}$ to its closest cluster center (based on the smallest Euclidean distance)

$$\mathcal{C}_{k^*} = \{x^{(i)} : k^* = \arg\min_k \|x^{(i)} - \mu_k\|^2\}$$

i.e., $\mathcal{C}_k$ is the set of examples assigned to cluster k with center μ_k. We then update the cluster means

$$\mu_k = \text{mean}(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{x \in \mathcal{C}_k} x$$

The iterations are stopped when cluster means does not change by much.

7 Principle Component Analysis (PCA)

The PCA algorithm includes the following steps:

1. Center the data (subtract the mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$ from each data point)

$$x^{(i)} \leftarrow x^{(i)} - \mu$$

2. Compute the covariance matrix

$$S = \frac{1}{N}\sum_{i=1}^{N} x^{(i)} x^{(i)^T} = \frac{1}{N} X X^T$$

3. Do an eigendecomposition of the covariance matrix S
4. Take first K leading eigenvectors $\{u_l\}_{l=1,\cdots,K}$ with eigenvalues $\{\lambda_l\}_{l=1,\cdots,K}$
5. The final K dim. projection of data is given by

$$Z = U^T X$$

where U is $D \times K$ and Z is $K \times N$