Machine Learning

Sina Aleyaasin

October 8, 2017

1 Supervised Learning

Consider the task of modelling the relationship between some dependent variable (*target*) on some explanatory independent variable (*feature*) given a data set.

Definition 1.1. A **training example** is an ordered *feature-target* pair

$$(x^{(i)}, y^{(i)}) \in \mathbb{X} \times \mathbb{Y} \tag{1}$$

Definition 1.2. A **training set** of size N is the set of training examples

$$\{(x^{(i)}, y^{(i)}) | i = 1, 2, \dots N\} \tag{2}$$

The objective of supervised learning is to produce some *hypothesis*

$$h : \mathbb{X} \to \mathbb{Y} \tag{3}$$

to model the relationship by means of the corresponding training set.

Definition 1.3. **Regression** is supervised learning for a continuous, real valued $\mathbb{Y} \equiv \mathbb{R}^{n+1}$.

Definition 1.4. **Classification** is supervised learning for a finite, discrete $\mathbb{Y}$. The hypothesis of a classification problem is also known as a **classifier**.

1.1 Linear Regression

Linear regression assumes a linear dependence of the target on the features.

$$h(x; \theta) \equiv h_\theta(x) = \theta^T x \tag{4}$$

where $\theta, x \in \mathbb{R}^{n+1}$.

Remark. Motivated by aesthetics, this notation adheres to the convention that every feature vector x has a constant first element $x_0 = 1$ to account for the intercept term θ_0.

The parameter θ is determined by minimizing some *loss function* that aims to quantify the “error” of the classifier. We consider here the loss function giving corresponding to the **ordinary least squares** regression model

$$J(\theta) = \sum_{i=1}^{N} J^{(i)}(\theta) = \sum_{i=1}^{N} \frac{1}{2} \Big(y^{(i)} - h_\theta(x^{(i)}) \Big)^2 \tag{5}$$

1.1.1 Minimizing loss by normal equations

In some instances, as with the least mean squares regression model, the loss function may be minimized analytically to yield a closed formed solution for θ

theorem and proof of linear algebra + calculate theta

1.1.2 Minimizing loss by gradient descent

For instances where no closed form solutions exist, one may perform a **gradient descent** until a desired threshold of convergence is reached.

$$\theta_{j+1} = \theta_j - \alpha \nabla_\theta J(\theta) \tag{6}$$

The parameter α is known as the **learning rate**. This difference equation describes **batch gradient decent**. When the training set is large, one common modification can be made

$$\theta_{j+1} = \theta_j - \alpha \nabla_\theta J^{(i++)}(\theta_j) \tag{7}$$

where the ++ operator indicates iteration through the training set with each global iteration. This is referred to as a **stochastic gradient descent**.

1.2 Probabilistic interpretation

In order to appreciate the choice of loss function in the ordinary-least-squares model, consider the "error" produced by a linear model in a training example

$$\begin{aligned} \epsilon^{(i)} &= y^{(i)} - h(x^i) \\ &= y^{(i)} - \theta^T x^{(i)} \end{aligned} \tag{8}$$

Let us assume that the distribution of $\epsilon^{(i)}$ in some training set is independently and identically distributed (IID) according to a Gaussian distribution

$$p(e^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e^{(i)2}}{2\sigma^2}\right) \tag{9}$$

which by (8) implies

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \tag{10}$$

This assumption may alternatively be expressed

$$e^{(i)} \sim \mathcal{N}(0, \sigma^2) \tag{11}$$

Equation (10) states the conditional probability of the random variable $x^{(i)}$ given the random variable $y^{(i)}$, parameterised by θ, takes the form of the given Gaussian distribution. Minimizing the error amounts to maximizing this probability. In order to prescribe this probability to a training set, we use matrix notation.

Definition 1.5. The **design matrix** of a training set size N is the matrix

$$X = \begin{bmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(N)T} \end{bmatrix} \tag{12}$$

The **likelihood** function of a training set can then be expressed

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X;\theta) \tag{13}$$

where $\vec{y} = \left[y^{(1)}, y^{(2)}, \dots, y^{(N)}\right]^T$ Using the independence assumption

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{N} p(y^{(i)}|x^{(i)};\theta) \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \end{aligned} \tag{14}$$

We node that maximizing the likelihood function is equivalent to maximizing any monotonically increasing function of the likelihood. We choose the logarithm function[1] to yield

$$\begin{aligned} \ell(\theta) &\equiv \log L(\theta) \\ &= \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\ &\sim \sum_{i=1}^{N} \frac{1}{2}\left(y^{(i)} - \theta^T x^{(i)}\right)^2 \end{aligned} \tag{15}$$

Not incidentally, we see from (4) that maximizing the likelihood (15) is equivalent to minimizing our loss function defined in (5).

1.2.1 Weighted linear regression

One common modification to the liner regression model is to account for *weights* in the loss function

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N} w^{(i)}\left(y^{(i)} - h_\theta(x^{(i)})\right)^2 \tag{16}$$

A standard choice for $w^{(i)}$ is

$$w^{(i)} = \exp\left(-\frac{\left(x^{(i)} - x\right)^T \left(x^{(i)} - x\right)}{2\tau^2}\right) \tag{17}$$

where τ is the **bandwidth parameter** and

elaborate on x

[1] In computer arithmetic, addition is less expensive that multiplication. By using the logarithm function, products become sums and computation efficiency improves.

1.3 Logistic regression

Logistic regression is an approach to *binary classification* $\mathbb{Y} = \{0, 1\}$ using a classifier naturally based on the **logistic function**.

$$h(x; \theta) = \frac{1}{1 + e^{-\theta^T x}} \tag{18}$$

Probabilistically, we interpret h such that

$$h_\theta(x) = P(y = 1|x; \theta) \tag{19}$$

and hence the conditional probability of y given x takes the form

$$P(y|x; \theta) = h_\theta(x)^y + (1 - h_\theta(x))^{1-y} \tag{20}$$

Extending (20) for a training set, using the assumption that the target errors are independently and identically distributed

$$\begin{aligned} L(\vec{y}) &= p(X; \theta) \\ &= \prod_{i=1}^{N} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{(1-y^{(i)})} \end{aligned} \tag{21}$$

calculation of log likelihood plus derivative

2 Generalised Linear Model

The models discussed in the previous section may be expressed as subsets of a generalised **exponential family** of distributions which posit the target error distribution according to an arbitrary statistical model.

$$p(y; \eta) = b(y) \exp\left(\eta^T T(y)\right) - \alpha(\eta) \tag{22}$$

Define terms and express following distributions as GLM

2.1 Gaussian distribution

2.2 Bernoulli distribution

2.3 Multinomial distribution

2.4 Poisson distribution

2.5 Gamma distribution

2.6 Dirichlet distribution

3 Constructing a Generalised Linear Model

Lay out strategy for designing. Exemplify with OLS, logistic regression, softmax regression.

3.1 Ordinary least squares

3.2 Logistic regression

3.3 Softmax regression

4 Generative learning

A *generative model* aims to model data with a predictive capacity for both the feature and target variables of a data set. This is in contrast to *discriminative* models such as the regression models in the previous section which aim to model the target variable based on input features. Probabilistically, we may interpret generative learning as determining a probability distribution of $x \in \mathbb{X}$ given $y \in \mathbb{Y}$

$$p(x|y) \tag{23}$$

4.1 Gaussian discriminative analysis

4.2 Naive Bayes

The Naive Bayes assumption states that the conditional probability of individual elements in a feature set given the target are *independent*. That is to say, for $x \in \mathbb{R}^{\{n+1\}}$,

$$p(x|y) \equiv p(x_1, x_2, \dots x_{n+1}|y) = \prod_{i=1}^{n+1} p(x_i|y) \tag{24}$$

Alternatively, this may be more intuitively expressed

$$p(x_i|y) = p(x_i|y, x_j) \qquad \forall i, j \in \{1, 2, \dots, n+1\} \tag{25}$$

4.3 Learning Theory

4.3.1 Training error

The *training error* of a classifier describes its failure ratio in predicting output $y^{(i)}$ in the training set

$$\xi_n(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left[h(x^{(i)}) \neq y^{(i)}\right] \tag{26}$$

where the operator $\mathbb{I}$ maps a proposition to the binary value corresponding to its truth value.

$$\mathbb{I}\,[true] = 1 \qquad \mathbb{I}\,[false] = 0$$

A small training error does not necessarily an indication of the quality of a classifier. A classifier may have a low training error but may not generalise well to data outside the training set. Such a classifier is the result of *overtraining.*

4.3.2 Generalization error

Generalization is the application of a classifier to data beyond the training set. The *generalisation error* measures the failure ratio of predicting $y^{(i)}$ outside the training set. A good quality classifier is one that generalises well.

$$\xi(h) = \frac{1}{n'} \sum_{i=n}^{n+n'} \left[h(x^{(i)}) \neq y^{(i)} \right] \tag{27}$$