Introduction to Machine Learning **Fall 2017**

Lecture 1: Introductions

Lecturer: Ethan Fetaya *Scribe: Lily Li*

1.1 Administration

Our Instructor is Ethan Fetaya. A recommended textbook is *Understanding Machine Learning: from Theory to Algorithm* by Shal Shalev-Shwartz, Shal Ben-David.

For graduate students there is a project instead of a final (it is still worth 30%). Details are forth-coming.

Assignments are due by 22:00 on the day (observe that it is *NOT* due at 23:59).

1.2 Introduction to Machine Learning

Learning is: goal orientated skill acquisition. As computer scientists we use code to solve our problems, however, there are some complications: the solution maybe difficult to formalize, the task maybe mailable.

1.2.1 ML Categories

1. Supervised learning: correct outputs known. Given input X and output Y. We assume there is an distribution D on $X \times Y$. There is a also a lose function: $l : Y \times Y \to R$. We are given a set of m independent and identically distributed input-output pairs. What we want is a **hypothesis** function $f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1x_1 + \cdots + w_dx_d$ for $\mathbf{w} \in \mathbb{R}^{d+1}$ which minimizes the loss l.
2. Unsupervised Learning: find structure in data.
3. Online Learning: data keeps coming in there are no separate learning and validation data.
4. Reinforcement Learning: maximize future reward.

1.2.2 ML Viewpoints

1. Agnostic approach: minimize loss on unseen data.
2. Discriminative approach: fit with some parametric model.
3. Generative approach: fit $P(x, y : \theta)$ by parametric model then use the model to improve $P(x, y : \theta)$.

1.3 Linear Regression

Let the inputs be: $\mathbf{x} \in \mathbb{R}^d$. The outputs are $\mathbf{y}$ where $y \in \mathbb{R}$. And the input-output pair $(x^{(1)}, y^{(1)}), ..., (x^{(k)}, y^{(k)})$.

Any (fixed) transformation $\phi(x) \in \mathbb{R}$, run linear regressions with features $\phi(x)$. Observe that a polynomial $w_0 + w_1 x + \cdots + w_d x^d$ are actually linear in the parameters w_i! (Just consider the x^i as the features.)

The common loss is often set to $L_2 = (y - \hat{y})^2$. Unfortunately this loss model punishes infrequent large mistakes. The best prediction under this model is the mean. Another loss model is $L_1 = |y - \hat{y}|$. The best prediction here is the median. Another loss function is the Huber loss (which stitches together L_1 and L_2 losses — L_2 near the zero point) in a smooth way.

Note: we often include the bias in $\mathbf{x}$ as follows: $x^{(i)} = [1, x_1^{(i)}, ..., x_d^{(i)}]$ where our prediction is $\mathbf{x}^T\mathbf{w}$.

The target vector is $\mathbf{y} = [y^{(1)}, ..., y^{(N)}]^T$. The feature vectors are: $\mathbf{f}^{(j)} = [\mathbf{x}_j^{(1)}, ..., \mathbf{x}_j^{(N)}]^T$ and the design matrix $\mathbf{X}$ has the property that $\mathbf{X}_{i,j} = \mathbf{x}^(i)_j$. It is easiest think about $f^{(j)}$ as the j^{th} column and $x^{(i)}$ as the i^{th} row.

Theorem 1.1 *The optimal* $\mathbf{w}$*, with respect to* L_2*, is*

$$\mathbf{w}^* = \arg\min \sum_{i=1}^{N} \left(y^{(i)} - \mathbf{w}^T x^{(i)} \right)^2 = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Proof: Remark that the prediction vector is $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$. Consider the L_2 loss of $\mathbf{w}$. All we really need to do here is simplify this equation then take the derivative with respect to $\mathbf{w}$. Lets do just that

$$\begin{aligned} L_2(\mathbf{w}) &= \| \mathbf{y} - \hat{\mathbf{y}} \|^2 \\ &= \| \mathbf{y} - \mathbf{X}\mathbf{w} \|^2 \\ &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T \cdot (\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \mathbf{y}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} \end{aligned}$$

Now if we take the partial derivative with respect to $\mathbf{w}$ we get:

$$\nabla L(\mathbf{w}^*) = 2\mathbf{X}^T\mathbf{X}\mathbf{w}^* - 2\mathbf{X}^T\mathbf{y} = 0.$$

By rearranging the above equation we obtain $\mathbf{w}^* = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$. ■

Since our prediction is $\hat{\mathbf{y}} = \mathbf{X}^T\mathbf{y}$ and we calculated that the optimal weights is $\mathbf{X}^T\mathbf{X}\mathbf{w}^* = \mathbf{X}^T\mathbf{y}$, this gives us some information about the residual $r = \mathbf{y} - \hat{\mathbf{y}}$. In particular if we substitute for the value of $\hat{\mathbf{y}}$ and multiply by $\mathbf{X}^T$, then we see that $\mathbf{X}r = 0$. This means that the residual (think of this as the remainder) is orthogonal to each $\mathbf{x}^{(i)}$.

There is something else here about covariance that I didn't quite catch.

1.3.1 Regularization

Typically, if you over fit data, the model tends to have terms with large norm. Thus it is a good idea to introduce a regularizer term $R(\mathbf{w})$. The modified optimal model $w^* = \arg\min_{\mathbf{w}} L_S(\mathbf{w}) + R(\mathbf{w})$.

Commonly used regularizers include:

1. L_2 regularization:
$$R(\mathbf{w}) = \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}.$$
Note that the analytic solution is $\mathbf{w}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{y}$ (why). Normally we do not regularize the bias w_0 and we use validation/ cross-validation to find a good λ.

2. L_1 regularization:

$$R(\mathbf{w}) = \lambda \|\mathbf{w}\|_1 = \lambda \sum |w_i|.$$

This regularization is convex (Simple Gaussian Distribution) but has no analytic solution. It also tends to induce *sparse* solutions.

1.4 Tutorial: Probability

Sample space Ω: set of all possible outcomes of the experiment. Observation $\omega \in \Omega$ are sample outcomes, realizations, or elements. $E \subset \Omega$ are subsets of the sample space.

Definition 1.2 ***Joint Probability*** *of A and B is $P(A, B)$. Note that the joint probability is simply $P(A, B) = P(A|B)P(B) = P(B|A)P(A)$.*

Events A and B are ***conditionally independent*** *given C if $P(A, B|C) = P(B|A, C)P(A|C) = P(B|C)P(A|C)$.*

Definition 1.3 ***Marginalization (Sum Rule)*** *$P(X) = \sum_Y P(X, Y)$.*

Law of Total Probability *$P(X) = \sum_Y P(X, Y)$.*

Bayes' Rules can be reworded to be more useful for Machine Learning as follows:

$$\begin{aligned} P(\theta|x) &= \frac{P(x|\theta)P(\theta)}{P(x)} \\ \text{Posterior} &= \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}} \\ \text{Posterior} &\propto \text{Likelihood} \times \text{Prior} \end{aligned}$$

where $P(x|\theta)$ is the likelihood, $P(\theta)$ is the prior, and x is the evidence.

The difference between discrete and continuous random variables is the difference between summation and integration when marginalizing. Further:

Discrete: distribution defined by probability mass function (PMF). Marginalization: $p(x) = \sum_y p(x, y)$.

Continuous: distribution defined by probability density function (PDF). Marginalization: $p(x) = \int_y p(x, y)dy$.

The mean μ is the **First Moment** and the variance σ^2 is the **Second Moment**. Variance is defined as $\int_{-\infty}^{\infty} (x - \mu)^2 p(x)dx$. With a little bit of algebra, you can work out that $\mathsf{Var}[x] = E[x^2] - E[x]^2$.

1.4.1 Covariance Matrix

Let $\mathbf{x}$ be a D-dimensional vector and μ be a D-dimensional mean vector. Then Σ is a $D \times D$ covariance matrix with determinant $|\Sigma|$. Note that the (i, j) entry of Σ is the covariance of x_i, x_j:

$$\mathsf{Cov}(x_i, x_j) = E[(x_i - \mu_i)(x_j - \mu_j)] = E[(x_i x_j)] - \mu_i \mu_j.$$

Thus the diagonal entries are the variance of each element. Σ has the property that it is positive and semi-definite.

Whitening is a linear transformation that takes a vector of random variables with a known covariance changes these into a set of new random variables with the identity matrix as its covariance. This means that the random variables are all independent of one another. Formally, the random d-dimensional vector is $\mathbf{x} = (x_1, ..., x_d)^T$, the mean $\mu = E[\mathbf{x}] = (\mu_1, ..., \mu_d)^T$ and postive definite $d \times d$ covariance matrix $\mathsf{Cov}(\mathbf{x}) = \sigma$ into. $\mathbf{x}$ is changed into $\mathbf{z} = (z_1, ..., z_d)^T = W\mathbf{x}$ with *white* covariance matrix, $\mathsf{Cov}(\mathbf{z}) = \mathbf{I}$.