Lec 4: Linear Methods-II

Lecturer: C. V. Jawahar *Date: Aug. 13, 2018*

4.5 A Related Problem

In the last lecture we looked at the line fitting when we were predicting y_i from $\mathbf{x_i}$. We minimized an error in y_i. If you draw then in a 2D plane (when $\mathbf{x}$ was 1d) the error is parallel to the y axis. In general, the error is always defined with respect to y.

A very related problem is how do we fit a line that minimizes the orthogonal distance from the samples to the line? i.e.,

$$\min_{\mathbf{w}} d_{\perp}(\mathbf{x}_i, \mathbf{w})$$

Note that $\mathbf{x_i}$ is a point and $\mathbf{w}$ is a line.

Another problem that we are interested in is to find a "representative" for the set of samples that we have. Who should represent the set $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}$?.

How do we define the problem? The problem is to find an entity (such as a point, line, plane etc.) that can represent the set, with minimal loss in "information/content".

4.6 Point that minimizes the distance

Let $\mathbf{z}$ be a point that minimizes the sum of distance to all the given N points. How do we find $\mathbf{z}$?.

$$\min_{\mathbf{z}} \sum_{i=1}^{N} [\mathbf{z} - \mathbf{x_i}]^T [\mathbf{z} - \mathbf{x}_i]$$

Expanding

$$\min_{\mathbf{z}} \sum_{i=1}^{N} \mathbf{z^T z} + \mathbf{x_i^T x_i} - \mathbf{2 z^T x_i}$$

Differentiating with respect to $\mathbf{z}$ and equating to zero:

$$2\mathbf{z} = \sum_{1}^{N} 2\mathbf{x_i}$$

or $\mathbf{z}$ is nothing but our familiar mean $\mu = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_i}$.

This is quite intuitive. Here we only argued that mean minimizes the sum of distances. Also mean is a good representative of the samples. If we want to represent a set of samples with a single (constant) representation, then it has to be mean!!.

If we want to represent all the samples with a single dimension (not a constant), what it should be? Note that this is also equivalent to finding a new feature that can be used to "approximate" all the d features we have.

We want to look at this new feature as projections on to a line $\mathbf{w}$. as $z_i = \mathbf{w}^T \mathbf{x_i}$. Geometrically, if the structure of the data needs to be preserved (or loss of "information" is small), then we want to minimize the orthogonal distance to the line.

4.7 Minimizing Orthogonal Distance

We are interested in finding (i) a fixed point and (ii) a direction that can define our line. Let $\mathbf{w}$ define the direction. We know the fixed point as mean.

Let us first assume that mean is subtracted from all the samples. Now $\mathbf{x_i}^T \mathbf{x_i}$ is nothing but the square of the distance from the origin, and $\mathbf{w}^T \mathbf{x_i}$ is nothing but the projection of the $\mathbf{x}_i$ on $\mathbf{w}$.

Simple geometry (figure missing) tells us that the orthogonal distance we want to minimze is nothing but

$$\mathbf{x_i}^T \mathbf{x_i} - (\mathbf{w}^T \mathbf{x_i})^2$$

(ref: good old Pythagoras theorem)

Our problem now is

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \mathbf{x_i}^T \mathbf{x_i} - (\mathbf{w}^T \mathbf{x_i})^2$$

First term is independent of $\mathbf{z}$ so we could convert this into a maximization problem as:

$$\max_{\mathbf{w}} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x_i})^2$$

or more compactly

$$\max_{\mathbf{w}} \mathbf{w}^T (\mathbf{X}^T \mathbf{X}) \mathbf{w}$$

Please refer to the notations that we used for the MSE in the last lecture.

There is an issue here. Unconstrained optimization of this can lead to $\mathbf{w}$ increasing (to infinity). How do we

prevent? We need to add a constraint like $\mathbf{w}^T\mathbf{w} = 1$. This means that $\mathbf{w}$ is only a direction and what we want is a unit norm vector that maximizes our objective.

The popular method for introducing the constraint into the optimization problem is with lagrangians. (read more somewhere else.) Popular notation is λ. This modifies the problem as

$$\max_{\mathbf{w}} \mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w} - \lambda(\mathbf{w}^T\mathbf{w} - 1)$$

Now we use our simple trick of differentiating with respect to $\mathbf{w}$ and equating to zero.

$$(X^TX)\mathbf{w} = \lambda\mathbf{w}$$

Or $\mathbf{w}$ is the eigen vector of the $C = \mathbf{X}^T\mathbf{X}$. Since the mean was substracted from all the samples, it is easy to see that C is the covariance matrix.

Therefore the line that minimizes the orthogonal distance passes through the origin and has a direction of the first eigen vector of the covariance matrix.

4.7.1 Why first?

Why did we pick the first eigen vector? Note that C has many more eigen vectors. (Q: How many?) What is special for the first (largest) or principal one? (in general, it is assumed that eigen values are sorted in non-increasing order.)

We know that $(X^TX)\mathbf{w} = \lambda\mathbf{w}$ and what we want to maximize is $\mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w}$. Substituting, we realize that what we want to maximize is $\lambda\mathbf{w}^T\mathbf{w}$. If $\mathbf{w}$ is a unit vector, it is clear that what we want is the eigen vector corresponding to the largest eigen value.

Q: Can the eigen value be negative or imaginary for C? What are the properties of C?

Q: Is C is Positive Semi Definite (PSD) matrix? See the definitions and start.

Note: You may also see the notation of Σ for the covariance matrix.

4.8 Discussions

We now know how to optimize MSE (mean squared error). We also know how to minimize the orthogonal distance and fit a line/plane. In both cases, we formed an appropriate objective function. We minized the objective and found an expression to compute to the optimal solutions in one step.

- This is not always possible. Not all problems are this simple.
- We may not have all the data when we start. We continue to get data online in a streaming manner.
- we do not want to work with "huge" matrices in the casses of large data sets.

This points to the need of incremental techniques to address this class of problem.

4.9 Introduction to Gradient Descent

The most popular technique at this stage is gradient descent. i.e.,

- start with a random initialization of the solution.
- incrementally change the solution by moving in the negative gradient of the objective function.
- repeat the previous step until some convergence criteria is met.

Or the key equation for change in weight is:

$$\mathbf{w}^{k+1} \leftarrow \mathbf{w}^k - \eta\nabla J \tag{4.9}$$

Note that $\mathbf{w}$ is a vector. ∇J is a also a vector. Often $\mathbf{w^0}$ is a random vector. For some of the proofs later, we may use $\mathbf{w^0}$ to be zero vector $\mathbf{0}$.

4.9.1 An Intuitive Explanation

Let us look at a simple quadratic error/loss function. (Plot J vs $\mathbf{w}$). Let us assume that $\mathbf{w}$ is a scalar for simplicity.

Let us assume that we start with an arbitrary $\mathbf{w}^0$. How do we want to change $\mathbf{w}$? increase or decrease? (there are only two options for 1D case!!). This is same as negative gradient of the objective. But how much we should increase or decrease? This is the learning rate η. usually a small quantity adhocly set.

With small learning rate, the iterative algorithm takes more time to converge. With large learning rate there is a chance that the algorithm may get diverged or even oscillating.

4.10 Revisiting MSE

Let us now revisit the MSE we did in the last lecture. Our problem is to

$$\min_{\mathbf{w}} J = \min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w^T x_i})^2$$

The objective

$$\nabla J = \frac{2}{N}\sum_{i=1}^{N}(y_i - \mathbf{w^T x_i})(-\mathbf{x_i})$$

Q: Write psedocode for gradient descent based MSE; Implement and verify that the solutions of the closed form (last lecture) and this are the same.

This is a well behaved problem (a convex optimization). The solution is simple, and also we reach the same final vector independent of the initialization.

What should be the termination criteria? You can terminate when the changes in the solution is very small (say $< \epsilon$).

4.10.1 Practical Issues and Concerns

Gradient descent is a powerful (if not the most powerful at this moment) optimization tool that we will use in many situations in the past.

There are many concerns. How do we initialize? What learning rate we should choose? How do we terminate?

These concerns become more serious when the optimization problem is non-convex. They may not be serious at this stage.