CSE 475: Statistical Methods in AI **Monsoon 2019**

SMAI-M-2019 9: LM-III Gradient Descent - I

Lecturer: C. V. Jawahar *Date: DATE*

9.50 Challenges with Scale and Optimization

In the last lectures, we had seen how a specific problem of interest can be formulated as an optimization problem. We found closed form solutions to these problems.

However,

1. Such closed form solutions are not always available, specially for most of the optimization problems of our interest.
2. We should also expect many useful modification to such optimization problems in practice, for example, a researcher/practioner may modify the loss function to suite the specific problem requirement. Even if the original problem has a simple closed form solution, this new problem may not have. This handicaps the practical situation.
3. In many practical situations of interest, we need to work with large data sets. This makes the closed form solution, computationally unattractive.
4. More ..

This drives us to look for a simple, yet effective, optimization scheme — popularly known as gradient descent (GD) optimization.

9.51 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:

$$\mathbf{w}^{k+1} \leftarrow \mathbf{w}^k - \eta \nabla J \tag{9.14}$$

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 $\mathbf{0}$.

9.51.1 An Intuitive Explanation

Let us look at a simple quadratic error/loss function. (Plot J vs $\mathbf{w}$; Figure missing). 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.

9.52 Review of Matrix Calculus

Please note that we may be defferentiating scalar valued functions of vectors/matrices at different places. If you are not familiar with this, here is a quick reading:

Thomas Minka, "Old and New Matrix Algebra Useful for Statistics",

`https://tminka.github.io/papers/matrix/minka-matrix.pdf`

Those who are interested in also may read: The Matrix Cookbook

`https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf`

(worth reading, even if we may not read all the results in this course.)

9.53 Specific Examples

9.53.1 Revisiting MSE/Regression

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 $< \epsilon$).

9.53.2 Revisiting PCA

- Derive an iterative solution to PCA.
- How do we enfornce the constraint like $||\mathbf{w}|| = 1$?

9.54 Gradient Descent Procedure

9.54.1 Basic Gradient Descent Procedure

We are interested in finding the optimal $\mathbf{w}$ or $\mathbf{w}^*$ corresponding to the minima of $J(\mathbf{w})$. We know the gradient descent optimization procedure for this as:

1. Start with an arbitrary $\mathbf{w}$, $k = 0$
2. Improve $\mathbf{w}$ as
$$\mathbf{w}^{k+1} \leftarrow \mathbf{w}^k - \eta \nabla J$$
3. $k \leftarrow k + 1$
4. Repeat steps 2-3 until some convergence criteria is met.

Convergence criteria can be (i) the change in weight $\mathbf{w}$ or something similar. It is quite intuitive to see that the solution is improving in each iteration.

9.54.2 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. There are also many other concerns such as:

- How do we speed up the GD/optimization
- How do we avoid getting trapped in local minima? (or How to find a superior minima)
- Are there better update rules?
- etc.

9.54.3 Stochastic Gradient Descent Procedure

9.54.4 Variations in GD

Single Sample, Vs Batch Vs Mini Batch

Stochastic

SGD with Mini Batch

9.55 Convergence Analysis of GD

1. Convergence of GD does not imply global optima of the loss/objective.
2. Situations where divergence or oscilations can happen.
3. Formal analysis?