DS4400 Notes

DS4400 Notes 01/24

1. Convex functions:

   A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex iff $\forall \theta_1, \theta_2 \in \mathbb{R}^d$ and $\forall \alpha \in [0,1]$ we have $f(\alpha\theta_1 + (1-\alpha)\theta_2) \le \alpha f(\theta_1) + (1-\alpha) f(\theta_2)$

   In the special case $(d = 1)$ $f : \mathbb{R} \to \mathbb{R}$, f is convex iff $\forall \theta, f''(\theta) \ge 0$

   When the function is convex, **local min** $\equiv$ **global min**. When the system is not convex, we might find only a **local min** but not a **global min**

2. Dealing with non convex function:

   In gradient descent:

   (a) use larger ρ in the beginning and gradually decrease ρ with interation.

   (b) Run SGD/GD with multiple random initializaitons $\theta_1^{(0)}, \theta_2^{(0)} \ldots$ and keep the best solution.

3. $\min_\theta \sum_{i=1}^N (y_i - \theta^T x_i)^2 \triangleq J(\theta)$

   In linear regression, $J(\theta)$ is convex.

4. Robustness of Regression to outliers:

   (a) Run outlier detection algorithm, remove detected outliers, then run Linear Regression on remaining points.

   (b) Robust Regression cost function. $\min_\theta \sum_{i=1}^N e_i^2$, $e_i \triangleq y_i - \theta^T x_i$
   e^2 is extremly unhappy with large errors.
   we might use $|e|$ to replace the function. This might be more tolerance. Then, $\min_\theta \sum_{i=1}^N |y_i - \theta^T x_i|$

5. Exercise: D = $\{(x_1, y_1 = 100) \ldots (x_10, y_10 = 100), (x_{11}, y_{11} = 0), (x_{12}, y_{12} = 0)\}$

   e^2: $10(\theta - 100)^2 + 2\theta^2 \to$
   $\frac{\partial}{\partial\theta} = 20(\theta - 100) + 4\theta = 0 \to$
   $\theta = 83.3$

   $|e| : \min_\theta \sum_{i=1}^{12} |\theta - y_i| = 10|\theta - 100| + 2\theta$
   $(\theta \le 100) = \min_\theta 10(100 - \theta) + 2\theta$
   $= 1000 - 8\theta \to \theta = 100$
   $(\theta \ge 100) = \min_\theta 10(\theta - 100) + 2\theta$
   $= 12\theta - 1000 \to \theta = 100$

6. How to solve l1-norms cost functions?

   (a) No closed form

   (b) we need to be careful with gradient descent

   (c) We need to use convex programming toolboxs (convex optimizations)

7. Huber loss funct

$$l_\delta(e) = \begin{cases} \frac{1}{2}e^2 & |e| \le \delta \\ \delta|e| - \frac{\delta^2}{2} & |e| \ge \delta \end{cases}$$

$$\frac{\partial l_\delta(e)}{\partial e} = \begin{cases} e & -\delta \le ele\delta \\ \delta & e > \delta \\ -\delta & e < \delta \end{cases}$$

   in huber loss function, we don't have closed form solution but we can run gredient descent now.

8. Definition: Overfitting:

   Learning a system from traning data that does very well on training data itself (e.g, very low regression error on traning data), but performs poorly on test data.

9. Definition: Overfitting in Linear Regression

   $\Phi^T\Phi\theta = \Phi^T Y$
   $\Rightarrow \theta^* = (\Phi^T\Phi)^{-1}\Phi^T Y$

   $\text{rank}(\Phi^T\Phi) \leq \min\{rk(\Phi^T), rk(\Phi)\} = rk(\Phi) \leq \min\{N, d\}$

   $\Phi^T\Phi$ is $d \times d$ matrix, then rank is $\leq d$.

   Therefore, when $N < d$ it is not invertible which means we have multiple solutions and results in overfitting.

DS4400 Notes 01/28

1. Definition: Overfitting
   Refers to situation where the learned model does well on traning data and poorly on testing data.
   As d (dimension of system) increases, then training error godes down (can be exactly ZERO for sufficiently large d)

2. In Linear regression:

$$\min \sum_{i=1}^{n} (\theta^T\phi(x_i) - y_i)^2$$

   set the derivative to 0 and we find

$$\Phi^T\Phi\theta = \Phi^T Y$$

   Then $\theta^* = (\Phi^T\Phi)^{-1}\Phi^T Y$

   When is it the case that $\Phi^T\Phi$ is not invertible?
   Since $\Phi^T\Phi \in \mathbb{R}^{N\times d}$

$$rk(\Phi^T\Phi) \leq rk(\Phi) \leq min\{N, d\}$$

   $\Phi^T\Phi \in \mathbf{R}^{d\times d}$ is invertible when $rk(\Phi^T\Phi) = d$. Therefore, when $N < d, rk(\Phi^T\Phi) = N$, $\Phi^T\Phi$ is not invertible. There will be infinitely many solutions for θ.

   Generally, need sufficient # samples

3. Test overfitting.
   If $\Phi^T\Phi$ is not invertible,
   $\exists v \neq 0, \Phi^T\Phi v = 0$
   $\Rightarrow \theta^* + \alpha v$ is also a solution for any $\alpha \in R$
   $\Phi^T\Phi(\theta^* + \alpha v) = \Phi^T\Phi\theta^* + \Phi^T\Phi(\alpha v)$
   $= \Phi^T\Phi\theta^* + \alpha\Phi^T\Phi v$
   $= \Phi^T\Phi\theta^* = \Phi^T Y$

   We can find large α so that θ^* have extremly large entries.

   Generally, if the entries are very large (abs) we might have overfitting

4. Treat overfitting
   We want to change regreession optimization to prevent θ from very large terms.

   then we change the cost function:

$$\min_{\theta} \sum_{i=1}^{N} (\theta^T\phi(x_i) - y_i)^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

   λ: regularization parameter (> 0)
   $\sum_{j=1}^{d} \theta_j^2$: regularizer.
   $\lambda \to 0$: back to overfitting
   $\lambda \to \infty : \theta^* = 0$, underfitting

   (a) closed-form
   $\frac{\partial J}{\partial \theta}$
   $= 2\Phi^T(\Phi\theta - Y) + \lambda\frac{\partial \sum_{j=1}^{N} \theta_j^2}{\partial \theta}$
   $= 2\Phi^T(\Phi\theta - Y) + 2\lambda\theta$
   Let it be zero:

$$\Phi^T\Phi\theta + \lambda\theta = \Phi^T Y$$

$$(\Phi^T\Phi + \lambda I_d)\theta = \Phi^T Y$$

   Then $\theta^* = (\Phi^T\Phi + \lambda I_d)^{-1}\Phi^T Y$

   (b) Gradient descent
   Find initial $\theta^{(0)}$
   $\theta^t = \theta^{(t-1)} - \rho\frac{\partial J}{\partial \theta}|_{\theta^{(t-1)}}$
   $= \theta^{(t-1)} - 2\Phi^T(\Phi\theta^{(t-1)} - Y) + 2\lambda\theta^{(t-1)}$

5. Hyperparameter Tunning

   GD: set learning rate ρ

   Robust Reg: Huber loss δ

   overfitting and regularization: λ

   ρ, δ, λ = hyperparameters

   How to pick hyperparameters?

   BAD APPROACH 1:

(a) pick some set of possible $\lambda_i \in \{\lambda_1, \lambda_2 \dots\}$
Run regression with λ_i and find θ_i^*
Measure regression error:

$$\epsilon_{tr}(\lambda) = \sum_{i=1}^{N} ((\theta^*(\lambda))^T x_i - y_i)^2$$

To sum: just find λ for which $\epsilon_{tr}(\lambda)$ is minimum

This approach is setting λ back to 0

Test data needed!!!

(a) We need to Train λ_i on **training set** to minimize the cost function

$$2\Phi^T(\Phi\theta - Y) + 2\lambda\theta$$

to find θ_i^*

(b) Measure regression error on the **hold-out set** D^{ho}

$$\epsilon_{tr} = \sum_{x_i, y_i \in D^{ho}} (y_i - (\theta^*(\lambda))^T x_i)^2$$