Review quizzes

- You are given the training data, and you assume the relationship between the target variable and the features are linear, i.e., $y_i \approx \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots + \theta_d x_{id} \; \forall i$
  - How to obtain $\theta_0, \theta_1, \ldots, \theta_d$ such that $\sum (y_i - \widehat{y}_i)^2$ is minimized?
- Briefly explain gradient descent
- Why do we want to limit the magnitude of the learned parameters in linear regression?

Review quizzes

- What is regularization terms and why do we need them?
- What is Lasso?
- What is Ridge regression?
- Compared to Ridge regression, why does Lasso tend to shrink some parameters to zero?

A toy example of gradient descent for linear regression

- Assumption: $\hat{y}_i = \theta_0 + \theta_1 x_i$
  - $J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{2n}\sum_{i=1}^{n}(\theta_0 + \theta_1 x_i - y_i)^2$
- Gradient descent procedure
  1. Start with random values
     - $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)} = \left(\theta_0^{(0)}, \theta_1^{(0)}\right)$
  2. Slightly move θ_0 and θ_1 to reduce $J(\boldsymbol{\theta})$
     - $\theta_i^{(k+1)} = \theta_i^{(k)} - \alpha \left.\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_i}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}}$
     - $k = k + 1$
  3. Keep doing step 2 until converges

A toy example of gradient descent for linear regression (cont')

- Derivatives

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} = \frac{1}{n}\sum(\theta_0 + \theta_1 x_i - y_i)$$
$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} = \frac{1}{n}\sum(\theta_0 + \theta_1 x_i - y_i)x_i$$

- Substitute into the gradient descent equation

$$\theta_0^{(k+1)} = \theta_0^{(k)} - \alpha\frac{1}{n}\sum\left(\theta_0^{(k)} + \theta_1^{(k)} x_i - y_i\right)$$
$$\theta_1^{(k+1)} = \theta_1^{(k)} - \alpha\frac{1}{n}\sum\left(\theta_0^{(k)} + \theta_1^{(k)} x_i - y_i\right)x_i$$

A toy example of gradient descent for linear regression (cont')

- Generate the experimental data:

$$y_i = 3 + 5x_i + \epsilon, \text{ where}$$

$$\epsilon \sim N(0, 0.1)$$

x_i	y_i
1	7.958791127
2	13.04018641
3	18.00001875
4	23.03532824
5	28.10760623
6	33.05051692
7	38.11747573
8	42.91467587
9	47.95638784
10	53.09378294

$\alpha = 0.05$

k	$\hat{\theta}_0^{(k)}$	$\hat{\theta}_1^{(k)}$
0	0	0
10	0.4518289	1.58461955
50	1.539053893	4.324536645
100	2.174594941	4.979231032
200	2.692306641	5.047381882
300	2.866051014	5.026263422
400	2.926347718	5.017705744
500	2.947328771	5.014694799
1000	2.958471874	5.013094274
1500	2.95852878	5.0130861
2000	2.958529071	5.013086058

Gradient descent by Excel

972	4.0419507	6.00201285
973	4.04195199	6.00201266
974	4.04195326	6.00201248
975	4.04195452	6.0020123
976	4.04195576	6.00201212
977	4.041[illegible]957	6.00201194
978	4.04195[illegible]22	6.[illegible]020117[illegible]
979	4.04195942	6.00201159
980	4.04196062	6.00201142
981	4.0419618	6.00201125
982	4.04196297	6.00201108
983	4.04196413	6.00201092
984	4.04196527	6.00201075
985	4.0419664	6.00201059
986	4.04196752	6.00201043
987	4.04196863	6.00201027
988	4.04196973	6.00201011
989	4.04197082	6.00200996
990	4.04197189	6.0020098
991	4.04197296	6.00200965
992	4.04197401	6.0020095
993	4.04197505	6.00200935
994	4.04197608	6.0020092
995	4.0419771	6.00200905
996	4.04197811	6.00200891
997	4.04197911	6.00200877
998	4.04198009	6.00200862
999	4.04198107	6.00200848
1000	4.04198204	6.00200835
1001	4.041983	6.00200821
1002	4.04198394	6.00200807
1003	4.04198488	6.00200794
1004	4.04198581	6.0020078
1005	4.04198672	6.00200767
1006	4.04198763	6.00200754
1007	4.04198853	6.00200741
1008	4.04198942	6.00200728
1009	4.0419903	6.00200716
1010	4.04199117	6.00200703
1011	4.04199203	6.00200691
1012	4.04199288	6.00200679
1013	4.04199372	6.00200667
1014	4.04199456	6.00200655
1015	4.04199538	6.00200643
1016	4.0419962	6.00200631
1017	4.04199701	6.00200619
1018	4.04199781	6.00200608
1019	4.0419986	6.00200597

Most equations have no closed form solution

- Polynomial equations with degree lower than 4 have closed-form solutions
  - E.g., $ax^2 + bx + c = 0 \Rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
- Polynomial equations with real-valued coefficients have no closed-form solution
  - E.g., $x^5 - x + 1 = 0 \Rightarrow x = ?$
- Linear programming generally has no closed-form solution
  - Minimize $c^T x$ subject to $a_i^T x \leq b_i, i = 1, \ldots, m$
- In ML (and many CS sub-domains), getting closed-form solutions may not be important
  - Floating point is imprecise anyways
    - E.g., 0.1 + 0.2 == 0.3?

What could be included for your mid-term presentation

- Reminder: midterm project proposal on **11/10**
- If you want to apply ML on certain topic, the presentation may include (but not limit to)
  - Introduction
    - Background knowledge
    - The motivation of your project
    - The overall vision/goal of your project
  - Dataset
    - What is your target dataset
    - How to collect the dataset
  - Current progress
    - E.g., naïve method(s) as baseline solution(s)
  - A plan/schedule for the following weeks
  - What do you expect to accomplish by the end of the semester
- If you want to develop a new method, you may include
  - Motivation
  - Current ideas

Exercise 3: linear regression and gradient descent

- Requirement
  - Coding (85%)
    - Implement the "gradient_descent" function
    - Use your linear regression script to predict the target (Y1) of the Energy Efficiency dataset
      - https://archive.ics.uci.edu/ml/datasets/Energy+efficiency
      - Separate the data into training (50%) and test (50%) datasets.
    - We will give start code; you can only modify certain functions
  - Report (15%)
    - Report the R^2 score for both the training and the testing data
    - A brief discussion of the results
- Please submit your code and report to LMS
- Due date: 11/2 23:59:59.