School of Computing and Information Systems
The University of Melbourne
COMP30027 MACHINE LEARNING (Semester 1, 2019)

Tutorial exercises: Week 7

1. What is the **Gradient Descent** method, and why is it important?
2. What is **Regression**? How is it similar to **Classification**, and how is it different?
   (a) What is **Linear Regression**? In what circumstances is it desirable, and it what circumstances is it undesirable?
   (b) How do we build a (linear) regression model? What is **RSS** and what advantages does it have over (some) alternatives?
3. Recall that the update rule for Gradient Descent with respect to RSS is as follows:

$$\beta_k^{i+1} \quad := \quad \beta_k^i + 2\alpha \sum_{j=1}^{N} x_{jk}(y_j - \hat{y_j^i})$$

   Build a Linear Regression model, using the following instances:

| x | y |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 2 | 3 |

4. What is **Logistic Regression**?
   (a) How is Logistic Regression similar to **Naive Bayes** and how is it different? In what circumstances would the former be preferable, and in what circumstances would the latter?
   (b) What is "logistic"? What are we "regressing"?
   (c) How do we train a Logistic Regression model? In particular, what is the significance of the following:

$$\text{argmax}_\beta \sum_{i=1}^{n} y_i \log h_\beta(x_i) + (1 - y_i) \log(1 - h_\beta(x_i))$$

1. What is the **Gradient Descent** method, and why is it important?

Gradient Descent is a machinism of finding the MINIMIUM of mutlivariate function by finding the partial deriviative.

Many Applications : finding the regression weights which minimize error function (RSS)

2. What is **Regression**? How is it similar to **Classification**, and how is it different?

   (a) What is **Linear Regression**? In what circumstances is it desirable, and it what circumstances is it undesirable?

   (b) How do we build a (linear) regression model? What is **RSS** and what advantages does it have over (some) alternatives?

When class is Continuous (numeric) → Regression → can't get likelihood of each class.

when class is nominal → classification

(a) build a Linear model to predict target value by finding a weight for each attribute $\sum_i w_i a_i$

(b) Using Gradient Descent. to learn weights with respect to error function

We always assume error function is Convex, so we can always find a solution. RSS : minimizes the sum of the square difference between true value and prediction value

3. Recall that the update rule for Gradient Descent with respect to RSS is as follows:

$$\beta_k^{i+1} \quad := \quad \beta_k^i + 2\alpha \sum_{j=1}^{N} x_{jk}(y_j - \hat{y_j^i})$$

Build a Linear Regression model, using the following instances:

| | x | y |
|---|---|---|
| a | 1 | 1 |
| b | 2 | 2 |
| c | 2 | 3 |

Init,

learning rate

$\hat{y} = 0 + 0x$. $\beta = \langle 0, 0 \rangle$ $\alpha = 0.05$

first Round $i = 0$.

a: $\hat{y} = 0$ $y = 1$ error $= y - \hat{y} = 1$

b: $\hat{y} = 0$ $y = 2$ error $= 2$

c: $\hat{y} = 0$ $y = 3$ error $= 3$

$\beta_0^1 = \beta_0^0 + 2 \times 0.05 \times \sum_i x_{i0} (y_j - \hat{y_j})]$

$= 0 + 2 \times 0.05 \times [1 \times 1 + 1 \times 2 + 1 \times 3]$

$= 0.6$

$\beta_1^1 = \beta_1^0 + 2 \times 0.05 \times \sum_i x_{i1} (y_j - \hat{y_j})$

$= 0 + 2 \times 0.05 \times [1 \times 1 + 2 \times 2 + 2 \times 3]$

$= 1.1$

Now: $\hat{y} = 0.6 + 1.1 x_1$

Second Round $i = 1$

a: $\hat{y} = 1.7$ $y = 1$ error $= -0.7$

b: $\hat{y} = 2.8$ $y = 2$ error $= -0.8$

c: $\hat{y} = 2.8$ $y = 3$ error $= 0.2$

$\beta_0^{11} = \beta_0^1 + 0.1 \times [1 \times -0.7 + 1 \times -0.8 + 1 \times 0.2]$

$= 0.6 - 0.13 = 0.47$

$\beta_1^{11} = \beta_1^1 + 0.1 \times [1 \times -0.7 + 2 \times -0.8 + 2 \times 0.2]$

$= 1.1 - 0.19 = 0.91$

Now: $\hat{y} = 0.47 + 0.91 x_1$

after several round ... $\hat{y} = 1.5x - 0.5$

How to set learning rate?

- In first round, RSS is increasing, then we start again with another $\alpha$

- In few rounds later, RSS increasing.
  - Convergence
  - If we want more accurate take current $\beta$, and choose a smaller $\alpha$.

What is **Logistic Regression**?

(a) How is Logistic Regression similar to **Naive Bayes** and how is it different? In what circumstances would the former be preferable, and in what circumstances would the latter?

(b) What is "logistic"? What are we "regressing"?

(c) How do we train a Logistic Regression model? In particular, what is the significance of the following:

$$\text{argmax}_\beta \sum_{i=1}^{n} y_i \log h_\beta(x_i) + (1 - y_i) \log(1 - h_\beta(x_i))$$

4. Logistic Regression: We build a (Linear) regression model, when the target is 1 for positive class. 0 for negative class.

(a) Similarity:

1. Both are attempt to find class by calculating $T(c|T)$

Difference:

1. NB makes assumption: assume Independence.

2. LG builds model directly, since don't need to generate class probability only discriminate classes.

(b) Logistic function: $\frac{1}{1+e^{-m}}$

(c) → we want 1 → positive, 0 → negative

use gradient Ascent to find $\beta$ which maximize this objective function.