CPSC 340: Tutorial 8

Aaron Mishkin

UBC

W2017

Today

1. Multi-Class Classification
2. Assignment Code
3. MAP Estimation

Multi-Class Classification

Recall the problem statement for classification:

- There is a feature vector x_i and a categorical label y_i for every example $i = 1...n$.

Multi-Class Classification

Recall the problem statement for classification:

- There is a feature vector x_i and a categorical label y_i for every example $i = 1...n$.
- In binary classification every label y_i was either -1 or 1.

Multi-Class Classification

Recall the problem statement for classification:

- There is a feature vector x_i and a categorical label y_i for every example $i = 1...n$.
- In binary classification every label y_i was either -1 or 1.
- In multi-class classification every label y_i is one of k classes. For example:

Multi-Class Classification

Recall the problem statement for classification:

- There is a feature vector x_i and a categorical label y_i for every example $i = 1...n$.
- In binary classification every label y_i was either -1 or 1.
- In multi-class classification every label y_i is one of k classes. For example:
  - $y_i \in \{1, 2, ..., k\}$

Multi-Class Classification

Recall the problem statement for classification:

- There is a feature vector x_i and a categorical label y_i for every example $i = 1...n$.
- In binary classification every label y_i was either -1 or 1.
- In multi-class classification every label y_i is one of k classes. For example:
  - $y_i \in \{1, 2, ..., k\}$
  - $y_i \in \{\text{dog}, \text{cat}, \text{canary}...\}$

Multi-Class Classification

Recall the problem statement for classification:

- There is a feature vector x_i and a categorical label y_i for every example $i = 1...n$.
- In binary classification every label y_i was either -1 or 1.
- In multi-class classification every label y_i is one of k classes. For example:
  - $y_i \in \{1, 2, ..., k\}$
  - $y_i \in \{\text{dog}, \text{cat}, \text{canary}...\}$
  - etc.

Multi-Class Classification

Recall the problem statement for classification:

- There is a feature vector x_i and a categorical label y_i for every example $i = 1...n$.
- In binary classification every label y_i was either -1 or 1.
- In multi-class classification every label y_i is one of k classes. For example:
  - $y_i \in \{1, 2, ..., k\}$
  - $y_i \in \{\text{dog}, \text{cat}, \text{canary}...\}$
  - etc.
- Our basic goal remains the same: train a model to correctly predict classes for new examples.

Naive Approach: One vs All

- A separate binary classifier is trained for each class in "One vs All" logistic regression.

Naive Approach: One vs All

- A separate binary classifier is trained for each class in "One vs All" logistic regression.
- This requires fitting k weight vectors. We use w_c to refer the weight vector trained to predict class c.

Naive Approach: One vs All

- A separate binary classifier is trained for each class in "One vs All" logistic regression.
- This requires fitting k weight vectors. We use w_c to refer the weight vector trained to predict class c.
- The parameter for the overall model is the matrix W:

$$W = \begin{bmatrix} \vdots & \vdots & & \vdots \\ w_1 & w_2 & \dots & w_k \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

Naive Approach: One vs All

- A separate binary classifier is trained for each class in "One vs All" logistic regression.
- This requires fitting k weight vectors. We use w_c to refer the weight vector trained to predict class c.
- The parameter for the overall model is the matrix W:

$$W = \begin{bmatrix} \vdots & \vdots & & \vdots \\ w_1 & w_2 & \dots & w_k \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

- The dimensions of W are $d \times k$.

Prediction

- To predict on example x_i we compute $w_c^\top x_i$ for every class c and predict $y_i = argmax_c\{w_c^\top x_i\}$.

Prediction

- To predict on example x_i we compute $w_c^\top x_i$ for every class c and predict $y_i = argmax_c\{w_c^\top x_i\}$.
- This is equivalent to $y_i = argmax_c\{x_i^\top W\}$.

Prediction

- To predict on example x_i we compute $w_c^\top x_i$ for every class c and predict $y_i = argmax_c\{w_c^\top x_i\}$.
- This is equivalent to $y_i = argmax_c\{x_i^\top W\}$.
  - We are taking the maximum over elements of the row vector.

Prediction

- To predict on example x_i we compute $w_c^\top x_i$ for every class c and predict $y_i = argmax_c\{w_c^\top x_i\}$.
- This is equivalent to $y_i = argmax_c\{x_i^\top W\}$.
  - We are taking the maximum over elements of the row vector.
- To predict multiple examples at once:

$$Y = argmax_c\{XW\}$$

.

Prediction

- To predict on example x_i we compute $w_c^\top x_i$ for every class c and predict $y_i = argmax_c\{w_c^\top x_i\}$.
- This is equivalent to $y_i = argmax_c\{x_i^\top W\}$.
  - We are taking the maximum over elements of the row vector.
- To predict multiple examples at once:

$$Y = argmax_c\{XW\}$$

.

- We will use w_{y_i} to refer to the column of W corresponding to the correct label for x_i.

Softmax Loss

- What can go wrong with "one vs all" logistic regression?

Softmax Loss

- What can go wrong with "one vs all" logistic regression?
- The **softmax loss** is the objective used in multi-class logistic regression:

$$f(W) = \sum_{i=1}^{n} \left[-w_{y_i}^\top x_i + \log\left(\sum_{c=1}^{k} \exp(w_c^\top x_i)\right)\right]$$

Softmax Loss

- What can go wrong with "one vs all" logistic regression?
- The **softmax loss** is the objective used in multi-class logistic regression:

$$f(W) = \sum_{i=1}^{n} \left[-w_{y_i}^\top x_i + \log\left(\sum_{c=1}^{k} \exp(w_c^\top x_i) \right) \right]$$

- The log-sum-exp is a smooth approximation to the max function. This means soft max loss is a differentiable approximation to:

$$\sum_{i=1}^{n} \left[-w_{y_i}^\top x_i + \max_{c=1}^{k}(w_c^\top x_i) \right]$$

Softmax Loss

- What can go wrong with "one vs all" logistic regression?
- The **softmax loss** is the objective used in multi-class logistic regression:

$$f(W) = \sum_{i=1}^{n} \left[-w_{y_i}^\top x_i + \log\left(\sum_{c=1}^{k} \exp(w_c^\top x_i) \right) \right]$$

- The log-sum-exp is a smooth approximation to the max function. This means soft max loss is a differentiable approximation to:

$$\sum_{i=1}^{n} \left[-w_{y_i}^\top x_i + \max_{c=1}^{k}(w_c^\top x_i) \right]$$

- In other words, we want $w_c^\top x_i$ to be largest for the correct label $c = y_i$.

Indicator Functions

- What are we indicating? Usually set membership or the satisfaction of a condition.

Indicator Functions

- What are we indicating? Usually set membership or the satisfaction of a condition.
- Indicator functions (also characteristic functions) are functions of the form:

$$I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

Indicator Functions

- What are we indicating? Usually set membership or the satisfaction of a condition.
- Indicator functions (also characteristic functions) are functions of the form:

$$I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

- In the case of softmax loss, the following indicator function is useful:

$$I_{y_i}(c) = \begin{cases} 1 & \text{if } c = y_i \\ 0 & \text{otherwise} \end{cases}$$

Indicator Functions

- What are we indicating? Usually set membership or the satisfaction of a condition.
- Indicator functions (also characteristic functions) are functions of the form:

$$I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

- In the case of softmax loss, the following indicator function is useful:

$$I_{y_i}(c) = \begin{cases} 1 & \text{if } c = y_i \\ 0 & \text{otherwise} \end{cases}$$

- Keep this in mind when deriving the gradient.

Today

1. Multi-Class Classification
2. Assignment Code
3. MAP Estimation

Question One Code

Let's take a look at the assignment code!

Today

1. Multi-Class Classification
2. Assignment Code
3. MAP Estimation

MAP with Gaussian Prior on W

- Recall that we denote the normal distribution by $N(\mu, \sigma^2)$
- Recall that the p.d.f for the normal distribution is:

$$p(x) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp(-\frac{(\mu - x)^2}{2\sigma^2})$$

- Assume $y_i | x_i, w \sim N(w^T x_i, 1)$ and $w \sim N(0, 1)$.
- We can show that MAP estimation yields L_2 regularized least squares regression under these assumptions!

$$\tag{1}$$