Instruction for Logistic Regression

1. General Introduction

The tags are in Begin-Inside-Outside (BIO) format. Tags starting with B indicating the beginning of a piece of information, tags beginning with I indicating a continuation of a previous type of information, and O tags indicating words outside of any information chunk. For example, in the sentence below, Newark is the departure city (fromloc.city_name), Los Angeles is the destination (toloc.city_name), and the user is requesting flights for Wednesday (depart_date.day_name). In this homework, you will treat the BIO tags as arbitrary labels for each word.

Gradient decent for this program can be defined as:

$$\nabla_{\boldsymbol{\theta}^{(k)}} J^{(i)}(\boldsymbol{\theta}) = -\left(\mathbb{I}(y^{(i)}=k) - \frac{\exp\left(\boldsymbol{\theta}^{(k)}{}^{\top}\mathbf{x}^{(i)}\right)}{\sum_{j=1}^{K} \exp\left(\boldsymbol{\theta}^{(j)}{}^{\top}\mathbf{x}^{(i)}\right)}\right)\mathbf{x}^{(i)}$$

The possibility for each sample x can be defined as:

$$\mathbb{P}(y^{(i)}=k|\mathbf{x}^{(i)}) = \frac{\exp(\boldsymbol{\theta}^{(k)}{}^{\top}\mathbf{x}^{(i)})}{\sum_{j=1}^{K} \exp(\boldsymbol{\theta}^{(j)}{}^{\top}\mathbf{x}^{(i)})}$$

Negative log likelihood can be defined as:

$$\begin{aligned} J(\boldsymbol{\theta}) &= -\frac{1}{N}\ell(\boldsymbol{\theta}) \\ &= -\frac{1}{N}\log\left(\prod_{i=1}^{N}\prod_{k=1}^{K}\mathbb{P}(y^{(i)}=k|\mathbf{x}^{(i)})^{\mathbb{I}(y^{(i)}=k)}\right) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{I}(y^{(i)}=k)\log\frac{\exp(\boldsymbol{\theta}^{(k)}{}^{\top}\mathbf{x}^{(i)})}{\sum_{j=1}^{K}\exp(\boldsymbol{\theta}^{(j)}{}^{\top}\mathbf{x}^{(i)})} \end{aligned}$$

Gradient update follows the formula below:

$$\boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^{(k)} - \eta\left[\nabla_{\boldsymbol{\theta}^{(k)}} J^{(i)}(\boldsymbol{\theta})\right]$$

2. Input files:

Eight command-line arguments:<train input> <validation input> <test input> <train out> <test out> <metrics out> <num epoch> <feature flag>. These arguments are described in detail below:

1. <train input>: path to the training input .tsv file.
2. <validation input>: path to the validation input .tsv file
3. <test input>: path to the test input .tsv file

4. <train out>: path to output .labels file to which the prediction on the training data should be written
5. <test out>: path to output .labels file to which the prediction on the test data should be written.
6. <metrics out>: path of the output .txt file to which metrics such as train and test error should be written
7. <num epoch>: integer specifying the number of times SGD loops through all of the training data (e.g., if <num epoch> equals 5, then each training example will be used in SGD 5 times).
8. <feature flag>: integer taking value 1 or 2 that specifies whether to construct the Model 1 feature set or the Model 2 feature set —that is, if feature_flag==1 use Model 1 features; if feature_flag==2 use Model 2 features

3. Models:

1. Model 1 p(yt | wt, theta): This model defines a probability distribution over the current tag yt using the parameters and a feature vector based on only the current word wt. This model should be used when <feature flag> is set to 1.

2. Model 2 p(yt | wt 1, wt, wt+1, theta): This model defines a probability distribution over the current tag yt using the parameters and a feature vector based on the previous word wt 1, the current word wt, and the next word wt+1 in the sequence. This model should be used when <feature flag> is set to 2.

For model 2, boundary cases are handled when you are either looking at the first word or the last word of the sequence. Extra beginning of sentence (‘BOS’) feature and end of sentence (‘EOS’) feature to use in place of a word in the wt 1 or wt+1 position when t is the first or last word in a sentence, respectively.