CS 594: Advanced Machine Learning **Due: 17:00, Mar 6, 2018**

Project 1: Conditional Random Fields for Structured Output Prediction

Student: George Maratos Ashwani Khemani *Email: gmarat2@uic.edu, akhema2@uic.edu*

1a Solution

We can simplify $\log p(y^t, X^t)$ as follows:

$$\log \frac{1}{Z_X} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{y_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{y_s, y_{s+1}}\right) = \sum_{s=1}^{m} \langle \mathbf{w}_{y_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{y_s, y_{s+1}} - \log Z_x \tag{1}$$

The $\nabla_{\mathbf{w}_y} \sum_{s=1}^{m} \langle \mathbf{w}_{y_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{y_s, y_{s+1}}$ is the following:

$$\sum_{s=1}^{m} [\![y_s = y]\!] x_s^t \tag{2}$$

This is because while taking the derivative of a dot product involving w_y, X_s will only appear whenever $y_s = y$. And, the sum of transitions will disappear because it does not depend on w.

The $\nabla_{\mathbf{w}_y} \log Z_X$ is computed via the chain rule in the following way:

$$\nabla_{\mathbf{w}_y} \log Z_X = \frac{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)} * \sum_{s=1}^{m} [\![y_s = y]\!] x_s^t \tag{3}$$

This can be further reduced, by substituting $p(y|X)$ into equation 3 and rearanging the sums, in the following way:

$$\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} p(y|X) \sum_{s=1}^{m} [\![y_s = y]\!] x_s^t = \sum_{s=1}^{m} \sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} p(y|X) [\![y_s = y]\!] x_s^t \tag{4}$$

Finally, we recognize that the inner summation is a marginalization over y except the label we are taking the gradient against. Therefore we can further reduce the equation to:

$$\sum_{s=1}^{m} p(y_s = y|X^t))x_s^t \tag{5}$$

Using equations 2 and 5, the gradient is

$$\nabla_{\mathbf{w}_y} \log p(\mathbf{y}^t|X^t) = \sum_{s=1}^{m} (\llbracket y_s^t = y \rrbracket - p(y_s = y|X^t))\mathbf{x}_s^t, \tag{6}$$

Now computing gradient for T

The $\nabla_{T_ij} \sum_{s=1}^{m} \langle \mathbf{w}_{y_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{y_s,y_{s+1}}$ is the following:

$$\sum_{s=1}^{m-1} \llbracket y_s = i, y_{s+1} = j \rrbracket \tag{7}$$

This is because when we differentiate with respect to T_{ij}, the only terms that remain are the ones that specify a transition between i and j. There is only a single weight that lies at this transition point so when we differentiate it, it becomes 1. Also, the dot product goes away because it does not depend on T.

The $\nabla_{T_{ij}} \log Z_X$ is computed via the chain rule in the following way:

$$\nabla_{T_{ij}} \log Z_X = \frac{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s,\hat{y}_{s+1}}\right)}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s,\hat{y}_{s+1}}\right)} * \sum_{s=1}^{m-1} \llbracket y_s = i, y_{s+1} = j \rrbracket \tag{8}$$

This can be further reduced, by substituting $p(y|X)$ into equation 8 and rearanging the sums, in the following way:

$$\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} p(y|X) \sum_{s=1}^{m-1} \llbracket y_s = i, y_{s+1} = j \rrbracket = \sum_{s=1}^{m-1} \sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} p(y|X) \llbracket y_s = i, y_{s+1} = j \rrbracket \tag{9}$$

Finally, we recognize that the inner summation is a marginalization over y except the label we are taking the gradient against. Therefore we can further reduce the equation to:

$$\sum_{s=1}^{m-1} p(y_s = i, y_{s+1} = j|X^t) \tag{10}$$

Using equations 7 and 9 the gradient is:

$$\sum_{s=1}^{m-1} [\![y_s = i, y_{s+1} = j]\!] - p(y_s = i, y_{s+1} = j|X^t) \tag{11}$$

1b Solution

The features will take the following form. If they are a feature that depends on the letter's label, they will look as follows:

$$x_i [\![y_i = j]\!] \tag{12}$$

where i will be between $1...m$ and j will be between 1...26. If the feature depends on the transition between two letters then that feature will take the following form.

$$[\![y_i = k, y_i + 1 = z]\!] \tag{13}$$

where i is between $1...m$ and k, z are between 1...26

The gradient of $\log Z_x$ is described in equations 3 and 8. The feature function $\phi(x)$ will "place" the input in specific positions so that the correct dot product and transitions are computed. For example, the feature function enables the selection of w_7 to be used in $\langle x_i, w_7 \rangle$ when $y_i = 7$.

Considering features that depend on the label first. The exectation of some features with respect to $p(\mathbf{y}|X)$ will be:

$$\sum_{s=1}^{m} p(\mathbf{y}|X)\phi(X) \tag{14}$$

And this is equivalent to:

$$\sum_{s=1}^{m} p(\mathbf{y}|X^t)x_s [\![y_s = y]\!] \tag{15}$$

The feature function (via the indicator) will only be non-zero for terms s.t. $y_s = y$ and the others will go to zero. Therefore we can say that the above is equivalent to equation 3 because the $p(\mathbf{y}|X)$

term will become $p(y_s = y|X)$ via marginalization, and x_s will be selected by the feature function. The gradient with respect to T has a similar argument. But instead what will be important is the feature $[\![y_s = i, y_{s+1} = j]\!]$ which will marginalize $p(\mathbf{y}|X)$ to $p(y_s = i, y_{s+1} = j)$.

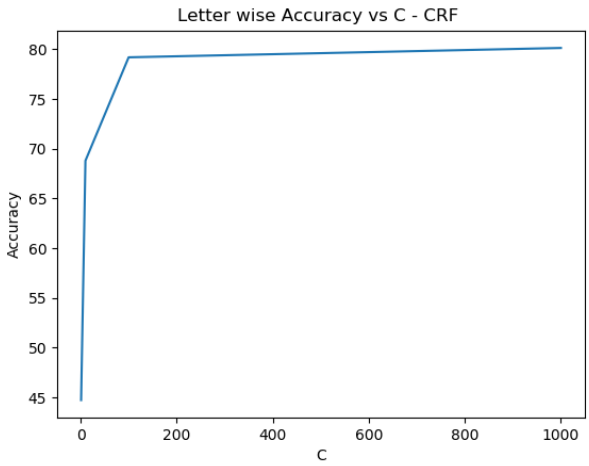
Maximum Objective Value for 1C	200.33257896998455
Average Log Probability for 2A	-31.28
Optimal Objective Value for 2B	3701.15

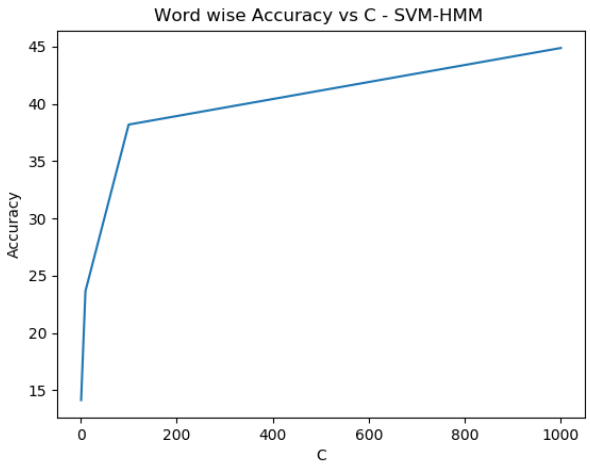
Observation for question 3

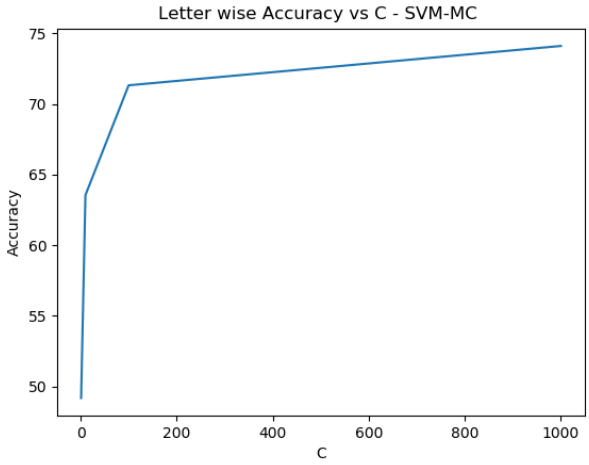
In general , as the value of C increase the letter wise accuracy and word wise accuracy increase for the given models : SVM-MC , SVM-HMM , CRF . We observed that the word wise accuracy was substantially lower than letter wise accuracy for all the given models . This behavior is as expected as word wise classification is a harder task as compared to letter wise classification .Both CRF and SVM-HMM perform better as compared to SVM-MC both word wise and letter wise . The performance of CRF is generally better then both the SVM models . The SVM-struct performs slightly better than the CRF model .

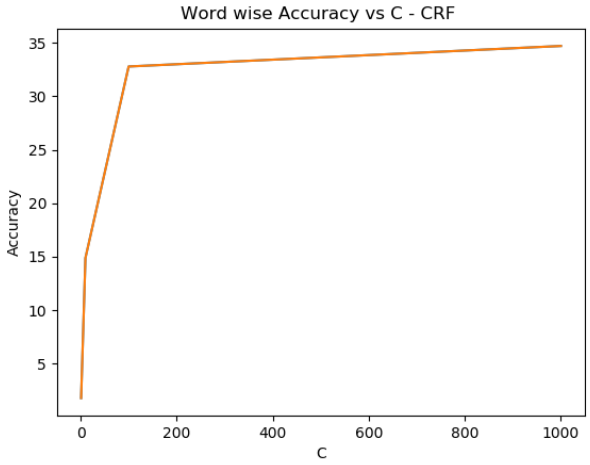
Observation for question 4

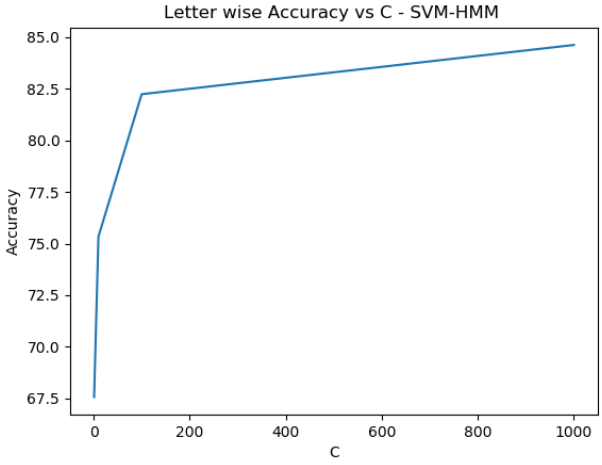
In general , as the number of transformation increase , the letter wise / word wise accuracy decrease for the given models : SVM-MC , SVM-HMM , CRF , This is as expected because once we tamper the data with increasing number of transformations , the performance of the model should decrease . CRF performs better than SVM-MC as it make use of the adjacent letter to do prediction . So , the effect of tampering is not seen as much for CRF , but the same is not true for SVM-MC as it performance gets worse with increase in number of transformations .

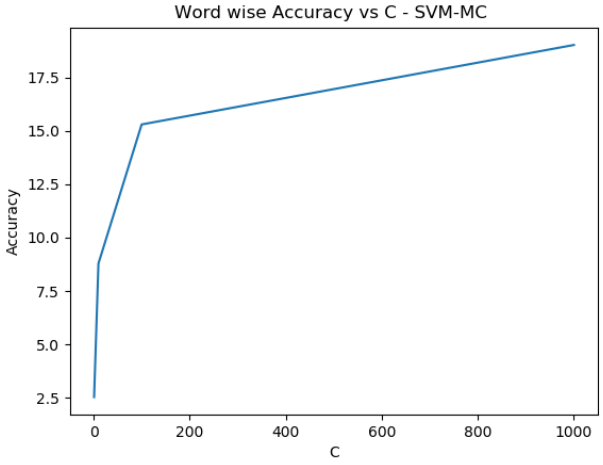

Letter wise Accuracy vs C - CRF
Accuracy
C

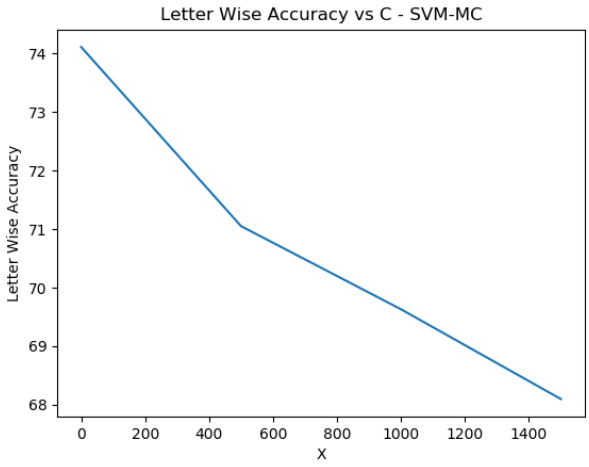

Word wise Accuracy vs C - SVM-HMM
Accuracy
C

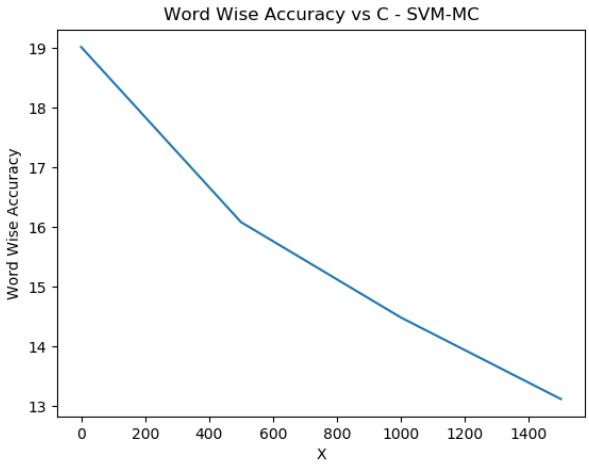

Letter wise Accuracy vs C - SVM-MC
Accuracy
C


Word wise Accuracy vs C - CRF
Accuracy
C


Letter wise Accuracy vs C - SVM-HMM
Accuracy
C


Word wise Accuracy vs C - SVM-MC
Accuracy
C


Letter Wise Accuracy vs C - SVM-MC
Letter Wise Accuracy
X
74
73
72
71
70
69
68
0
200
400
600
800
1000
1200
1400


Word Wise Accuracy vs C - SVM-MC
Word Wise Accuracy
X
19
18
17
16
15
14
13
0
200
400
600
800
1000
1200
1400