CS688: Graphical Models - Spring 2015

Assignment 2: Part A

Trung Kien Tran 28957765

The data was load

Question 1. (*20 points*) **Exhaustive Inference:** In this question, you will implement simple exhaustive inference for the CRF model. The code packages provides a pre-trained model for the OCR task including the feature parameters (*feature-params.txt*) and the label-label transition parameters (*transition-params.txt*). Use these parameters to answer the following questions. For grading purposes, make sure to list results table rows and/or columns using the character ordering "etainoshrd".

1.1 (*2 points*): *For the first test word only, compute the node potentials $\phi'(y_{ij})$ obtained by conditioning the CRF on the observed image sequence. After conditioning, there is one node potential per position in the test word. Each node potential is a vector with one entry per character label. Report the node potential as a table for each position in the test word.*

The code for this part are $ques11.m$ and $nodePotential.m$
The node potential values of each position in the first test word "that"

	x_{10}	x_{11}	x_{12}	x_{13}
e	-7.6444	-4.0745	-10.2081	6.4649
t	18.4684	5.7448	0.8973	24.5313
a	-6.3286	1.1764	17.1910	-13.3429
i	10.4225	-1.7931	-12.0177	5.8712
n	-4.9672	1.2122	5.5794	-10.9548
o	-1.9340	-1.7849	-0.5940	-11.4965
s	-0.9452	-8.2999	-21.4264	-5.4946
h	-5.6571	3.0952	9.1489	-7.1956
r	5.3953	6.8066	9.4824	8.0457
d	-6.9098	-0.3416	1.9472	3.5715

1.2 (*2 points*): *For the first three test words, compute the value of the negative energy of the true label sequence after conditioning on the corresponding observed image sequence:*

$$-E_W(\mathbf{x}_i, \mathbf{y}_i) = \sum_{j=1}^{L_i} \phi'(y_{ij}) + \sum_{j=1}^{L_i-1} \sum_{c=1}^{C} \sum_{c'=1}^{C} W^T_{cc'}[y_{ij} = c][y_{ij+1} = c'] = \sum_{j=1}^{L_i} \phi'(y_{ij}) + \sum_{j=1}^{L_i-1} W^T_{y_{ij},y_{ij+1}}$$

.

The code for this part are $ques12.m$, $negEnergy.m$ and $chars2id.m$
The negative energy of the true label sequence of the first three test words
First word: "that" 63.9793
Second word: "hire" 89.6109
Third word: "rises" 96.9406

1.3 (*6 points*): *For the first three test words, use exhaustive summation over all possible character label sequences to compute the value of the log partition function for the CRF model after conditioning on the corresponding observed image sequence. Report the value you compute.*

The code for this part is $ques13.m$, $negEnergy.m$ and $logSumExp.m$
The value of the log partition function for the CRF model after conditioning of the first three test words
First word: “that” 67.6019
Second word: “hire” 89.6144
Third word: “rises” 103.5276

1.4 (*6 points*): *For the first three test words, compute the most likely joint labeling (character sequence) word. Report both the labeling and its probability under the model.*

The code for this part is $ques14.m$, $negEnergy.m$ and $id2chars.m$
First predicted word: “trat” with probability 0.7958
Second predicted word: “hire” with probability 0.9965
Third predicted word: “riser” with probability 0.9370

1.5 (*4 points*): *For the first test word only, compute the marginal probability distribution over character labels for each position in the word. Report each marginal distribution using a table.*

The code for this part is $ques15.m$ and $nodePotential.m$
Marginal distribution over characters for each position in the first word:

	x_{10}	x_{11}	x_{12}	x_{13}
e	7.4548e-07	2.6473e-05	5.7411e-08	1.0000
t	0.0023	6.9207e-09	5.4313e-11	0.9977
a	6.1035e-11	1.1090e-07	1.0000	5.4864e-14
i	0.9896	4.9008e-06	1.7774e-10	0.0104
n	2.6255e-05	0.0011	0.9989	6.5885e-08
o	0.1672	0.1941	0.6386	1.175e-05
s	0.9889	6.3249e-04	1.2598e-09	0.0105
h	3.7052e-07	0.0023	0.9977	7.9550e-08
r	0.0127	0.0520	0.7557	0.1796
d	2.5073e-05	0.0320	0.1593	0.8086

Question 2. (*40 points*) **Sum-Product Message Passing:** In this question, you will implement the sum-product inference algorithm for the CRF model. The code packages provides a pre-trained model for the OCR task including the feature parameters (*feature-params.txt*) and the label-label transition parameters (*transition-params.txt*). Use these parameters to answer the following questions.

2.1 (*6 points*): *For the first test word only, condition on the observed image sequence to obtain a chain-structured Markov network. Next, convert the Markov network into a clique tree with cliques* $C_1 = \{Y_1, Y_2\}, C_2 = \{Y_2, Y_3\}, C_3 = \{Y_3, Y_4\}$ *and edges* $C_1 - C_2$ *and* $C_2 - C_3$*. Compute the clique potentials* $\psi_1(Y_1, Y_2)$, $\psi_2(Y_2, Y_3)$ *and* $\psi_3(Y_3, Y_4)$*. Include the node potential* $\phi'(Y_1)$ *in* C_1, $\phi'(Y_2)$ *in* C_2*, for* $\phi'(Y_3)$ *in* C_3 *and* $\phi'(Y_4)$ *in* C_3*. Each clique potential is a* 10×10 *table. Report the* 3×3 *block of entries between the labels “t,a,h” for each of the three clique potentials.*

The code for this part is $ques21.m$, $nodePotential.m$, and $scatterMatrix.m$
Clique potentials for the labels ‘t’, ‘a’, and ‘h’ for each of the three clique potentials:

	t	a	a
t	17.8146	18.7491	18.8389
a	-6.0479	-6.5593	-6.2812
h	-5.2916	-5.6098	-5.7933

	t	a	a
t	5.0911	-6.0255	6.1103
a	1.4571	0.9457	1.2237
h	3.4607	3.1425	2.9590

	t	a	a
t	24.7749	-12.1649	-5.9328
a	42.0030	3.6174	10.0427
h	34.0456	-4.1467	1.8171

2.2 (*8 points*): *For the first test word only, use the clique tree potentials to compute the log-space sum-product messages $\delta'_{1\to2}(Y_2)$, $\delta'_{2\to1}(Y_2)$, $\delta'_{2\to3}(Y_3)$, $\delta'_{3\to2}(Y_3)$ from the clique tree potentials. Report the value of each message in a table.*

The code for this part is $ques22.m$, $nodePotential.m$, and $messages.m$
Forward message $0 \to 1$:

18.5893	17.8153	18.7494	18.5227	18.1808	18.6773	18.0913	18.8341	18.3634	18.2164

Forward message $1 \to 2$:

25.6511	25.2369	25.5984	25.5779	25.2716	25.6012	25.0715	25.3880	25.4145	25.2026

Backward message $2 \to 1$:

37.7353	48.0291	42.9495	40.4300	40.9076	40.0510	33.4551	45.1460	49.0110	42.4119

Backward message $1 \to 0$:

14.4439	24.7749	42.0030	12.5677	29.8224	24.1459	2.7272	34.0456	33.9083	26.2260

2.3 (*8 points*): *For the first test word only, use the messages and the clique tree potentials to compute the log beliefs at each node in the clique tree $\beta'(Y_1, Y_2)$, $\beta'(Y_2, Y_3)$ and $\beta'(Y_3, Y_4)$. Report the 2×2 block of log belief entries between the first two labels “t” and “a” only for each of the three cliques.*

The code for this part is $ques23.m$, $nodePotential.m$, $messages.m$, $scatterMatrix.m$, and $calBeliefs.m$
Log belief entries between ‘t’ and ‘a’ for each of the three cliques:

	t	a
t	65.8437	61.6986
a	41.9812	36.3903

	t	a
t	47.6812	65.8438
a	44.9813	61.6980

	t	a
t	50.0117	13.0720
a	67.6013	29.2158

2.4 (*8 points*): *For the first test word only, use the computed log beliefs to compute the marginal probability distribution over each position in the word. Report the marginal distributions as tables. Also use the beliefs to compute the pairwise marginals* $P(y_{i1}, y_{i2}|\mathbf{x}_i)$. *Report the* 3×3 *block of entries between the labels "t,a,h".*

The code for this part is $ques24.m$, $nodePotential.m$, $messages.m$, $scatterMatrix.m$, $calBeliefs.m$ and $calMarginals.m$

Marginal distributions for first test word:

	x_{10}	x_{11}	x_{12}	x_{13}
e	7.2227e-12	1.2658e-05	1.1321e-12	8.8683e-09
t	0.9995	0.1725	2.2945e-08	1.0000
a	2.6262e-11	0.0027	0.9995	2.1357e-07
i	4.7272e-04	1.7528e-04	1.6119e-13	7.4054e-09
n	7.1555e-11	2.0074e-04	3.6976e-06	3.2900e-16
o	2.1138e-09	1.4005e-04	1.7611e-08	1.4410e-16
s	3.2960e-19	1.0646e-07	5.1721e-18	5.3711e-14
h	4.3493e-11	0.0267	2.8353e-04	1.3178e-14
r	2.6281e-06	0.7966	2.5376e-04	6.3940e-08
d	1.0694e-11	9.3629e-04	9.4638e-08	6.3736e-10

Pairwise marginals for the labels 't', 'a', 'h':

	t	a	h
t	0.1724	0.0027	0.0267
a	7.4658e-12	2.7860e-14	3.3086e-13
h	1.5904e-11	7.2001e-14	5.3897e-13

	t	a	h
t	2.2314e-09	0.1723	6.5686e-05
a	1.4997e-10	0.0027	1.2616e-06
h	1.2104e-09	0.0267	7.7862e-06

	t	a	h
t	2.2945e-08	2.0796e-24	1.0581e-21
a	0.9995	2.1337e-17	1.3171e-14
h	2.8353e-04	7.3432e-21	2.8571e-18

2.5 (*10 points*): *Generalize the steps given above to compute the single variable and pairwise marginal probabilities for any sequence of input images. Apply your completed algorithm to compute the marginal probability distribution over the character labels given the image sequences for each word in the test set. For each position in each test word, predict the character with maximum probability. List your predictions for the first five test sequences. In addition, use the true character labels in test_words.txt to compute the average character-level accuracy over the complete test set. Report the accuracy that you find to three decimal places.*

The code for this part is $ques25.m$, $nodePotential.m$, $messages.m$, $scatterMatrix.m$, $calBeliefs.m$ and $calMarginals.m$
Predictions for the first five words: trat hire riser edison shore
True first five word: that hire rises edison shore
Overall character accuracy: 0.9167

Question 3. (*34 points*) **Maximum Likelihood Learning Derivation:** In this problem, you will derive the maximum likelihood learning algorithm for conditional random field models.

3.1 (*8 points*): *Write down the average log likelihood function for the CRF given a data set consisting of N image sequences $\mathbf{x}_i$ and label sequences $\mathbf{y}_i$.*

$$\mathcal{L}(W \mid x_{1:N}, y_{1:N}) = \frac{1}{N}\sum_{n=1}^{N} \log P_W(y_n, x_n)$$

Where the probability function

$$P_W(y_n, x_n) = \frac{-E_W(x_n, y_n)}{Z}$$

that the negative energy function:

$$-E_W(x_n, y_n) = \exp\left[\sum_{j=1}^{L_i}\sum_{c=1}^{C}\sum_{f=1}^{F} W^F_{cf}[y_{nj} = c]x_{njf} + \sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{C} W^T_{cc'}[y_{nj} = c][y_{nj+1} = c']\right]$$

and the partition function:

$$Z = \sum_{x'_i}\sum_{y'_i}\exp\left[\sum_{j=1}^{L_i}\sum_{c=1}^{C}\sum_{f=1}^{F} W^F_{cf}[y'_{ij} = c]x'_{ijf} + \sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{C} W^T_{cc'}[y'_{ij} = c][y'_{ij+1} = c']\right]$$

3.2 (*8 points*): *Derive the derivative of the average log likelihood function with respect to the feature parameter W^F_{cf}. Show your work.*

We first apply the chain rule

$$\frac{\partial \mathcal{L}}{\partial W^F_{cf}} = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{P_W(y_n, x_n)}\frac{\partial P_W}{\partial W^F_{cf}} P_W(y_n, x_n)\right]$$

Then get partial derivative of P_W with respect to W^F_{cf}

$$\begin{aligned}\frac{\partial P_W}{\partial W^F_{cf}} &= \frac{\exp(-E_W(y_n, x_n))}{\sum_{y'}\sum_{x'}\exp(-E_W(y_n, x_n))}\frac{-\partial E_W(y_n, x_n)}{\partial W^F_{cf}} \\ &= \frac{\exp(-E_W(y_n, x_n))}{\left[\sum_{y'}\sum_{x'}\exp(-E_W(y_n, x_n))\right]^2}\sum_{y'}\sum_{x'}\exp(-E_W(y', x'))\frac{-\partial E_W(y', x')}{\partial W_c f^F} \\ &= P_W(y_n, x_n)\frac{-\partial E_W(y_n, x_n)}{\partial W^F_{cf}} - P_W(y_n, x_n)\sum_{y'}\sum_{x'} P_W(y', x')\frac{-\partial E_W(y', x')}{\partial W^F_{cf}}\end{aligned}$$

The derivative of the energy function E_W with respect to W^F_{cf}:

$$\frac{\partial E_W(y_n, x_n)}{\partial W^F_{cf}} = \sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{f=1}^{F}[y_{nj} = c][x_{njf}]$$

Then

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W^F_{cf}} &= \frac{1}{N}\sum_{n=1}^{N}\Big[\frac{1}{P_W(y_n, x_n)}\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{f=1}^{F}[y_{nj} = c][x_{njf}] - P_W(y_n, x_n)\sum_{y'}\sum_{x'} P_W(y', x')\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{f=1}^{F}[y'_j = c][x'_{jf}]\Big] \\ &= \frac{1}{N}\sum_{n=1}^{N}\left[\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{F}[y_{nj} = c][x_{njf}] - \sum_{y'}\sum_{x'} P_W(y', x')\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{f=1}^{F}[y'_j = c][x'_{jf}]\right]\end{aligned}$$

Since n is not used in the second term, then it could be simplified as

$$\frac{\partial \mathcal{L}}{\partial W^F_{cf}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{F}[y_{nj} = c][x_{njf}] - \sum_{y'}\sum_{x'} P_W(y', x')\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{f=1}^{F}[y'_j = c][x'_{jf}]$$

3.3 (*8 points*): *Derive the derivative of the average log likelihood function with respect to the transition parameter $W^T_{cc'}$. Show your work.*

For transition parameter $W^T_{cc'}$, we have similar with W^F_{cf}
For the derivative of $E_W(y_n, x_n)$

$$\frac{\partial E_W(y_n, x_n)}{\partial W^T_{cc'}} = \sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{C}[y_{nj} = c][y_{nj+1} = c']$$

After substituting, we get

$$\frac{\partial \mathcal{L}}{\partial W^F_{cf}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{C}[y_{nj} = c][y_{nj+1} = c'] - \sum_{y'}\sum_{x'} P_W(y', x')\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{C}[y'_j = c][y'_{j+1} = c']$$

3.4 (*8 points*): *Explain how, as a byproduct of the sum-product algorithm's computation of the single-variable and pairwise marginal probabilities, you can efficiently compute both the value of the log-likelihood function and the values of the above derivatives.*

To computed the sum-product algorithm's computation of the single-variable probability, we could compute the value of the log-likelihood function.
To compute pairwise marginal probabilities, we could sum over the log-marginals for each.
To compute the derivatives, could use the marginals that we need at each iteration x_i, y_i, such as below

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W^F_{cf}} &= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{F}[y_{nj}=c][x_{njf}] - \sum_{y'}\sum_{x'}P_W(y',x')\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{f=1}^{F}[y'_j=c][x'_{jf}] \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{c'=1}^{F}[y_{nj}=c][x_{njf}] - \sum_{j=1}^{L_i-1}\sum_{c=1}^{C}\sum_{f=1}^{F}\sum_{y'_j}\sum_{x'_{jf}}P_W(y'_j,x'_{jf})[y'_j=c][x'_{jf}]\end{aligned}$$

3.5 (*2 points*): *Using a data set consisting of the first 50 training data cases only, compute the average log likelihood of the true label sequences given the image sequences using the supplied model parameters.*

The code for this part is $ques35.m$, $nodePotential.m$, $messages.m$, $scatterMatrix.m$, $calBeliefs.m$ and $calMarginals.m$
Average log likelihood of first 50 training examples: -4.5652

3.6 (*0 points*): *Using a data set consisting of the first 50 training data cases only, compute the derivative with respect to each model parameter of the average log likelihood of the true label sequences given the image sequences using the supplied model parameters. There is nothing to hand in for this question, but we will provide the solution to help you debug your code. These files are in the model folder and called feature-gradient.txt and transition-gradient.txt.*

Question 4. (*6 points*) **Numerical Optimization Warm-Up:** In part B of the assignment, you will implement the above learning algorithm using a numerical optimizer to maximize the log likelihood. In this question, you will experiment with optimizing a basic function.

4.1 (*3 points*): *Consider the objective function $f_w(x,y) = -(1-x)^2 - 100(y-x^2)^2$. Derive the derivatives of $f(x,y)$ with respect to x and y (the gradient function). Show your work.*

$$\begin{aligned}\frac{\partial f}{\partial x} &= -(2(1-x)(-1)) - 100(2(y-x^2)(y-2x)) \\ &= 2(1-x) - 200(y-x^2)(-2x) \\ &= -400x^3 + 400xy - 2x + 2\end{aligned}$$

$$\frac{\partial f}{\partial y} = -200(y-x^2)$$

4.2 (*3 points*): *Select a numerical optimizer for the programming language you are using. If you haven't used it previously, study its documentation carefully. Implement the objective function and the gradient function in the form required by your numerical optimizer. Write code to use the optimizer to* ***maximize*** *$f(x,y)$. Report both the location of the maximum and the value of the objective function at the maximum.*