# CSL 864: Assignment 2 (Part A)

Due: Friday October 16, 12 noon

**Max Points: 46 (+ 10 extra credit)**

**Notes:**

- You should submit all your code as well as any graphs that you might plot (see below).
- Include a **single write-up (pdf) file** which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- You can use any programming language from the set C++, Java, Python, Matlab. If you would like to use any other language, please check with us before you start.
- Your code should have appropriate documentation for readability.
- You will be graded based on what you submit as well as your ability to explain your code.
- Refer to the course website for assignment submission instructions.
- This assignment is supposed to be done individually. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. Any cheating will result in a penalty of **-10** points on your total course score (in addition to a 0 on the assignment). Stricter penalties (**including a fail grade**) may follow.
- Note: The problem below has been borrowed (with minor changes) from the Graphical Models course offered by Andrew McCallum at UMass Amherst.

**Exact and Approximate Inference for Graphical Models:**
In this problem, we will build upon the OCR problem from Assignment 1. In particular, we will be experimenting with the following two inference algorithms: clique tree message passing and loopy BP for the OCR problem. Consider the OCR model as described earlier in Assignment 1. We will add another kind of factors to the model, called the *Pair-Skip* factors. These factors capture the similarity between the pair of words which have the same image id(s) occurring in them. We can imagine that for images that are similar across two different words, we would like the predicted characters to be the same. Consider the undirected graphical model shown in Figure 1 below, consisting of two observed image sequences. Note that there is an image id (728) that appears in both the sequences. To capture the similarity as described above, we add the pair-skip factor between these variables. Note that the inference would now have to be performed jointly over a pair of words. Given a pair of words consisting of words $w1$ and $w2$, the complete model score over them will consist of the following four classes of factors:

- **OCR Factors, $\psi_o$**: These factors capture the predictions of a character-based OCR system, and hence exist between every image variable and its corresponding character variable. The number of these factors for pair is $len(w_1) + len(w_2)$. The value of factor between an image variable and the character variable at position $i$ is dependent on $img(i)$ and $char(i)$, and is stored in ocr.dat file described in the Assignment 1.
- **Transition Factors, $\psi_t$**: Since we also want to represent the co-occurrence frequencies of the characters in our model, we add these factors between all consecutive character variables of a word. The number of these factors for pair is $(len(w_1) - 1) + (len(w_{12}) - 1)$. The value of factor between two character variables at positions $i$ and $i + 1$ is dependent on $char(i)$ and $char(i + 1)$, and is high if $char(i + 1)$ is frequently preceded by $char(i)$ in English words. These values are given to you in trans.dat file as described in the Assignment 1.

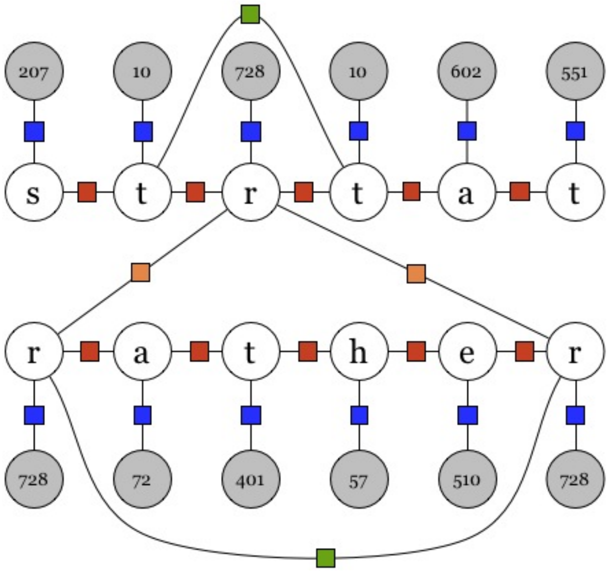

Figure 1: Model with Pair-Skip Factors. Source: Graphical Models Course (Spring 2011) by Andrew McCallum, UMass Amherst.

- **Skip Factors,** $\psi_s$: Another intuition that we would like to capture in our model is that similar images in a word always represent the same character. Thus our model score should be higher if it predicts the same characters for similar images. These factors exist between every pair of image variables that have the same id, i.e. this factor exist between all $i$,$j$, $i \neq j$ such that $img(i) == img(j)$. The value of this factor depends on $char(i)$ and $char(j)$, and is **5.0** if $char(i) == char(j)$, and **1.0** otherwise.
- **Pair-Skip Factors,** $\psi_D$: These factors exist between pairs of image variables that belong to different words and have the same id. The value of this factor depends on the characters, i.e. its value is **5.0** if the characters are the same, and **1.0** otherwise.

You can download all the data from here.

- **ocr.dat,trans.dat:** Same as Assignment 1.
- **data-tree.dat (and truth-tree.dat):** These files contain all the tree-structured instances of the OCR task. The observed dataset (**data-tree.dat**) consists observed images of one word on each row (same format as before), with a empty line between pairs. The true words for this pair of observed images are stored respectively as rows in **truth-tree.dat**(with the empty lines). As in Assignment 1, you should iterate through both the files together to ensure you have the true words for each pair, along with their observed images. See Figure 2 for illustration.

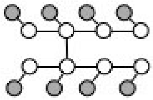

Figure 2: Tree Structure (across words). Source: Graphical Models Course (Spring 2011) by Andrew McCallum, UMass Amherst.

- **data-treeWS.dat (and truth-treeWS.dat):** These files contain all the tree-structured instances of the OCR task along with skip factors. The format is same as the data-tree.data and truth-tree.data. See Figure 3 for illustration.
- **data-loopy.dat (and truth-loopy.dat):** These files contain all the loopy instances of the OCR task. The format is same as the data-tree.data and truth-tree.data. See Figure 4 for illustration.
- **data-loopyWS.dat (and truth-loopyWS.dat):** These files contain all the loopy instances of the OCR task along with skip factors. The format is same as the data-tree.data and truth-tree.data. See Figure 5 for illustration.

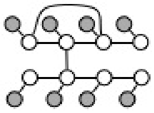

Figure 3: Tree Structure (across words) with Skip Factors. Source: Graphical Models Course (Spring 2011) by Andrew McCallum, UMass Amherst.

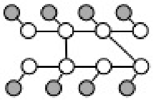

Figure 4: Loopy Structure (across words). Source: Graphical Models Course (Spring 2011) by Andrew McCallum, UMass Amherst.

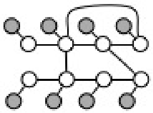

Figure 5: TLoopy Structure (across words) with Skip Factors. Source: Graphical Models Course (Spring 2011) by Andrew McCallum, UMass Amherst.

**Core Tasks**

* **Graphical Model:** Implement the graphical model containing the factors above. For any given assignment to the character variables, your should be able to calculate the model score (see below). Your implementation should allow switching between the following four factor models:

1. OCR model (OCR) : only contains the OCR factors
2. OCR + Transition model (OCR+T): contains OCR and Transition factors
3. OCR + Transition + Skip model (OCR+T+S): contains OCR, Transition and skip factors
4. OCR + Transition + Skip + Pair-Skip model (OCR+T+S+PS): Containing all four types of factors

1. **Inference (30 points):** Implement and evaluate the following inference algorithms:

   - **Message Passing Over a Junction Tree (MP) (15 points):** Convert the given graph into a clique tree corresponding to the varibale elimination ordering based on min-fill heuristic (refer to Chapter 9, K&F for details). Implement the two way sum-product message passing over this junction tree. Implement the version where you designate one (any) of the nodes in the tree as the root, send all the messages up towards the root, pass the messages downwards from the root (after receiving all the upward messages) and finally update the beliefs at each node (as done in the Minor 2 question).
   - **Loopy Belief Propagation (LBP) (15 points):** Implement the loopy version of the belief update message passing. Decide an appropriate stopping criteria for your algorithm (does it always converge?). Use the synchronous version of the algorithm. Note that one round of message passing involves updating all the messages (and beliefs) in the network once based on messages in the last time step.

* **Model Accuracy:** In this assignment, instead of calculating the score over the MAP assignment, we will instead use the score over the state correponding to the highest marginal for each node. In particular, we will calculate the marginal probability for each of the character variables and treat as the model prediction the value with the highest marginal probability for that character variable. For each of the tasks below you will report:

(a) Character-wise accuracy (Ch-Acc): Ratio of correctly predicted characters (according to the highest marginal) to total number of characters

(b) Word-wise accuracy (Wd-Acc): Ratio of correctly predicted words (according to highest marginal) to total number of words

(c) Data Log-likelihood (LL): For each word in the dataset, calculate the log of the marginal probability of the word as the product of the marginal probabilities of its character variables. Report the average of this value over all the words in the dataset.

(d) Running time (Time): Report the time taken for acutal inference (exclude the time taken for constructing the required network structures e.g. the clique tree in message passing).

2. **Experiments (16 points):**

- **Comparing Inference Algorithms (8 points):** Run each of MP (message passing), LBP (loopy belief propagation) for each of the four datasets and report Ch-Acc, Wd-Acc, LL and Time in the following tabular format (one table for each dataset). Comment on your results.

| **Algorithm** | Ch-Acc | Wd-Acc | LL | Time (seconds) |
|---|---|---|---|---|
| MP | | | | |
| LBP | | | | |

- **Adding Sophistication in Factors (8 points):** Run the MP (message passing) over clique tree for the loopy-WS dataset for each of the four set of factor models described above. Report the values for each of the metrics, Ch-Acc, Wd-Acc, LL and Time in the following tabular format. Comment on your results.

| **Model** | Ch-Acc | Wd-Acc | LL | Time (seconds) |
|---|---|---|---|---|
| OCR | | | | |
| OCR+T | | | | |
| OCR+T+S | | | | |
| OCR+T+S+PS | | | | |

**Extra Credit** (**10 points**):
**Maximum-A-Posteriori (MAP) Inference:** Implement the Max-Product (MAP inference) variant of the MP and LBP inference algorithms. Repeat the two sets of experiments described above. How do your results compare with the highest marginal case. Comment.