INFERENCE AND REPRESENTATION: HOMEWORK 6

STUDENT: MARGARITA BOYARSKAYA

Problem 1. *Structured SVM for POS tagging.*

a) In implementing the solution, I defined the range of C values to be $C \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 510, 50, 100, 500\}$. Training the SSVM model on the first 4500 sentences and grid-searching across the values of C, I discovered that the value minimizing the Hamming loss is $C = 0.1$, corresponding to a 0.1207 validation error. When re-training on the full dataset, a 0.119 Hamming loss was obtained.

The code that solves the problem is provided in the file titled *a.py*, attached. The attached file titled *1a Terminal Saved Output* contains the results of running the script. In particular, the code prints the following results:

Best C 0.1 , training error: 0.116533949824 , validation error: 0.120678322598
test error: 0.119232286052

b) In this part of the problem, I vary the size of the training set to be the first $100, 200, 500$, and 1000 sentences. The code is presented in *b.py*, and the results of its execution are attached in file *1b Terminal Saved Output*.

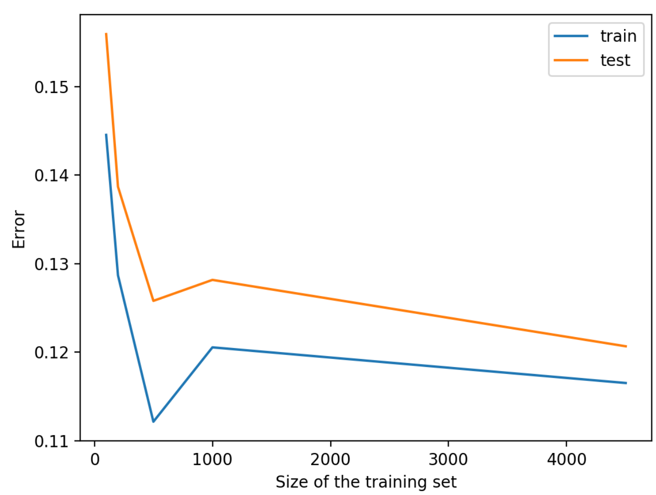
The results of error computation for various training size parameters are:

b = 100 C = 0.0001 : 0.741501840911
b = 100 C = 0.0005 : 0.740816850758
b = 100 C = 0.001 : 0.740131860605
b = 100 C = 0.005 : 0.761709050432
b = 100 C = 0.01 : 0.773782001884
b = 100 C = 0.05 : 0.813254559466
b = 100 C = 0.1 : 0.828495590376
b = 100 C = 0.5 : 0.855467077661
b = 100 C = 1 : 0.853069612124
b = 100 C = 5 : 0.828923709222
b = 100 C = 10 : 0.819847589691
b = 100 C = 50 : 0.829180580529
b = 100 C = 100 : 0.828752461683
b = 100 C = 500 : 0.829351828067
b = 200 C = 0.0001 : 0.740388731912
b = 200 C = 0.0005 : 0.738419385221
b = 200 C = 0.001 : 0.740046236835

b = 200 C = 0.005 : 0.778833804264
b = 200 C = 0.01 : 0.788937409025
b = 200 C = 0.05 : 0.827468105146
b = 200 C = 0.1 : 0.843479749979
b = 200 C = 0.5 : 0.869509375803
b = 200 C = 1 : 0.871307474955
b = 200 C = 5 : 0.82301566915
b = 200 C = 10 : 0.848103433513
b = 200 C = 50 : 0.816508262694
b = 200 C = 100 : 0.814538916003
b = 200 C = 500 : 0.835516739447
b = 500 C = 0.0001 : 0.735165681993
b = 500 C = 0.0005 : 0.743128692525
b = 500 C = 0.001 : 0.769928932272
b = 500 C = 0.005 : 0.794074835174
b = 500 C = 0.01 : 0.819333847076
b = 500 C = 0.05 : 0.859491394811
b = 500 C = 0.1 : 0.872506207723
b = 500 C = 0.5 : 0.887832862403
b = 500 C = 1 : 0.877558010104
b = 500 C = 5 : 0.839969175443
b = 500 C = 10 : 0.861717612809
b = 500 C = 50 : 0.801438479322
b = 500 C = 100 : 0.830379313297
b = 500 C = 500 : 0.856751434198

Evidently, the first experiment attains minimal loss at $C = 0.5$, the second – at $C = 1$, third – at $C = 0.5$, and the full training set, as established earlier, points at $C = 0.1$ as a minimizing value.

Below is the plot showing both the test and train error as a function of the amount of training data:



c) The solution code is included in *c_and_d.py*.

For a particular random triplet documented in the code output printout (*1c Terminal Saved Output*), the choice fell on pronouns, punctuation, and and noun classes. The resulting transition weights were as follows:

Chosen classes: ['pronoun' 'punctuation' 'noun']
pronoun - punctuation -0.00338585625322
punctuation - pronoun -0.0234312092397
pronoun - noun 1.14404578091
noun - pronoun -0.496991564155
punctuation - noun 0.655646657842
noun - punctuation 0.562797652016

Analyzing these results, one can see that they readily submit to an intuitive interpretation. For example, the "noun - pronoun" value of -0.53 make sense considering that pronouns rarely come after nouns in English. The inverse relationship, however, is one that warrants a highly positive score, as it corresponds to a typical ordering of words in English.

The "noun - punctuation" score of 0.60 is in line with the fact that a noun is a plausible terminal word in a sentence, although not the only possible one. The opposite relationship is realized whenever a sentence ends with punctuation and a new one begins with a noun – again, results make sense.

Punctuations are highly unlikely to precede a punctuation mark, the only example immediately coming to mind being the "- said she" type of phrases typical of creative writing, which is not the domain from which the data hails. The inverse, "punctuation-pronoun" appears to be a very common ordering in every day speech and in colloquial writing. I believe that the low score is, again, due to a highly formal style of the sentences in the dataset, as beginning a sentence with a pronoun other than "it" is considered poor writing style in an academic setting.

Let us examine most relevant features for each class:

Most relevant features for tag pronoun :
Prefix: it, Suffix: ir, Suffix: ey, Prefix: he, Suffix: em, Suffix: we, Prefix: yo, Prefix: hi, Suffix: my, Prefix: wh
Most relevant features for tag punctuation :
'Prefix:)', 'Suffix:)', 'Prefix: ;', 'Suffix: ;', 'Prefix: (', 'Suffix: (', 'Prefix: "', 'Suffix: "', 'Suffix: $', 'Prefix: $'
Most relevant features for tag noun :
Suffix: rs', 'Prefix: on', 'Initial Capital', 'Suffix: es', 'Suffix: gs', 'Suffix: ts', 'Suffix: cy', 'Suffix: ns', 'Suffix: ls', 'Prefix: it'

It is worth noticing that all punctation marks are counted twice: as both a prefix and a suffix, essentially covering only the top five most common punctuation marks in the dataset.

d) The solution code is included in *c_and_d.py*.

Below is a sample of the output (provided in full in *1d Terminal Saved Output*) for one selected sentence:

Sentence # 288 :

Witnesses — noun ; Predicted: noun
said — verb ; Predicted: verb
most — adjective ; Predicted: adjective
shops — noun ; Predicted: noun
were — verb ; Predicted: verb
closed — adjective ; Predicted: verb
in — preposition ; Predicted: preposition
towns — noun ; Predicted: noun
and — other ; Predicted: other
villages — noun ; Predicted: noun
in — preposition ; Predicted: preposition
the — determiner ; Predicted: determiner
areas — noun ; Predicted: noun
; — punctuation ; Predicted: punctuation
with — preposition ; Predicted: preposition
the — determiner ; Predicted: determiner
exception — noun ; Predicted: noun
of — preposition ; Predicted: preposition
Hebron — noun ; Predicted: noun
; — punctuation ; Predicted: punctuation
a — determiner ; Predicted: determiner
West — noun ; Predicted: noun
Bank — noun ; Predicted: noun
city — noun ; Predicted: noun
still — adverb ; Predicted: noun
under — preposition ; Predicted: preposition
Israeli — adjective ; Predicted: adjective
occupation — noun ; Predicted: noun
. — punctuation ; Predicted: punctuation

Error rate for sentence 288 : 0.069

Problem 2. *Max-product belief propagation.*

For the given undirected graph $G = (V, E)$ with $V = \{1, 2, ..., 6\}$, distribution over $x_1, ..., x_6$ is expressed by factorization:

$$p_x(x) \propto \prod_{i \in V} \psi_i(x_i) \prod_{(i,j) \in E} \psi_{ij}(x_i, x_j). \tag{1}$$

I begin by using the MAP elimination algorithm with ordering $(6,5,4,3,2,1)$, thus rooting the tree at node x_1. Eliminating nodes $6,5,4$, and 3, I pass the messages, expressed as below:

$$\begin{aligned}
m_{62}(x_2) &= \max_{x_6} \psi_6(x_6)\psi(x_2,x_6),\\
m_{52}(x_2) &= \max_{x_5} \psi_5(x_5)\psi(x_2,x_5),\\
m_{41}(x_1) &= \max_{x_4} \psi_4(x_4)\psi(x_1,x_3),\\
m_{31}(x_1) &= \max_{x_3} \psi_3(x_3)\psi(x_1,x_4).
\end{aligned}$$

Eliminating node x_2 results in:

$$\begin{aligned}
m_{21}(x_1) = &\max_{x_2} \psi_2(x_2)\psi(x_1,x_2)m_{52}(x_2)m_{62}(x_2) =\\
&= \max_{x_2} \psi_2(x_2)\psi(x_1,x_2)[\max_{x_5} \psi_5(x_5)\psi(x_2,x_5)][\max_{x_6} \psi_6(x_6)\psi(x_2,x_6)].
\end{aligned}$$

Finally, I express the marginal:

$$\bar{p}_1(x_1) = \psi_1(x_1)m_{41}(x_1)m_{31}(x_1)m_{21}(x_1), \tag{2}$$

which obtains a maximum at:

$$x_1^* \in \arg\max_{x_1} \bar{p}_1(x_1) = \arg\max_{x_1} \psi_1(x_1)m_{41}(x_1)m_{31}(x_1)m_{21}(x_1).$$

Now, using the given potential values to calculate the maximals m_{ij} above, where $(i,j) \in \{(6,2),(5,2),(4,1),(3,1)\}$, one finds that all four of these obtain maximum value 4 at either one of two points: $x_i = A, x_j = B$, or $x_i = B, x_j = B$.

The maximal m_{21} is realized at point $x_2 = B, x_1 = B$ corresponding to value 64. The variables x_5 and x_6 are inconsequential in calculating the marginal and can take any value.

Finally, the most likely distribution:

$$x^* \in \arg\max_{x_1} \bar{p}_1 = \arg\max_{x_1} 1 * 64 * 4^2 = X,$$

where $X = \{(B,B,A/B,A/B,A/B,A/B)\}$ is a set of strings with only first two places fixed at value B, and the rest taking values A or B, resulting in 16 possibilities.

Problem 3. *Equality of model moments and empirical moments.*

We have $p(x,\theta) = \frac{1}{Z(\theta)} exp(\langle\theta, f(x)\rangle)$. The log-likelihood is:

$$\ell = \sum_{n=1}^{L} \log p(x_n,\theta) = \sum_{n=1}^{L} \Big(log \frac{1}{Z(\theta)} + \langle\theta, f(x)\rangle \Big) = \langle\theta, \sum_{n=1}^{L} f(x_n)\rangle - L\log Z(\theta).$$

Taking the differential, we get:

$$\nabla_{\theta_k}\ell = \sum_{n=1}^{L} f_k(x_n) - L\frac{\partial \log Z(\theta)}{\partial\theta_k}.$$

Let us examine the second expression:

$$\frac{\partial \log Z(\theta)}{\partial\theta_k} = \frac{1}{Z(\theta)} \sum_x \frac{\partial}{\partial\theta_k} exp\Big(\sum_{k'} \theta_{k'} f_{k'}(x)\Big) =$$

$$= \frac{1}{Z(\theta)} \sum_x exp\Big(\sum_{k'} \theta_{k'} f_{k'}(x)\Big) f_k(x) = \sum_x P(x|\theta) f_k(x).$$

Thus, the differential is:

$$\nabla_{\theta_k} \ell = \sum_{n=1}^{L} f_k(x_n) - L \sum_x P(x|\theta) f_k(x),$$

and at the maximum of the likelihood,

$$\sum_x P(x|\theta_{ML}) f_k(x) = \frac{1}{L} \sum_{n=1}^{L} f_k(x_n). \quad \square$$

Problem 4. *MaxEnt implies exponential family.*

Consider a distribution $p(x)$ over a finite set $\{x_1, \dots, x_N\}$. Re-writing it as vector $(p_1, \dots, p_N)$, our goal becomes that of maximizing:

$$-\sum_n p_n \log p_n$$

subject to constraints $\sum_n p_n = 1$ and $\sum_n p_n f_k(x_n) = a_k$.

Let us denote:

$$\Lambda = -\sum_n p_n \log p_n + \lambda_0 \Big(\sum_n p_n - 1\Big) + \sum_k \lambda_k \Big(\sum_n p_n f_k(x_n) - a_k\Big).$$

Now, introducing Lagrange multipliers, we derive the following constraints:

$$\frac{\partial \Lambda}{\partial p_n} = -\log p_n - 1 + \lambda_0 + \sum_k \lambda_k f_k(x_n) = 0 \tag{3}$$

$$\frac{\partial \Lambda}{\partial \lambda_0} = -\sum_n p_n - 1 = 0 \tag{4}$$

$$\frac{\partial \Lambda}{\partial \lambda_k} = -\sum_n p_n f_k(x_n) - a_k = 0. \tag{5}$$

From (3) we immediately see that:

$$p_n \propto exp\Big(\sum_k \lambda_k f_k(x_n)\Big) = exp\Big(\langle \lambda, f(x_n) \rangle\Big),$$

where $\lambda = (\lambda_1, \dots, \lambda_K)^T$. This proves that $p(x)$ belongs to an exponential family. $\square$

References

[1] http://people.kmi.open.ac.uk/stefan/www-pub/e.schofield-phd.pdf

[2] https://www.nst.ei.tum.de/fileadmin/w00bqs/www/publications/as/2012WS-HS-MaxSumFactorGraph.pdf