CS 475 Machine Learning (Spring 2017): Assignment 3

Due on April 7th, 2017 at 3:00PM

Raman Arora

Instructions: Please read these instructions carefully and follow them precisely. Feel free to ask the instructor if anything is unclear!

1. Please submit your solutions electronically via Gradescope.
2. Please submit a PDF file for the written component of your solution including derivations, explanations, etc. You can create this PDF in any way you want: typeset the solution in LATEX (recommended), type it in Word or a similar program and convert/export to PDF, or even hand write the solution (legibly!) and scan it to PDF. We recommend that you restrict your solutions to the space allocated for each problem; you may need to adjust the white space by tweaking the argument to `\vspace{xpt}` command. Please name this document <firstname-lastname>-sol3.pdf.
3. Submit the empirical component of the solution (Python code and the documentation of the experiments you are asked to run, including figures) in a Jupyter notebook file. In addition, we require that you save the Python notebook as a pdf file and append it to the rest of the solutions.
4. In addition, you will need to submit your predictions on a sentiment classification task to `Kaggle`, as described below, according to the competition rules.
5. **Late submissions:** You have a total of 72 late hours for the entire semester that you may use as you deem fit. After you have used up your quota, there will be a penalty of 50% of your grade on a late homework if submitted within 48 hours of the deadline and a penalty of 100% of your grade on the homework for submissions that are later than 48 hours past the deadline.
6. **What is the required level of detail?** When asked to derive something, please clearly state the assumptions, if any, and strive for balance: justify any non-obvious steps, but try to avoid superfluous explanations. When asked to plot something, please include the figure as well as the code used to plot it. If multiple entities appear on a plot, make sure that they are clearly distinguishable (by color or style of lines and markers). When asked to provide a brief explanation or description, try to make your answers concise, but do not omit anything you believe is important. When submitting code, please make sure it's reasonably documented, and describe succinctly in the written component of the solution what is done in each `py`-file.

Name: ______________________________

I. Support Vector Machines

In this problem, we will consider some details of the dual formulation of SVM, the one in which we optimize over the Lagrange multipliers α_i. In class we saw how to derive a constrained quadratic program, which can then be "fed" to an off-the-shelf quadratic program solver. These solvers are usually constructed to handle certain standard formulations of the objective and constraints.

Specifically, the canonical form of a quadratic program with linear constraints is, mathematically:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}\boldsymbol{\alpha} + \mathbf{f}^\top \boldsymbol{\alpha}, && (1) \\ \text{such that: } \mathbf{A}\cdot\boldsymbol{\alpha} \le \mathbf{a}, && (2) \\ \mathbf{B}\cdot\boldsymbol{\alpha} = \mathbf{b}. && (3) \end{aligned}$$

The vector $\boldsymbol{\alpha} \in \mathbb{R}^N$, where N is the number of training examples, contains the unknown variables to be solved for. The matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$ and vector $\mathbf{f} \in \mathbb{R}^N$ specify the quadratic objective; the matrix $\mathbf{A} \in \mathbb{R}^{k_{ineq}\times N}$ and vector $\mathbf{a} \in \mathbb{R}^{k_{ineq}}$ specify k_{ineq} inequality constraints. Similarly, $\mathbf{B} \in \mathbb{R}^{k_{eq}\times N}$ and vector $\mathbf{b} \in \mathbb{R}^{k_{eq}}$ specify k_{eq} equality constraints. Note that you can express a variety of equality constraints by adding rows to $\mathbf{A}$ and elements to $\mathbf{a}$; think how you would do it to express, e.g., a "greater or equal" constraint.

Problem 1 [20 points] Describe in detail how you would compute $\mathbf{H}$, $\mathbf{f}$, $\mathbf{A}$, $\mathbf{a}$, $\mathbf{B}$, and $\mathbf{b}$, to set up the dual optimization problem for the kernel SVM

$$\underset{\mathbf{w}}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max\left[0, 1 - y_i\left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) - w_0\right)\right] \right\}$$

given a kernel function $k(\cdot,\cdot)$ corresponding to the dot product in $\boldsymbol{\phi}$ space, and N training examples $(\mathbf{x}_i, y_i)$

II. Sentiment analysis

In this problem, we will develop a sentiment analysis tool. We have provided a data set containing short customer reviews (or snippets of reviews) for products. Each has been labeled as a positive or negative review. For instance, below is an example of a positive review

```
i downloaded a trial version of computer associates ez firewall and antivirus and
fell in love with a computer security system all over again .
```

and a negative one

```
i dont especially like how music files are unstructured ; basically they are
just dumped into one folder with no organization , like you might have in windows
explorer folders and subfolders .
```

from the training set. We will use Support Vector Machines to learn to classify such review sentences into positive and negative classes.
We will use the word occurrence features: if a particular word (or more generally, a token, which may include punctuation, numbers, etc.) w occurs in an example, the corresponding feature is set to 1, otherwise to 0.

Problem 2 [**40 points**] Fill in the missing pieces of code to fully implement the SVMs. This includes fleshing out the input to the optimization, and calculation of the model predictions. Using the provided `dev` (development) set as a validation set, tune an SVM predictor you think is best, and use it to compute and submit predictions on the `test` set to Kaggle: https://inclass.kaggle.com/c/cs475-sentiment-analysis.

You are free to experiment with various aspects of SVMs, but at the minimum please do the following:

1. Run linear SVM
2. Run at least one non-linear SVM (with some non-linear kernel)
3. Evaluate a range of values of C (regularization parameter) and the kernel-specific parameter(s) and discuss your findings – how do these values affect the performance of the classifier on training/validation data?

Some ideas for additional (optional) exploration:

- Consider various scaling on the input features, for instance, z-scoring or unit length normalization of each example.

- Using bigram features. Consider pairs of consecutive words, instead of, or in addition to, the unigram features (individual word occurences).

- Experiment with the frequency cutoff for including a word (or bigram) in the dictionary used to extract features. The default value for this we recommend is 5 (i.e., if a word appear less than 5 times in the data set it is ignored), but perhaps you will get better results with other values. This and the previous points are already possible with the code (see arguments in `utils.preprocess`).

- Once you identify good setting for your hyperparameters (C etc.) you could retrain the classifier on the combined `train` and `dev` sets, and then test it on `test`.

Note: you will need to install a convex optimization package `cvxopt` which is likely not included by default with your Python installation. If you are using Anaconda, you may be able to install is with the simple command

```
conda install -c omnia cvxopt
```

or

```
pip2 install cvxopt
```

You can find other instructions on how to install it here:
https://anaconda.org/anaconda/cvxopt
or here:
http://cvxopt.org/install/
Start working on this early, and ask course staff for help with the software issues, if you need it!

III. Graphical Models

Problem 3 [10 points] Consider the Bayesian Network given in Figure 1.

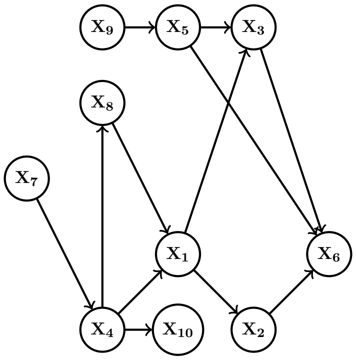


Figure 1: A directed graph

Are the sets **A** and **B** d-separated given set **C** for each of the following definitions of **A**, **B** and **C**? Justify each answer.

a. $\mathbf{A} = \{x_1\}$, $\mathbf{B} = \{x_9\}$, $\mathbf{C} = \{x_5, x_4\}$

b. $\mathbf{A} = \{x_2\}$, $\mathbf{B} = \{x_7\}$, $\mathbf{C} = \{x_1, x_9\}$

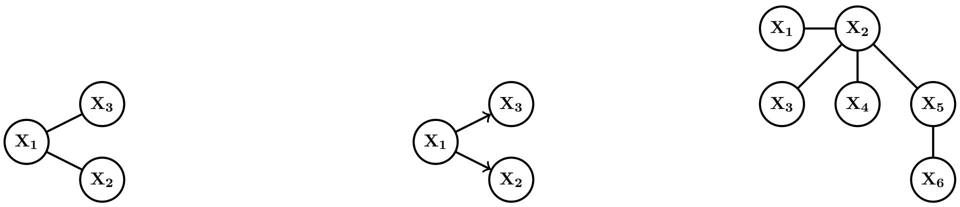
c. $\mathbf{A} = \{x_4\}$, $\mathbf{B} = \{x_5\}$, $\mathbf{C} = \{x_6, x_3\}$

d. $\mathbf{A} = \{x_4, x_8\}$, $\mathbf{B} = \{x_6, x_3\}$, $\mathbf{C} = \{x_1, x_7, x_9\}$

e. $\mathbf{A} = \{x_1\}$, $\mathbf{B} = \{x_2, x_3, x_4, x_5, x_8\}$, $\mathbf{C} = \{x_6, x_7, x_9, x_{10}\}$

Now assume that Figure 1 is a Markov Random Field, where each edge is undirected (just drop the direction of each edge.) Re-answer each of the above questions with justifications for your answers.

Problem 4 [10 points] A Markov Random Field usually cannot help us with the factorization of the distribution function. However, some MRFs can be converted to Bayesian Networks. For example, consider the graph structure in Figure 2a.



(a) Original undirected graph (b) Converted directed graph (c) An Undirected Graph

Figure 2: Graphs for Problem 4

From this graph, we know that X_2 and X_3 are conditionally independent given X_1. We can draw the corresponding directed graph as Figure 2b. This suggests the following factorization of the joint probability:

$$P(X_1, X_2, X_3) = P(X_3|X_1)P(X_2|X_1)P(X_1)$$

Now consider the graphical model in Figure 2c. As before, we can read the conditional independence relations from the graph.

(a) Following the example above, show a factorization of the joint distribution.

(b) Is this factorization unique, meaning, could you have written other factorizations that correspond this model?

(c) If the factorization is unique, show why it is unique. If it is not unique, provide an alternate factorization.

Problem 5 [20 points]

In this problem we will derive belief propagation for parameter estimation and prediction in a generative model with continuous variables.

Let $(X_i)_{i=1}^N$ be a sequence of real valued random variables whose values we observe as data. Let the sequence $\mathcal{D} = (x_i)_{i=1}^N$ represent the observed values of these random variables, i.e. $\mathcal{D}$ is our data. Our task is to predict the value of X_{N+1}. Imagine that we know special domain knowledge about the dynamics of the process that generated $\mathcal{D}$ in the sense that we know that the observed values are actually an interpolation of a sequence of hidden random variables $(H_i)_{i=0}^N$ whose values $(h_i)_{i=0}^N$ are generated as follows: h_0 comes from the gaussian distribution $\mathcal{N}(\mu_0/\lambda_0, \sigma^2/\lambda_0^2)$. h_i comes from the conditional gaussian disrtibution $\mathcal{N}(\kappa h_{i-1}, \sigma^2)$ and $x_i|h_{i-1}, h_i$ comes from $\mathcal{N}(\lambda h_{i-1} + (1-\lambda)h_i, \sigma^2)$. This process can be modelled via the factor graph in figure 3 with the following factors:

$$\xi_0(h_0) = \frac{\lambda_0^2}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\lambda_0 h_0 - \mu_0)^2}{2\sigma^2}\right) \tag{4}$$

$$\xi_i(x_i, h_i, h_{i-1}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - (\lambda h_{i-1} + (1-\lambda)h_i))^2}{2\sigma^2}\right) \exp\left(-\frac{(h_i - \kappa h_{i-1})^2}{2\sigma^2}\right) \tag{5}$$

$$\forall i \in [1, \ldots, N]$$

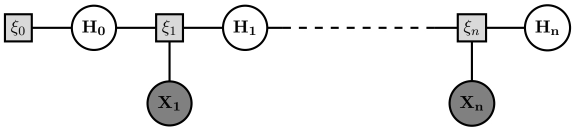


Figure 3: The Factor Graph of Observations

(a) Prove that $p(x_i|h_{i-1}) = \mathcal{N}((\lambda + \kappa(1-\lambda))h_{i-1}, \sigma^2(1 + (1-\lambda)^2))$.

You may find the following identities useful:

$$\int_{\mathbb{R}} \exp\left(-\frac{ax^2 + 2bx + c}{2\sigma^2}\right) dx = \sqrt{\frac{2\pi\sigma^2}{a}} \exp\left(-\frac{c - b^2/a}{2\sigma^2}\right), \qquad a > 0 \tag{6}$$

$$\int_{\mathbb{R}} \exp\left(-\frac{(ax-b)^2 + (cx-d)^2}{2\sigma^2}\right) dx = \sqrt{\frac{2\pi\sigma^2}{a^2 + c^2}} \exp\left(-\frac{(bc - ad)^2}{2\sigma^2(a^2 + c^2)}\right) \tag{7}$$

$$\int_{\mathbb{R}} \exp\left(-\frac{(ax-b)^2}{2\sigma_1^2} - \frac{(cx-d)^2}{2\sigma_2^2}\right) dx = \sqrt{\frac{2\pi}{a^2/\sigma_1^2 + b^2/\sigma_2^2}} \exp\left(-\frac{(bc - ad)^2}{2(a^2\sigma_2^2 + c^2\sigma_1^2)}\right) \tag{8}$$

(b) Prove that $p(h_i|x_i) = \mathcal{N}(\frac{\kappa x_i}{\lambda+(1-\lambda)\kappa}, \frac{\sigma^2(\lambda^2+\kappa^2)}{(\lambda+(1-\lambda)\kappa)^2})$.

(c) Prove that $p(x_{i+1}|x_i) = \mathcal{N}(\kappa x_i, \sigma^2(1 + \lambda^2 + (1 - \lambda)^2 + \kappa^2))$.

Now consider the scenario where the values of λ_0, μ_0, σ are known to us apriori but we do not know the multiplicative rate of change κ or the interpolation factor λ. A practical scheme that we can use in such a situation – where we want to make a prediction based on a model but we do not know all the system parameters – is the following maximum likelihood based method.

Step 1 Formulate the probability of the data $p(\mathcal{D}) = \{X_i\}_{i=1}^N$ as a function of the unknown parameters κ, λ.

Step 2 Maximize the probability of data to compute $\widehat{\kappa}^{\text{MLE}}, \widehat{\lambda}^{\text{MLE}}$.

Step 3 Use $\widehat{\kappa}, \widehat{\lambda}$ (omitting the MLE superscript) to compute the distribution of H_n and X_{N+1} and ultimately predict X_{N+1}.

Let $\mathcal{H}$ denote a sequence of values of the hidden random variables. As we saw in the class the sum-product algorithm (also called Belief Propagation), is an algorithm for efficiently computing $p(\mathcal{D})$. According to our model:

$$p(\mathcal{D}) = \int p(\mathcal{D}, \mathcal{H}) = \frac{1}{Z} \int_{h_0} \cdots \int_{h_N} \xi_0(h_0) \prod_{i=1}^{N} \xi_i(x_i, h_i, h_{i-1})$$

In the following questions you will compute the messages $\{m_i\}_{i=0}^N$, where message m_i is sent from factor $\xi_i \to H_i$ as shown in figure 4. Assume that H_N is chosen as the root node, and you run the belief propagation algorithm to compute messages $(m_i)_{i=0}^N$.

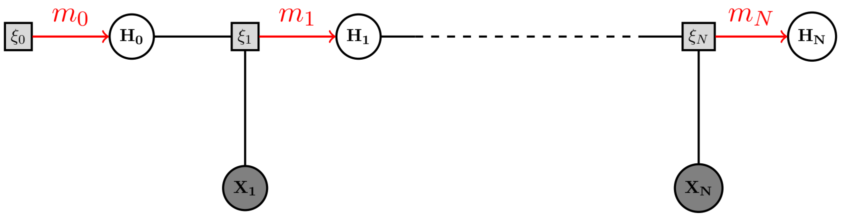


Figure 4: The Meessages from ξ_i to H_i.

(d) Express the probability of data, $p(\mathcal{D})$, in terms of m_N.

(e) What is $m_0 : \mathbb{R} \to \mathbb{R}_+$?

(f) Prove that

$$m_i = a_i \exp - \frac{(\lambda_i h_i - \mu_i)^2}{2\sigma^2}, \qquad \forall i \in [0, \ldots, N].$$

Is m_N a function of λ, κ? You do not need to fully simplify the expression.

(g) Assume that we maximized the value of $p(\mathcal{D})$ with respect to λ, κ to get the MLE estimates $\widehat{\kappa}, \widehat{\lambda}$. What will be the mean of the distribution of H_n given $\widehat{\kappa}, \widehat{\lambda}$ and $\mathcal{D}$? How will you compute the distribution of X_{N+1} from m_N?

(h) Can you think of a way to use the expression for $p(x_{i+1}|x_i)$ to estimate λ, κ from data? Is this method different from the maximum likelihood estimator that you derived above?