Machine Learning
Seoul National University

Homework #(**3**)
JARRY Guillaume

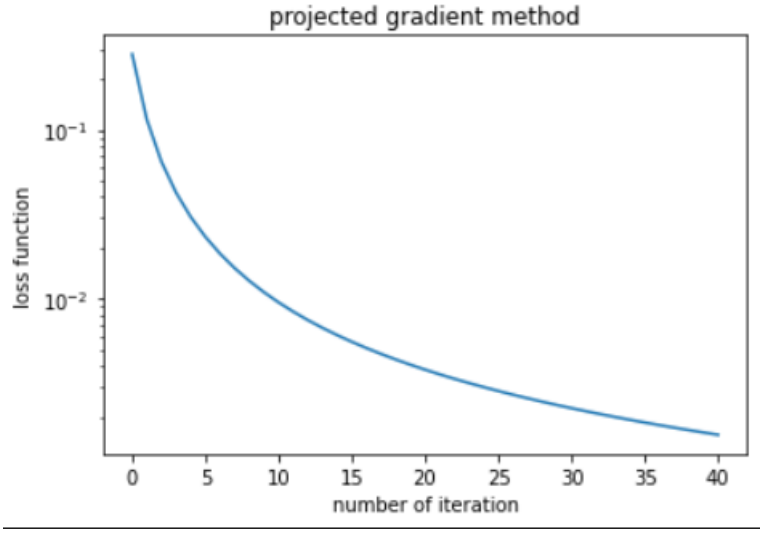
---

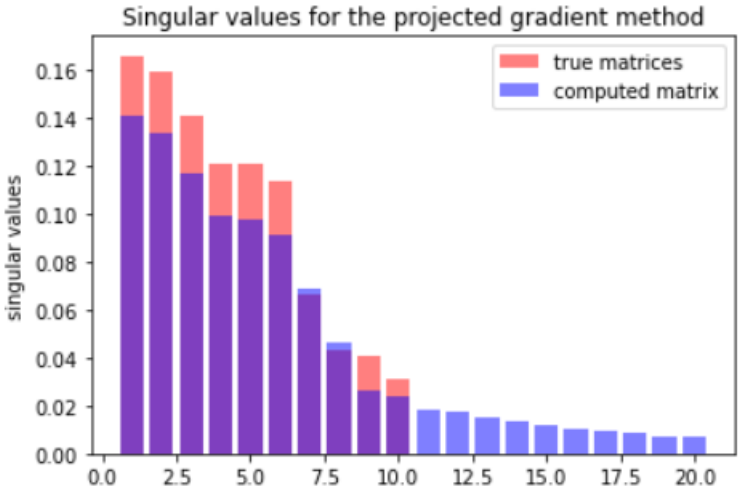
1 Q1

1) For this question, we have implemented the following code: (Let us remark that we didn't consider the multiplicity of the eigenvalues for the plotting, ie each bar has a unique height. We filtered the SVD to remove all duplicates).

```
    loss = [loss_function(X0, Y)]
    for step in step_size:
        X0 = X0 - step * (np.multiply(X0, O) - Y)
        X0 = nuclear_projection(X0)
        loss.append(loss_function(X0, Y))

    return X0, loss

X_min, loss = projected_gradient(X0, Y, step_size)
```

Which produced the two following figures:





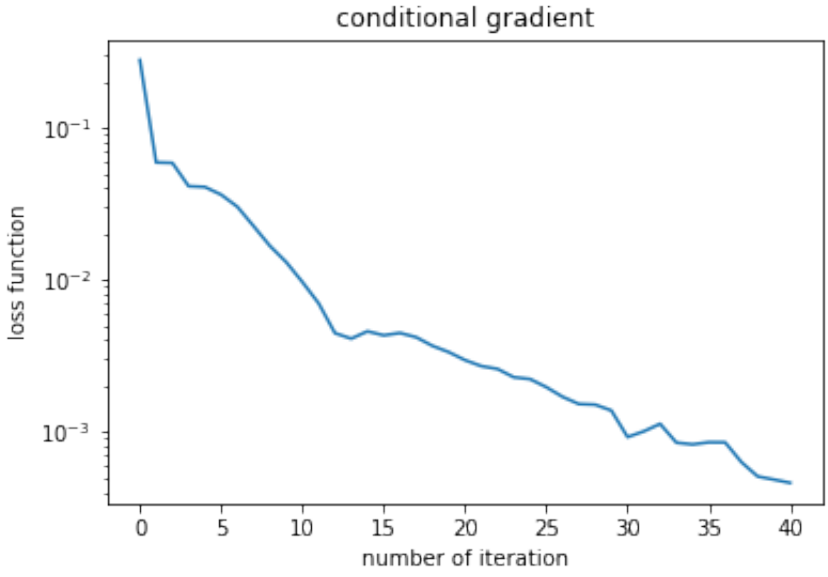
2) For this question, we have implemented the following code:

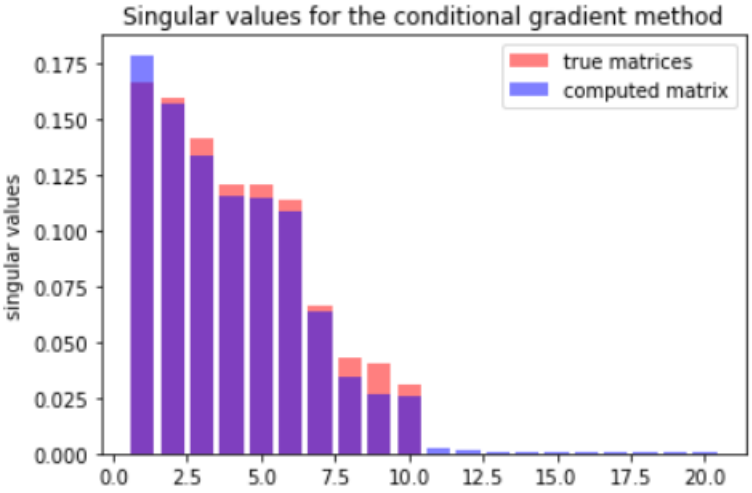
```
    x = np.random.normal(0, 1, A.shape[0])
    tmp = np.dot(A.T, x)
    y =  tmp/np.linalg.norm(tmp)
    for i in range(10):
        tmp1 = np.dot(A.T, y)
        x = tmp1/np.linalg.norm(tmp1)
        tmp2 = np.dot(A.T, x)
        y =  tmp2/np.linalg.norm(tmp2)

    return np.outer(x, y)

def conditional_gradient(X0, Y, O, step_size):
    loss = [loss_function(X0, Y)]

    for step in step_size:
        s = power_iteration(Y - (np.multiply(X0 , O)))
        X0 = (1 - step) * X0 + step * s
        loss.append(loss_function(X0, Y))

    return X0, loss

X, loss = conditional_gradient(X0, Y, O, step_size)
```

Which produced the following two graphs:

Machine Learning
Seoul National University

Homework #(3)
JARRY Guillaume





We can notice that the conditional gradient approximates the gap in the eigenvalues occuring between the tenth and the eleventh by having a more significant gap than the projected descent method.

2 Q2

1) The ordering is not topological because 10 is a parent of 3, not a descendent of 3. We therefore suggest the following topologycal ordering (descendiing order ie, 1 at the top): $\{1, 2, 7, 8, 6, 4, 10, 3, 9, 5\}$

2) We can see that the nodes $\{1, 2, 7, 8\}$ are independant from one another, therefore, computng the probability can be decomposed into (accounting for all the independance relationship):

$$P(X) = P(X_1)P(X_2)P(X_7)P(X_8) \times P(X6|X_2X_1)P(X_4|X_2, X_6, X_7) \times P(X_{10}|X_2)P(X_3|X_{10})$$

$$\times P(X_9|X_7, X_8, X_3)P(X_5|X_9)$$

We can then decide to group together terms of the above expression by cliques to get the standard factorization:

$$P(X) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C)$$

Where Z is the normalization factor and the set of cliques $\mathcal{C}$ is :

$$\mathcal{C} = \{C_1 = \{1, 2, 4, 6, 7\}, C_2 = \{2, 10, 3\}, C_3 = \{3, 9, 7, 8, 5\}\}$$

With a possible set of (ψ_C) being:

$$\psi_{C_1} = P(X_1)P(X_6|X_1, X_2)P(X_4|X_6, X_2, X_7)$$

$$\psi_{C_2} = P(X_2)P(X_{10}|X_2)P(X_3|X_{10})$$

$$\psi_{C_3} = P(X_7)P(X_8)P(X_9|X_7, X_8, X_3)P(X_5|X_9)$$

3) Assuming no conditioning, we get the following set of independent pair:

- (1,2) (1, 7) (1, 8) (1, 10) (1, 3) (1, 9) (1, 5) Independence of 1 with the descendants of 2, 7 (except 4 obviously) and 8
- (6, 7) (6, 8) (6, 9) (6, 5) (6, 3) (6, 10) Independence of 6 7 (except 4 obviously) and 8
- (8, 2) (8, 10) (8, 3) (8, 7) Independence of 8 with 2 and its descendant till 9 excluded and 7).
- (7, 2) (7, 8) (7, 10) (7, 3) Independence of 7 with 2 and ts descendants.
- (4, 8) Independence of 4 and 8.

4) We suggest: $A = \{10, 3, 5, 7, 8\}$

5) We suggest: $B = \{5, 1, 6\}$

3 Q3

1) We will prove this result by induction on the diameter of the tree. For the initilization, if $D = 1$, then of course, the sum product algorithm converges in one iteration.

Then, suppose the property is true for $D \in N$. Then consider a tree of diameter $D+1$ and let us write L the set of all leaf. For every element l of L, there will be only one node connected to l called s_l (because l is a leaf node, it is only connected to one other node). At the first iteration, we get:

$$\mu_{l \to s_l}(x_{s_l}) = \sum_{x_l} \psi_l(x_l)\psi_{l,s_l}(x_l, x_{s_l})$$

At each iteration, the messages sent by one leaf to its only neighboor is constant because it does not depend on messages from other node (since it is not connected to them). Therefore, we can reduce the number of nodes in our graph by removing every leaf in the tree and adequating for it by changing ψ_{s_l} to:

$$\psi^*_{s_l} = \psi_{s_l}(x_{s_l})\mu_{l \to s_l}(x_{s_l})$$

Because the message is always constant, the new graph will have the same factorization as the previous one. Let us call G' this new graph. It has diameter $D-1$ (since we remove the leaves at both extremes of the tree). Using the induction hypothesis, then, the SPA will converge on this new graph in at most the diameter $D-1$ iterations. Then, to compute the number of step of our algorithm, we need to add the first iteration (computing the first message of the leaves to the neighboor), and the last message from node s_l to the leaf. This gives $D+1$ iterations !

2) We will prove this property by induction on the number N of nodes in the tree.

For the initialization, if the tree has only one node, then the fixed points gives the only probability for the only node so it is trivial.

For the heredity, suppose the property true for $N-1$ and let us prove it for a tree containing N nodes. For the sake of conveniance, we will have the nodes of the tree named in such a way that N is a leaf and its only neighboor is 1. We will then use the same trick as the previous question, and noticing that at the first iteration, our leaf N sends the following message to 1:

$$P(X_1 = x_1, \dots, X_{N-1} = x_{N-1}) = \sum_{x_N} P(X_1 = x_1, \dots, X_N = x_N)$$

$$\propto \mu^*_{N \to 1}(x_1) \prod_{i=1}^{N-1} \psi_i(x_i) \prod_{j,k \in E\ (1,m)} \psi_{j,k}(x_j, x_k)$$

Therefore, we recognize thanks to the previous question that we can recreate an equivalent graph of $N-1$ nodes where the potential of node 1 is changed to: $\psi^*_1 = \psi_1(x_1)\mu_{N \to 1}(x_1)$, where $\mu_{N \to 1}(x_1)$ is constant because it is not modified by other messages from other nodes. On this new graph, thanks to the induction hypothesis, the SPA algorithm should converge to the proper messages and we will have the following equality:

$$\forall i \in \{1, \dots, N-1\}, \qquad P(X_i = x_i) = \psi_i(x_i) \prod j \in \mathcal{N}(i) \mu^*_{j \to i}(x_i)$$

ie, the probability of one node is proportional to the product of all incoming message and to the potential of that node.

Then we have to show that this property is also true for the last node N. Since there is only one incoming message, it becomes $p(x_m) \infty \psi_N(x_N)\mu^*_{1\to N}(x_N)$. And then, we also notice that $p(x_1)$ is a product of all incoming messages (including $\mu_{N\to 1}(x_1)$) and that:

$$\mu_{1\to N} = \sum_{x_1} \left(\psi_1(x_1)\psi_{1,N}(x_1, x_N) \prod_{i \in N(1)/\{N\}} \mu_{i\to 1}(x_1) \right)$$

We get:

$$\mu^*_{1\to N}(x_N) = \sum_{x_1} \frac{p(x_1)}{\mu^*_{N\to 1}(x_1)} \psi_{1,N}(x_1, x_N)$$

And so when we multiply by the potential we hope to get to the probablity:

$$\psi_N(x_N)\mu^*_{1\to N}(x_N) = \psi_N(x_N) \times \sum_{x1} \frac{p(x_1)}{\sum_{y_N} \psi_N(y_N)\psi_{1,N}(x_1, y_n)} \psi_{1,N}(x_1, x_N)$$

So then, we remind ourselves that, according to the defintion:

$$p(x_m|x_1) = \frac{p(x_1, x_m)}{p(x_1)} = \frac{\psi_N(x_N)\psi_{1,N}(x_1, x_N)}{\sum_{y_N} \psi_N(y_N)\psi_{1,N}(x_1, y_n)}$$

And therefore we recognize in the expression, thanks to the law of total probability :

$$\psi^*_N(x_N)\mu^*_{1\to N}(x_N) \infty \sum_{x_1} p(x_1)p(x_m|x_1) = p(x_m)$$