Advanced Machine Learning
2019 - 2020

Exam

Exercise 1

Based on your reading of the article "A Flexible EM-like Clustering Algorithm for High-Dimensional Noisy Data" by Roizman *et al.*, answer the following questions :

1. After a short summary of the article, give the main contributions of this work.
2. Regarding the clustering algorithm :
   (a) Define and explain the interest of the proposed statistical model.
   (b) Which parameters are used to characterize the clusters ?
   (c) What is the role of the τ_i's parameters ?
   (d) Justify the flexibility of the proposed algorithm ?
   (e) Is it a robust algorithm ?
   (f) Why this algorithm is adapted to high-dimensional settings ?
3. Compare the computational cost of the proposed algorithm versus the classical EM one (based on a multivariate Gaussian Mixture Model).
4. Table 4 : after recalling the metrics used for performance evaluation, explain the important difference obtained on Setups 3 and 4 between F-EM and the two others algorithms.
5. Regarding results in Table 8, explain why the spectral clustering performance strongly decreases when applying to MNIST 3-8-6 and MNIST 3-8-6+noise datasets.
6. In the case of unknown clusters number, propose an adaptation of the F-EM algorithm that accounts for the model order selection.

Exercise 2

1. Explain the general principle of stochastic optimization methods.
2. Give at least two applicative examples where such kind of approach is particularly useful.

   In binary classification, a useful loss function is the *logistic loss*, defined as :

$$(\forall x \in \mathbb{R}^n) \quad f(x) = \sum_{i=1}^{m} \log(1 + \exp(-y_i x^\top \theta_i))$$

   with $n \geq 1$ the number of parameters of the classifier, $m \geq 1$ the number of feature vectors, and for every $i \in \{1, \ldots, m\}$, y_i equals -1 or $+1$ is the label associated to the feature vector $\theta_i \in \mathbb{R}^n$.

3. Give the expression of the gradient of function f at a given $x \in \mathbb{R}^n$.
4. Deduce the expression of the stochastic gradient descent (SGD) algorithm applied to the minimization of f.
5. What is called "learning rate" in SGD ? How to tune this parameter so as to ensure the convergence of SGD.
6. Modify the above optimization problem, so as to include a LASSO regularizer.
7. What is called "regularization weight" in LASSO ? How to tune this parameter in practical applications ?
8. Why SGD method cannot be used for addressing the new problem ? Which kind of techniques could be used instead in this context ?