Advanced Machine Learning
Mini-Projects and In-class Reading Paper list

Lecturer: Aude Billard
Guest lecturers: Athanasios Polydoros, Basilio Noris
Teaching Assistants: Lukas Huber, Bernardo Fischera, Thomas Pethick
E-mails: aude.billard@epfl.ch, athanasios.polydoros@epfl.ch, lukas.huber@epfl.ch, bernardo.fichera@e

Spring Semester 2020

1 Introduction

The mini-project accounts for **50%** of the grade and entails either implementing an algorithm of choice and one or more variants from the following list, evaluating the algorithm's performance and sensibility to parameter choices, *or* preparing and presenting to the class a paper covering an advanced topic in machine learning.
Projects are assigned on a first come, first served basis. In-class reading of papers are done individually. Coding projects are done in a team of three. Teams should be formed and projects should be selected before **Tuesday, March 6th 2020** through moodle.
Code and reports for the coding project and slides for in-class reading should be submitted through moodle

2 Coding mini-projects

- Groups of 2 students.
- Max 2 teams can take the same coding project.
- Teams with the same coding projects have to use **different datasets**

Implementation: It should be handed in as a self-contained piece of **code** (coding language - matlab, python, C/C++) with dataset and corresponding demo scripts which analyze the use of the algorithm (selection of a dataset, systematic assessment of strength/weaknesses taking one algorithm as example, e.g. SVM or Boosting).

Report: The team should write a **report** (maximum 8 pages double column format). The results section should discuss important items, such as computational costs at training versus testing, growth of computational costs with dimension of dataset, number of hyper-parameters and how sensitive (if at all) the algorithm is to the choice of these hyper-parameters. This should be illustrated with figures and tables on the particular dataset.

Matlab toolboxes

Toolbox	URL
Matlab Toolbox for Dimensionality Reduction	`https://lvdmaaten.github.io/drtoolbox/`
Statistics and Machine Learning Toolbox	`http://fr.mathworks.com/help/stats/index.html`
Least Squares - Support Vector Machine	`http://www.esat.kuleuven.be/sista/lssvmlab/`
LIBSVM	`www.csie.ntu.edu.tw/~cjlin/libsvm/`
GMM/GMR v2.0	`http://lasa.epfl.ch/sourcecode/?showComments=14#GMM`
Probabilistic Modeling Toolkit for Matlab/Octave	`https://github.com/probml/pmtk3`
Support Vector Clustering Toolbox	`https://sites.google.com/site/daewonlee/research/svctoolbox`
Gaussian Processes (GPML toolbox)	`https://github.com/SheffieldML/GPmat`

Python toolboxes

Toolbox	URL
`scikit-learn`. Machine Learning in Python	`http://scikit-learn.org/stable/`
`bnpy`. Bayesian NonParametric Machine Learning for Python	`https://bitbucket.org/michaelchughes/bnpy/`
`GPy` Gaussian processes	`https://github.com/SheffieldML/GPy`

3 Topics for projects

Teams can choose among the following topics:

3.1 Dimensionality reduction / Manifold learning projects

1. Compare LLE to one of its 2 variants, MLLE or HLLE:

   - **LLE** (Locally Linear Embedding)
     Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." science 290.5500 (2000): 2323-2326.
   - **MLLE** (Modified LLE)
     Zhang, Zhenyue, and Jing Wang. "MLLE: Modified locally linear embedding using multiple weights." Advances in neural information processing systems. 2007.
   - **HLLE** (Hessian LLE)
     Donoho, David L., and Carrie Grimes. "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data." Proceedings of the National Academy of Sciences 100.10 (2003): 5591-5596.

   All implementations are found in *scikit-learn* for python. LLE and HLLE are found in *drtoolbox* for MATLAB.

2. Choose 2 of the following methods and compare them:

   - **SNE** (Stochastic Neighbor Embedding)
     Hinton, Geoffrey E., and Sam T. Roweis. "Stochastic neighbor embedding." Advances in neural information processing systems. 2003.
   - **tSNE** (t-distributed SNE)
     Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.
   - **GPLVM** (Gaussian Process Latent Variable Models)
     Lawrence, Neil. "Probabilistic non-linear principal component analysis with Gaussian process latent variable models." Journal of machine learning research 6.Nov (2005): 1783-1816.

   All implementations are found in in *scikit-learn* and *GPy* for python and in *drtoolbox* and *GPmat* for MATLAB.

3.2 Clustering projects

1. Compare the 2 clustering methods:

    - **Kernel K-means**
      Welling, Max. "Kernel K-means and Spectral Clustering." 2013-03-15]. http://www, ics. uci. edu/-welling/teaching/273 ASpring09/SpectralClustering. pdf. (Resource for the Advanced ML class)
    - **SV Clustering** (Support Vector Clustering)
      Ben-Hur, Asa, et al. "Support vector clustering." Journal of machine learning research 2.Dec (2001): 125-137. (Resource for the Advanced ML class)

    The implementation of Kernel K-means is found in the *ML toolbox* used in class and the implementation of SV Clustering is found in the *Support Vector Clustering Toolbox*, both for MATLAB.

3.3 Classification projects

Choose 2 of the following algorithms and compare them:

- **AdaBoost**
  Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of computer and system sciences 55.1 (1997): 119-139.
- **RTF** (Random Tree Forests)
  Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
  `https://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm`
- **Gaussian Process Classification**
  Williams, Christopher KI. "Prediction with Gaussian processes: From linear regression to linear prediction and beyond." Learning in graphical models. Springer, Dordrecht, 1998. 599-621.
- **Feed-forward Neural Networks**
  Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533-536.

The implementation of AdaBoost is found in the *ML toolbox* used in class and RTF is found in the *pmtk3* toolbox for MATLAB. All implementations for python are found in *scikit-learn.* GP is found at GPmat and GPy for Matlab and python respectively. NN is found in Tensorflow.

3.4 Regression projects

1. Compare the following two algorithms:

    - **SVR** (Support Vector Regression)
      Drucker, Harris, et al. "Support vector regression machines." Advances in neural information processing systems. 1997.
    - **LWPR** (Locally Weighted Projection Regression)
      Vijayakumar, Sethu, and Stefan Schaal. "Locally weighted projection regression: An O (n) algorithm for incremental real time learning in high dimensional space." Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000). Vol. 1. 2000.

2. Compare the following two algorithms:

    - **GMR** (Gaussian Mixture Regression)
      Sung, Hsi Guang. Gaussian mixture regression and classification. Diss. Rice University, 2004.
    - **GPR** (Gaussian Process Regression)
      Williams, Christopher KI. "Prediction with Gaussian processes: From linear regression to linear prediction and beyond." Learning in graphical models. Springer, Dordrecht, 1998. 599-621.

    The implementation of all algorithms is found in the *ML toolbox* used in class (GMR found in *gmmbox*), for MATLAB. SVR and GMR are also found in *scikit-learn* for python. GP is found at GPmat and GPy for Matlab and python respectively.

If you wish to perform a different comparison than that proposed in the list above, this is possible, but please check with the teacher first.

4 In-class paper reading and presentation

The paper reading is worth 50% of the grade and consists in a 10-minutes presentation in front of the class, followed by 10-minutes questions. The list of papers is available in the following subsections, each paper can be assigned to only one person. Selection of papers is done on a first come first serve basis. Paper reading is done individually. Grades are based on the following criteria: Clarity of the slides (25%), clarity of the speech (25%), timeliness (10%), quality of answers to the questions (40%).
We provide a list of relevant papers below. All papers refer partly to techniques seen in class and partly to techniques we have not seen. For instance, paper on learning metrics for semi-supervised clustering refers to linkage clustering, a technique not described in class. If you wish to present a different paper than that proposed in the list, this is possible, but please check with the teacher first.
Paper reading is intended to introduce you to extensions to machine learning techniques seen in class. Moreover, as you present the paper in front of the class, this will enable the class to learn about these techniques. ***It is hence of utmost importance that your presentation of the paper be intelligible to the class!*** When presenting the paper, recall that the audience is only partly familiar with the material. Think that you are teaching this, this means that you must introduce the concepts in a very clear and didactic manner.
It is expected that you will read not only the paper you must present in class, but also related papers. For instance, if you are not familiar with some of the terms or techniques used in the paper, you should read the references provided in the paper. Do not stop at reading the summary of these techniques provided in the paper, as you must be able to explain these other techniques if asked! You may want to also search papers that use the technique developed in the paper you must read.
Finally, keep a critical eye and make sure to offer to us your personal opinion about the strengths and weaknesses of the work you have read, going beyond what the authors state themselves. Offer your opinion!

4.1 Dimensionality reduction / Manifold learning papers

1. Tipping, Michael E. "Sparse kernel principal component analysis." Advances in neural information processing systems. 2001.

4.2 Clustering papers

2. Bach, Francis R., and Michael I. Jordan. "Learning spectral clustering." Advances in neural information processing systems. 2004.
3. Stella, X. Yu, and Jianbo Shi. "Multiclass spectral clustering." null. IEEE, 2003.
4. Kulis, Brian, et al. "Semi-supervised graph clustering: a kernel approach." Machine learning 74.1 (2009): 1-22.

4.3 Classification papers

5. Burr Settles. "Active Learning Literature Survey. Computer Sciences Technical Report 1648." University of Wisconsin-Madison. 2009.
6. Schohn, Greg, and David Cohn. "Less is more: Active learning with support vector machines." ICML. 2000.
7. Zhu, Ji, and Trevor Hastie. "Kernel logistic regression and the import vector machine." Journal of Computational and Graphical Statistics 14.1 (2005): 185-205.
8. Platt, John. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods." Advances in large margin classifiers 10.3 (1999): 61-74.

4.4 Regression papers

9. Quionero-Candela, Joaquin, and Carl Edward Rasmussen. "A unifying view of sparse approximate Gaussian process regression." Journal of Machine Learning Research 6.Dec (2005): 1939-1959.

4.5 Inverse reinforcement learning papers

10. Ziebart, Brian D., et al. "Maximum Entropy Inverse Reinforcement Learning." AAAI. Vol. 8. 2008.
11. Levine, Sergey, Zoran Popovic, and Vladlen Koltun. "Nonlinear inverse reinforcement learning with gaussian processes." Advances in Neural Information Processing Systems. 2011.

4.6 Time series analysis papers

12. Wang, Jack, Aaron Hertzmann, and David M. Blei. "Gaussian process dynamical models." Advances in neural information processing systems. 2006.

4.7 Deep learning papers

13. Shalev-Shwartz, Shai, Ohad Shamir, and Shaked Shammah. "Failures of Gradient-Based Deep Learning." International Conference on Machine Learning. 2017.
14. Sutskever, Ilya, et al. "On the importance of initialization and momentum in deep learning." International conference on machine learning. 2013.

4.8 Papers on general machine learning topics

15. Bottou, Lon, and Yann L. Cun. "Large scale online learning." Advances in neural information processing systems. 2004.
16. Ramyachitra, D., and P. Manikandan. "Imbalanced dataset classification and solutions: a review." International Journal of Computing and Business Research (IJCBR) 5.4 (2014).
17. Lu, Jing, et al. "Large scale online kernel learning." Journal of Machine Learning Research 17.47 (2016): 1.