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

Lecturer: Aude Billard
Main Teaching Assistants: Bernardo Fischera, Mikhail Koptev, Roland Schwann, Mert Kayaalp
E-mails: aude.billard@epfl.ch, bernardo.fichera@epfl.ch,
mikhail.koptev@epfl.ch, roland.schwan@epfl.ch, mert.kayaalp@epfl.ch

Spring Semester 2021

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 two. Teams should be formed and projects should be selected before **Friday, March 5th 2021** 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.

3.5 Reinforcement Learning projects

Students can choose and compare any two algorithms available on the following toolbox:
`https://stable-baselines.readthedocs.io/en/master`

3.6 Gaussian Process projects

Students can choose between the following 3 projects using the toolbox:
`https://sheffieldml.github.io/GPy/`

1. Compare Full Gaussian Processes and Sparse Gaussian Processes.
2. Compare classification with Gaussian Processes with another classifier of choice (e.g., SVMs, neural networks)
3. Compare the two implementations available for GP-LVM on the toolbox.

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 Manifold learning papers

1. Brehmer, Johann, and Cranmer, Kyle. "Flows for simultaneous manifold learning and density estimation." Advances in neural information processing systems. 2020.
2. Zhou, Yufan, et al. "Learning Manifold Implicitly via Explicit Heat-Kernel Learning." Advances in neural information processing systems. 2020.
3. Vladymyrov, Max ."No Pressure! Addressing the Problem of Local Minima in Manifold Learning Algorithms" . Advances in neural information processing systems. 2019.
4. Allassonniere, Stéphanie, et al. "Learning spatiotemporal piecewise-geodesic trajectories from longitudinal manifold-valued data". Advances in neural information processing systems. 2017.
5. Park, Mijung, et al."Bayesian Manifold Learning: The Locally Linear Latent Variable Model (LL-LVM)." Advances in neural information processing systems. 2015.

4.2 Spectral Clustering papers

6. Lainen, Steinar, and Sun, He. "Higher-Order Spectral Clustering of Directed Graphs." Advances in neural information processing systems. 2020.
7. Wan, Yali, and Meila, Marina ."A class of network models recoverable by spectral clustering." Advances in neural information processing systems. 2015.
8. Balakrishnan, Sivaraman , et al. "Noise Thresholds for Spectral Clustering." Advances in neural information processing systems. 2011.

9. Kumar, Abhishek, et al. "Co-regularized Multi-view Spectral Clustering". Advances in neural information processing systems. 2011.
10. Bach, Francis, and Jordan, Michael. "Learning Spectral Clustering." Advances in neural information processing systems. 2003.

4.3 SVM papers

11. Wang, Wenbo and Qiao, Xingye. "Learning Confidence Sets using Support Vector Machines."Advances in neural information processing systems. 2018.
12. Henao, Ricardo . "Bayesian Nonlinear Support Vector Machines and Discriminative Factor Modeling." Advances in neural information processing systems. 2014.
13. Grandvalet, Yves, et al. "Support Vector Machines with a Reject Option." Advances in neural information processing systems. 2008.
14. Luss, Ronny, and D'aspremont, Alexandre. "Support Vector Machine Classification with Indefinite Kernels". Advances in neural information processing systems. 2007.
15. Dekel, Ofer, and Singer, Yoram . "Support Vector Machines on a Budget"

4.4 Learning Kernels papers

16. Sinha, Aman, and Duchi, John C. "Learning Kernels with Random Features." Advances in neural information processing systems. 2016.
17. Cortes, Corinna, et al. "Learning Non-Linear Combinations of Kernels." Advances in neural information processing systems. 2009.

4.5 Gaussian Process papers

18. Tobar, Felipe et al."Learning Stationary Time Series using Gaussian Processes with Nonparametric Kernels." Advances in neural information processing systems. 2015.
19. Jensen, Kristopher et al. "Manifold GPLVMs for discovering non-Euclidean latent structure in neural data". Advances in neural information processing systems. 2020.

4.6 HMM papers

20. Hughes, Micheal C. et al. "Scalable Adaptation of State Complexity for Nonparametric Hidden Markov Models." Advances in neural information processing systems. 2015.
21. Liu, Yu-Ying, et al. "Efficient Learning of Continuous-Time Hidden Markov Models for Disease Progression." Advances in neural information processing systems. 2015.
22. Subakan, Cem et al. "Spectral Learning of Mixture of Hidden Markov Models" Advances in neural information processing systems. 2014.