DD2434/FDD3434 Machine Learning, Advanced Course Projects, 2021

Jens Lagergren and Aristides Gionis

Deadline, see Canvas

Read before starting

Please read the project instructions carefully before starting working on the projects.

The project assignment is built on six projects, three small and three large projects. You are to choose *one* of the available projects (see Assignments page on Canvas). In most of the projects, you are asked to reproduce subsets of the results presented in a specified scientific paper, but the level of detail in the instructions vary. The only evaluated outcome of your project work is a written report, though you might be asked to submit your code in special circumstances. More details about the report and the available projects can be found in the rest of this document.

You are allowed to discuss the project with other groups, but not the solutions. You are encouraged to use the Discussion board on Canvas. Your report will be automatically checked for similarities to other groups' reports. Make sure to cite the resources you used, such as textbooks, datasets or websites, during your project.

Your report should be submitted before the deadline using Canvas. Write clearly, stating all the assumptions you have made, and explain your logical steps and derivations. Show the results of your experiments using images and graphs together with your analysis. You should be prepared to share your code, bundled as easy-to-run scripts that can be used to generate your results, if asked.

Being able to communicate results and conclusions is a key aspect of scientific and corporate activities. It is up to you as an author to make sure that your report is well-written, precise, and self-contained. Based on the report, and only the report, we will decide if you pass the project.

The grading of the assignment will be as follows,

P The group completed the small/large project.

F The group didn't satisfy the requirements of the small/large project.

Good Luck!

General Information Regarding The Projects

Groups

There are maximum 4 students per group. Feel free to use Canvas Discussions page to find a project group. The number of group members do not affect the assessment criteria. We recommend PhD students to form a group together.

Only one student per group should submit their project report to Canvas before the deadline. As group members, you are responsible to make sure the report is submitted on your group's behalf.

Implementation

Each group should implement the method as described in the paper. You may use any programming language. You should implement the entire algorithm "from scratch" – to a reasonable extent. That is, for instance, you are allowed to use the functionalities available in PyTorch or TensorFlow in order to build neural networks. You are also allowed to use the standard packages used in the original paper. On the other hand, you are of course not allowed to use

`sklearn.decomposition.LatentDirichletAllocation`

to implement the LDA for [Blei et al., 2003] project. Especially, the code has to be your own, and all group members must participate actively in the implementation, i.e., write code. Each piece of code that you write should have a comment stating who contributed to that piece of code. You do not have to submit the code for assessment, but be prepared to show it upon request (for more information about code plagiarism, see EECS Code of Honor).

You will then perform a subset of the experiments described in the article to validate some of the key findings, for instance: can you reproduce the claimed run times, the reported accuracies, the likelihood scores etc?. Ideally, you should get the same results. If this is not the case, you must make an argument about possible reasons, and prove your argument by small experiments which you design and carry out yourself. A negative reproduction result together with a good argument gives an equally high grade as a positive reproduction result. In addition, you are encouraged to either extend (even just incremental) the proposed work; validate it on some other datasets (preferably using real data); test the scalability properties; suggest an effective implementation (code optimization); compare with new developments in the field based on the original approach; perform an ablation study or other experiments which you find informative. You are welcome to discuss your project scope with the TAs in charge of the project category.

Written Report

The article, the re-implementation with potential contributions/modifications, and your results are presented in a written report. Your PDF should be submitted through Canvas (*a single copy per group*) by the deadline. The report should be *at most* 7 pages including references, images, and tables. In addition, the report should have a (single) cover page with title, group number, author list, and short abstract (it does not count to the upper limit of 7 pages). The report should be written in good English.

In the report, you should first in the Introduction describe the article in such a level of detail that your peer students in this course understand the method, and so that it is clear to the reader that you understand the method too. You can put it in a wider context of the more recent state-of-the-art work that is built on the original contribution. Towards the end of the introduction, you should clearly and concisely outline the scope and objectives of your project.

You should then present your re-implementation or extension of the method in the Methods section, and your reproduction of the results in the Results section. Again, please communicate on such a level that your peer students understand what you have done, and so that it is clear to the reader what results you got and if, how, and why they deviate from the results presented in the original article.

Alternatively, if you extended or modified the original method please state clearly the motivation and your reasoning behind it, and describe how your results relate to those in the original paper.

Finally, you should critically discuss the methods and findings included in your report. Please, argue in favor and against the method in relation to the authors' arguments and their line of reasoning. In this Discussion section please provide a wider context potentially taking into account more recent developments built on the original contributions made in the paper (or as a countermeasure to these original contributions).

All statements made in the report (e.g., "method X is better than method Y") should be supported by either a reference to the original paper or report where the statement was made or if the statement originates from you, you should explain why this statement is true. If you make a quantitative comparison, please support your findings with appropriate statistical evidence (in the spirit of statistical hypothesis testing).

A technically correct, well-organized report with good language and a clear line of argument will have good chances of being accepted. Missed hand-in deadline, violations of the length and formatting requirements, as well as statements not supported by references will have a heavy negative effect on the chances of passing.

All group members must participate actively in the writing of the report. By adding a group member to the author list of the report, you certify that this person has written at least one section of the report.

1.1 Small Project - Dimensionality Reduction and Random Projections

Paper: Bingham, Ella, and Heikki Mannila. "Random projection in dimensionality reduction: applications to image and text data." Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. 2001.

Tasks for [Bingham and Mannila, 2001]:

1. Implement the methods discussed in the paper: RP, SRP, PCA, and DCT.
2. Build a collection of datasets by searching the internet. Try to find the datasets in the original paper and search for one additional dataset from each category: image, text, other. Try to include new datasets that are larger than the ones in the original paper.
3. Evaluate the different methods on the data you collected with respect to running time and approximation error.
4. Discuss whether the results of the paper are reproducible in your experiments, and whether they generalize to the new datasets you tested.
5. Discuss the original paper in terms of approach, methodology, and conclusions.
6. Summarize the most interesting things that you learned in this project.

1.2 Small Project - Variational Inference

Paper: Corduneanu, Adrian, and Christopher M. Bishop. "Variational Bayesian model selection for mixture distributions." Artificial intelligence and Statistics. Vol. 2001. Waltham, MA: Morgan Kaufmann, 2001.

Tasks for [Corduneanu and Bishop, 2001]:

1. Implement the EM-VI algorithm discussed in the paper from scratch.
2. Try to replicate the results on the synthetic and real data sets used in the paper.
3. Find other similar data sets and evaluate how well the model performs on these. Is it able to generalize?
4. Make some changes to the initialization of the model and discuss the differences you see.
5. Compare your results with a plain EM and cross validation. For this part you can use external packages.
6. Discuss the original paper in terms of approach, methodology, and conclusions.

1.3 Small Project - Variational Autoencoders

Paper: Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

Tasks for [Kingma and Welling, 2013]: In general, the same holds as for the other projects, namely; a technically correct, well organized report with good language and a clear line of argument will have good chances of being accepted. More specifically for this project,

1. There are many different algorithms used for benchmarking in the paper, you do not need to reproduce all of them. In this smaller project, you are not expected to implement any of the baselines.

2. Nevertheless, you should strive to understand the distinctions between the proposed algorithms and the benchmarks. Without this understanding it will be difficult to make compelling arguments for the relevance of the proposed algorithms.
3. Implement and train VAE on MNIST from scratch and show the generated samples. The basic components of your VAE could be either MLP or CNNs.
4. Change the dimensionality of latent space and plot the variational lower bound along with the training iterations. Does the performance get boosted as the dimensionality of latent space increases?
5. Briefly describe the difference between VAE and vanilla autoencoder. Try to generate images by sampling (e.g., averaging two latent representations) from the latent space of vanilla autoencoder and compare with the images generated by VAE.
6. You may use any deep learning programming language (tf, torch, jax, you name it). To run TensorFlow or PyTorch on the GPU (which will make training sufficiently faster), you need to have access to CUDA compatible NVIDIA GPUs. Many will not have such GPUs available on their local computers and will therefor need to resort to other platforms, such as

   - Google Colab – free and all required packages are installed. The Colab used in the exercise session is/will be available on Canvas.
   - Azure – $100 student sign-up voucher available at https://azure.microsoft.com/en-us/free/students/.
   - Google Cloud Platform – $300 dollar sign-up voucher at https://cloud.google.com/free.

   Note, however, that the experiments might not require GPU resources. That is, perhaps training using the CPU suffices (training will, however, be considerably slower).

1.4 Large Project - Network Representation Learning

Paper: Khosla, Megha, Vinay Setty, and Avishek Anand. "A comparative study for unsupervised network representation learning." IEEE Transactions on Knowledge and Data Engineering (2019).

Tasks for [Khosla et al., 2019]:

1. Implement the following methods discussed in the paper: DeepWalk, Node2vec, Line-1, NetMF, and GraphSage. Use of standard Machine Learning and Deep Learning languages is allowed.
2. Collect all the datasets presented in the paper. Search for a few additional datasets, if possible from different domains. Try to include new datasets that are at least as large as the ones in the original paper.
3. Evaluate the different methods on the data you collected with respect to the tasks of link prediction and node classification.
4. Discuss whether the results of the paper are reproducible in your experiments, and whether they generalize to the new datasets you tested.
5. Discuss the original paper in terms of approach, methodology, and conclusions.
6. Summarize the most interesting things that you learned in this project.

1.5 Large Project - Variational Inference

Paper: Blei, David M., and Michael I. Jordan. ”Variational methods for the Dirichlet process.” Proceedings of the twenty-first international conference on Machine learning. 2004.

Tasks for [Blei and Jordan, 2004]:

1. Implement the VI algorithm presented in the paper from scratch.
2. Generate synthetic data in the same way as it has been generated in the paper.
3. Find robot data used in the paper.
4. Try to replicate the VI results for both the simulated and robot data.
5. Comment on whether the results of the paper are reproducible in your experiments.
6. Discuss the original paper in terms of approach, methodology, and conclusions.

1.6 Large Project - Importance Weighted Autoencoders

Paper: Burda Yuri, Grosse Roger, Salakhutdinov Ruslan. “Importance Weighted Autoencoders.” ICRL 2016

Tasks for [Burda et al., 2015]: In general, the same holds as for the other projects, namely; a technically correct, well organized report with good language and a clear line of argument will have good chances of being accepted. More specifically for this project,

1. You may use any deep learning programming language (tf, torch, jax, you name it).
2. You need to implement and constantly compare with the VAE baseline in your experiments.
3. Especially, you should strive to understand the distinctions between the IWAE algorithm and the VAE. Without this understanding it will be difficult to make compelling arguments for the relevance of the proposed algorithms.
4. Put the findings from the paper in a more recent context. That is, how are the findings in the paper used today? Pay extra attention to how the novel lower bound (Eq. (8)) is utilized in recent papers.
5. Experiment with k during training and testing. What conclusions can you draw?
6. Can you come up with other experiments? Would it be informative to include other datasets, such as FashionMNIST? This is a plus and not a requirement for passing.
7. To run TensorFlow or PyTorch on the GPU (which will make training sufficiently faster), you need to have access to CUDA compatible NVIDIA GPUs. Many will not have such GPUs available on their local computers and will therefor need to resort to other platforms, such as

   - Google Colab – free and all required packages are installed. The Colab used in the exercise session is/will be available on Canvas.
   - Azure – $100 student sign-up voucher available at `https://azure.microsoft.com/en-us/free/students/`.
   - Google Cloud Platform – $300 dollar sign-up voucher at `https://cloud.google.com/free`.

   Note, however, that the experiments might not require GPU resources. That is, perhaps training using the CPU suffices (training will, however, be considerably slower).

References

[Bingham and Mannila, 2001] Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250.

[Blei and Jordan, 2004] Blei, D. M. and Jordan, M. I. (2004). Variational methods for the dirichlet process. In *Proceedings of the twenty-first international conference on Machine learning*, page 12.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[Burda et al., 2015] Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

[Corduneanu and Bishop, 2001] Corduneanu, A. and Bishop, C. M. (2001). Variational bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, volume 2001, pages 27–34. Morgan Kaufmann Waltham, MA.

[Khosla et al., 2019] Khosla, M., Setty, V., and Anand, A. (2019). A comparative study for unsupervised network representation learning. *IEEE Transactions on Knowledge and Data Engineering*.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.