

## 02460 Advanced Machine Learning **LOGBOOK**

Laura Perge, [s182257@student.dtu.dk](mailto:s182257@student.dtu.dk)

Daniel Horvath, [s172185@student.dtu.dk](mailto:s172185@student.dtu.dk)

Lorant Gulyas, [s172188@student.dtu.dk](mailto:s172188@student.dtu.dk)

### Overall Project Goals

- Recall Kernel Density Estimation (KDE), identify its use cases and limitations, highlight that its usability is overshadowed by the curse of dimensionality.
- Define the version of the Expectation Maximization (EM) algorithm that extends the KDE's density estimation in a way that it is less prone to the previously established issue. Outline and exemplify potential constraints and solutions.
- Code implementation of the version of the EM algorithm derived in point 2) for KDE and integrate both hold-out and leave-one-out cross-validation within the iteration loops.
- Test the code to measure and detect causes of inefficiency and integrate solutions where possible. Increase computational performance by preparing an alteration of the classical Multivariate Gaussian PDF based on the Cholesky decomposition.
- Derive and illustrate how the variational inference (VI) can be integrated into the KDE. Contrast it with the EM algorithm, emphasize and discuss the ability of the VI to return approximate distribution of the covariance instead of a point estimate.
- Hypothesize that the optimized version of the KDE enhanced with the EM algorithm can be used to impute randomly missing values in case of higher dimensionality. Measure and assess the performance of the enhanced KDE by experimenting on datasets.
- If time allows, debate the losses coming from restricting the covariance matrix to be strictly diagonal.
- Build a compact but comprehensive report about our findings.

**Project delimitation, specific hypotheses and aims are included in the Objectives, further hypotheses can be found in Section 1, Introduction of the Report.**

## **Project Meetings**

### **Week 6: 14.03.2019-17.03.2019**

**Questions:** How the structure of the algorithm should look like, EM algorithm outside and CV folds inside or the other way around? Why do we have to use a unified Sigma for every datapoint and how will we adjust this Sigma at every iteration? How this will affect the visualizations?

**Reading:** Christopher Bishop: Pattern Recognition and Machine Learning (Chapter 9, 10.2, 10.2); Tue Herlau, Mikkel N. Schmidt and Morten Mørup: Introduction to Machine Learning and Data Mining (Chapter 19). *Everybody on their own without meeting together.*

**Discussions:** Gathering our thoughts right after the supervisor meeting, summarizing our written notes into a nicer format. Clarifying our ideas how should the results look like on a 2D dataset (i.e.: faithful). Creating an implementation sketch to achieve the goals for the next supervisor meeting. *Everybody was there in person, we helped each other by spearing to make things clear.*

**Implementation:** Derivation of the equations discussed in the supervisor meeting on paper. Implementing the results in python for the hold-out method (*Laura and Daniel*).

Going through the derivation, implementing the results for hold-out CV and k-fold CV (*Lorant*).

**Results:** The log-likelihood is converging, but sometimes the code fails with singular matrix error.

### **Week 7: 18.03.2019 – 24.03.2019**

**Questions:** A frequent but not always happening runtime error about singular matrices.

**Implementation:** Clearing up the code, merging different solutions and best practices (*Laura*). Perform E and M steps in separate functions, until convergence is reached for hold-out method (*Daniel*). Create a function which plots the results in a color coded 2D graph (*Lorant*).

**Discussions:** Adjusting implementation plans to create CV folds first, do E step on all folds, do M step on all folds and repeat the last two steps until convergence.

**Results:** Visualization of result works, the algorithm is slow.

### **Week 8: 25.03.2019 – 31.03.2019**

**Questions:** Why is there a difference between our custom-made multivariate function and scipy's multivariate\_normal?

**Implementation:** Custom made multivariate function to decrease computational time (*Laura*). Code debugging to figure out the source of the difference (*Laura, Daniel and Lorant*).

**Discussion:** Discussing further steps on how the algorithm could be speeded up, clearing up notes (*Laura and Lorant*).

**Reading:** How Cholesky decomposition works and how can we apply this in our case (*Lorant*).

### **Week 9: 01.04.2019 – 07.04.2019**

**Questions:** The source of the difference is still unknown.

**Discussion:** Built in function for creating Cholesky decomposition gives results in a different format. Adjust code and calculations to this. (*Everyone*)

**Implementation:** Finishing up writing the code to perform KDE with EM for custom made probability distribution function which operates on a different base (the result of Cholesky decomposition). (*Everyone*)

**Results:** The results are now the same with the custom function. Significantly faster execution.

#### **Week 10: 08.04.2019 – 14.04.2019**

**Discussion:** How to use variational inference methods in our situation.

**Reading:** Blei, Kucukelbir, McAuliffe: Variational Inference: A Review for Statisticians (*Everyone*); Christopher Bishop: Pattern Recognition and Machine Learning (Chapter 10) (*Lorant*); watching an online lecture held by David M. Blei (<https://simons.berkeley.edu/talks/david-blei-2017-5-1>) (*Daniel & Lorant*);

**Implementation:** Measure and compare performance of both versions of GMM. Clean up code and repository. (*Daniel*)

#### **Week 11: 15.04.2019 – 21.04.2019**

**Question:** What should be a proper prior for the covariance? How do we derive the E step and the M step (the updates of parameters of distribution)? How do we derive the ELBO to track the convergence?

**Reading:** Lecture notes about VB (*Laura & Lorant*); Blei, Kucukelbir, McAuliffe: Variational Inference: A Review for Statisticians (*Everyone*); Alvarez, Niemi, Simpson: Bayesian Inference for a Covariance (*Laura*);

**Discussion:** Discussions about the questions we have. (*Everyone*)

**Implementation:** Derive equations to do E step update and M step update (*Laura*)

**Results:** Deriving the E step based on steps we used in the lecture and deriving M step until a point where we need the first moment of the Wishart distribution.

#### **Week 12: 22.04.2019 – 28.04.2019**

**Question:** How to impute missing attributes?

**Reading:** Multivariate normal distribution: Marginals and Conditionals

[<http://wwwf.imperial.ac.uk/~das01/MyWeb/M3S3/Handouts/MVN.pdf>] (*Lorant*);

Conditional Distributions [<https://newonlinecourses.science.psu.edu/stat505/node/43/>] (*Daniel & Lorant*);

Conditional distribution of a Gaussian, video [[https://www.youtube.com/watch?v=G6\\_OdMXpiVY](https://www.youtube.com/watch?v=G6_OdMXpiVY)] (*Daniel & Lorant*);

Multivariate normal distribution (Chapter 3.2: Conditional Distribution)

[[http://www.maths.manchester.ac.uk/~mkt/MT3732%20\(MVA\)/Notes/MVA\\_Section3.pdf](http://www.maths.manchester.ac.uk/~mkt/MT3732%20(MVA)/Notes/MVA_Section3.pdf)](*Lorant*);

Some probability and Statistics [<http://www.math.chalmers.se/~rootzen/highdimensional/SSP4SE-appA.pdf>] (*Everyone*);

**Implementation:** Data imputation for an observation where there's one attribute missing randomly (*Daniel*).

**Discussion:** Derive and understand conditional distribution equations when a data observation is missing one or two variables (*Daniel & Lorant*).

**Results:** Missing attribute imputation work with an approximate 15-20% error rate.

### **Week 13: 29.04.2019 – 05.05.2019**

**Questions:** How do we prove that our imputation is mathematically justified – Conditional Gaussians or Conditional KDE? How do we show our results in a comprehensive way? How should the report look like?

**Discussion:** report content (*Everyone*)

**Reading:** conditional expectation with KDE, conditional expectation in general

[<http://www.math.chalmers.se/~rootzen/highdimensional/SSP4SE-appA.pdf>] (*Laura*)

**Implementation:** Adding KNN missing value imputation and simple KDE (with only kernels not entire covariance matrices) for comparison purposes for imputation (*Lorant*); Adding median imputation as baseline, adding GMM EM algorithm for comparison in terms of density estimation, trials with real world dataset with >10 dimensions: check results/model run (*Laura*)

**Result:** report skeleton with notes of content per sections, the current missing value imputation code, issues with high-dimensional data (too much memory requirement)

### **Week 14: 06.05.2019 – 12.05.2019**

**Questions:** Still: How do we prove that our imputation is mathematically justified – Conditional Gaussians or Conditional KDE? Fix: How can we make sure our algorithms run properly with higher dimensional data?

How do we impute values in case of multiple missing dimensions? How do we present our results in the report?

**Reading:** [<https://pdfs.semanticscholar.org/ba0c/60992c29bfb207b7b6f46f1a200833cf0927.pdf>, <http://www.maths.manchester.ac.uk/~peterf/MATH38011/NPR%20N-W%20Estimator.pdf>] (*Laura*)

**Implementation:** Prepare the Nadaraya-Watson imputation model for high dimensional data, also switch to real data set. (*Daniel & Laura*). Also perform these updates on the benchmarking script (*Lorant*)

**Result:** Multidimensional imputation works with Nadaraya-Watson model, comparison with different kernels and other imputation methods.

### **Week 15: 13.05.2019 – 16.05.2019**

**Discussion:** Figuring out the best way to compare our imputation methods. (*Everyone*)

**Implementation:** Finalizing benchmarking scripts, creating plots and tables. Scaling data. Writing paper. (*Everyone*)

**Results:** Paper ready, python files uploaded and shared.

## Supervisor Meetings

### Week 6: 14.03.2019

*General discussion about the project*

*Gathering application areas*

*Guidelines for next week (implement EM algorithm for KDE with holdout CV first, and then apply it for k-fold CV)*

### Week 7: 21.03.2019

*Presentation of results since last meeting and discussing common mistakes*

*Hints for making the code faster and more efficient*

*Action points for next meeting (implement a resource efficient algorithm, perform cost tracking on the finished algorithm)*

### Week 10: 11.04.2019

*Finalize the optimization of the EM algorithm, discuss initialization problems to avoid ending up with local minimums, Consider constraining sigma for further speed-up*

*Implement Variational Inference for the next supervisor meeting*

### Week 12: 25.04.2019

*Focus on missing value imputation using conditional multivariate normal distribution because of time limitations*

### Week 12: 02.05.2019

*Guidelines about the report (what to include in it, how to structure it, what should be the main focus)*

*Ideas on how to benchmark and compare the custom multivariate KDE with other imputation procedures*

### Week 14: 13.05.2019

*Discussion questions about specific parts of our paper*