קורס מערכות לומדות קיץ 2021

הערות ודגשים לגבי התרגילים הרטובים

סיכום נושאים שהועברו

הכנה **לקרב**

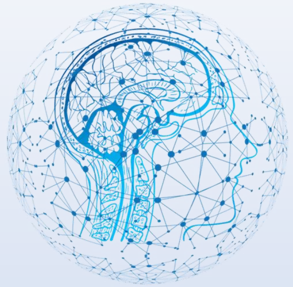
קורס מערכות לומדות קיץ 2021

הערות ודגשים לגבי התרגילים הרטובים

סיכום נושאים שהועברו

הכנה להאקט'ון

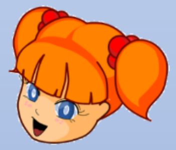
Session 2



- Ex. 0: basic script importing a Matlab .mat file to a *Pandas Dataframe*. Pandas Dataframes hold data in *named* columns with *indexed* rows. Useful tool, numerous use cases.
- Ex. 1: introduces NumPy and a simple vectorization solution.
- Ex. 2: introduces Matplotlib, a few common NumPy “tricks”.

<u>Session 2</u>

- <u>Ex. 0:</u> basic script importing a Matlab .mat file to a *Pandas Dataframe*. Pandas Dataframes hold data in *named* columns with *indexed* rows. Useful tool, numerous use cases.
- <u>Ex. 1:</u> introduces NumPy and a simple vectorization solution.
- <u>Ex. 2:</u> introduces Matplotlib, a few common NumPy “tricks”.

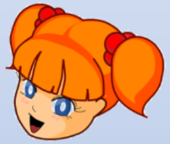
Take-home message:  > 

Session 3

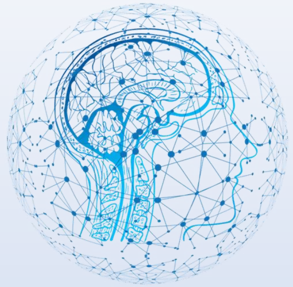
- Ex 3: PSD estimation on non-regularly sampled data.
  Main idea: demonstrate SciPy’s signal-processing capabilities.

Session 3 (cont')

- Ex 3: PSD estimation on non-regularly sampled data.
  Main idea: demonstrate SciPy's signal-processing capabilities.

  Take-home message:  >  > 

Session 3 (cont’)



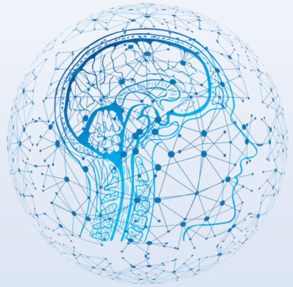
- Ex 4: Cython optimization, profiling.
  - Introduced lprun, Python’s line profiler.
  - Introduced Cython, Python’s static compiler.
    - Use NumPy arrays as “memoryviews”.
    - Demonstrate >1000 (~) speedup over native Python for loops.
    - Demonstrate bounds-check disabling.
    - Demonstrate parallel loop execution.
  - Take-home messages:
    - Typically not too difficult to obtain “native” machine speed.
    - Compiling also means worrying about cross-platform, etc.
    - Sometimes the best approach is **brute force.**

Session 4

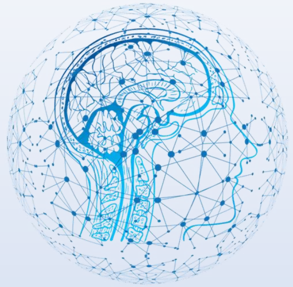
- Ex 5: Regressions
  - OLS (ordinary least squares) linear regression,
  - Robust linear regression,
  - Ordinary / robust nonlinear (polynomially-inflated) regression,
  - Leave-one-out testing.
  - Take-home message: not all statistics are sufficient statistics.

Session 4 (cont')

- Ex 6: feature extraction, parameter tuning / optimization
  - Time series feature extraction (via Pandas' rolling() function),
  - Introduction to Scikit-Learn's API: fit(), predict().
  - Controller parameter optimization via **Optuna's** black-box CMA-ES stochastic optimization.
  - Take-home messages:
    - When optimizing over a few (<= O(100)) parameters, **black-box** optimization can easily provide a solution (possibly global, but no performance guarantees).
    - The Optuna library is easy-to-use for such tasks.

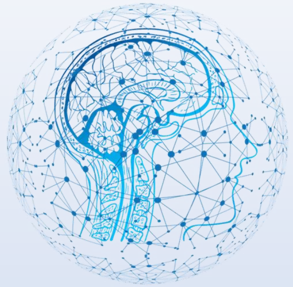


<u>Session 5</u>



- <u>Ex 7:</u> distribution drift (a.k.a. *domain shift*, *concept drift*)
  - Demonstration of the (in)famous XOR problem using a linear model.
  - Introduction and use of a few “classical” algorithms:
    - K-nearest-neighbors,
    - Logistic regression.
  - Introduction to a number of useful Scikit-Learn utility functions:
    - make_pipeline(),
    - train_test_split().
  - Introduction to Scikit-Learn’s score() API.
  - Take-home message: beware domain drift.

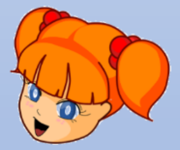
Session 6 (Midterm Hackathon)



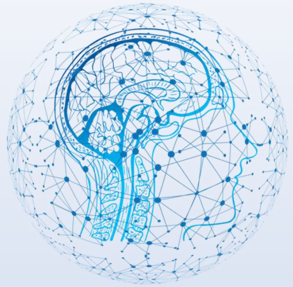
- Ex 8: midterm hackathon
- Take-home messages:
  - Using “all possible” features probably not a good idea.

  - Averaging (over some “natural” timeframe) can de-noise the features, assisting the downstream classification task.

  - If at all possible, decomposing the problem into “natural” sub-problems is typically the best alternative.



Session 7



- Ex 9: Noisy Features
  - Toy problem: XOR regression with noisy features.
  - Introduced a number of Scikit-Learn regression algorithms:
    - KNeighborsRegressor,
    - DecisionTreeRegressor,
    - AdaBoostRegressor.
  - Introduced the tqdm class.
  - Open problem: when using a 2nd degree polynomial LS regression, why on Earth does the cross-validated loss spike when the num. of noise features equals 4?
  - Clue (?): when using a 3rd degree polynomial LS regression, the loss spikes when the number of noise features equals 9.

Session 8

- Ex 10: CatBoost
  - Introduced CatBoostRegressor(), the CatBoost regression API.
  - Provides fast and reliable classification / regression.
  - Does not require feature scaling / normalization / etc.
  - Built-in support for categorical data (not cats).
  - Built-in support for regression uncertainty estimation via the ‘RMSEWithUncertainty’ loss function.

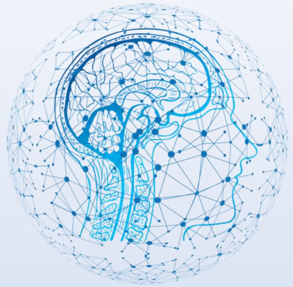
$$\mathrm{p}(y|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}) = \mathcal{N}(y|\mu^{(t)}, \sigma^{(t)}), \quad \{\mu^{(t)}, \log \sigma^{(t)}\} = F^{(t)}(\boldsymbol{x})\,. \tag{9}$$

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \mathbb{E}_{\mathcal{D}}[-\log \mathrm{p}(y|\boldsymbol{x}, \boldsymbol{\theta})] = -\frac{1}{N}\sum_{i=1}^{N} \log \mathrm{p}(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{\theta})\,. \tag{10}$$

Session 8 (cont')

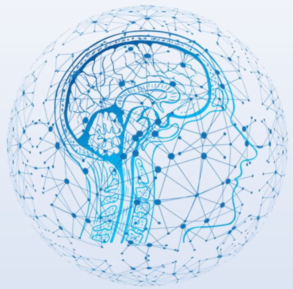
- Take-home messages:
  - CatBoost is a solid baseline when prototyping a new classifier / regressor.
  - Reliable, fast and accurate out-of-the-box performance.
  - Clear advantage over linear regression demonstrated in exercise.
- Additional topics:
  - Parameter tuning (Optuna).
  - Handling class imbalance: auto_class_weights.

Session 9



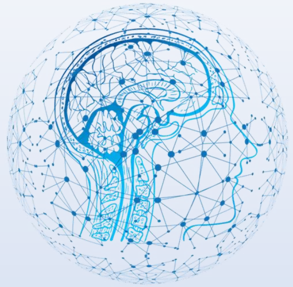
- Ex 11: Dimensionality Reduction
  - Introduced PCA (via Scikit-Learn) and UMAP dimensionality reduction.
  - PCA optimal in capturing maximal variance for given output dimensionality.
  - Take-home messages:
    - Proper tool for communication / compression.
    - Not necessarily optimal for classification.
    - Stable …
  - UMAP optimally embeds a neighbor graph (calculated in the original, high dimensional space) into a low dimensional space.

Session 9 (cont')



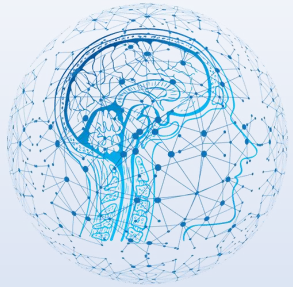
- UMAP:
  - Provides high-quality 2D / 3D visualizations,
  - Good starting point for clustering / outlier detection.
  - BUT:
    - Unstable: maps i.i.d. datasets to vastly different lower dim. embeddings.
    - Alternative loss function(s) may be preferable (see PACMAP).
- Additional topics:
  - Outlier Detection Algorithms: `IsolationForest`, `LocalOutlierFactor`.

Session 10



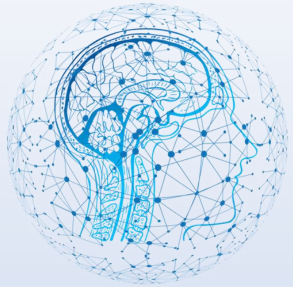
- Ex 12: Time Series Classification / Regression
- Main tool introduced: MiniROCKET (univariate, multivariate).
- Take-home messages:
    - Quick and easy baseline for time series classification / regression,
    - Achieves O(~optimal) performance on analytically tractable toy data,
    - On-par performance on relevant signal processing tasks:
    - Demonstrated (single sine wave) frequency estimation,
    - Demonstrated DTOA (delay estimation) between two noisy, time-shifted signals.
    - When tested on a wide range of real-world (univariate / multivariate) T.S. classification tasks, performance was competitive with (MUCH heavier) state-of-the-art machinery.
    - Begs the hypothesis: real-life (univariate / multivariate) time series probably NOT as complex as 2D (and beyond).

Session 11



- Ex 13: Clustering
  - Introduced HDBScan, a state-of-the-art clustering algorithm,
  - Demonstrated clustering via HDBScan for radar chirp periodicity detection (a.k.a. chaining),
- Take-home messages:
  - Provides a quick and solid baseline,
  - Most use cases involve optimizing a single parameter (min_cluster_size),
  - Can (in general) do well on up to ~50 to 100 dimensions,
  - Library includes a built-in outlier detection algorithm (GLOSH).

<u>Session 12</u>



- <u>Ex 14:</u> Ensemble methods via CatBoost
- <u>Idea:</u> CatBoost Ensembles (via Posterior Sampling) can provide total uncertainty estimates, as well as a decomposition into:
  - Data (aleatoric) uncertainty,
  - Knowledge (epistemic) uncertainty.
- Methods are considered state-of-the-art in modern NNs.
- Take-home messages:
  - <u>Results on toy problem:</u> less than spectacular.
  - Fancy formulas and colorful toy examples should always be taken with a grain of salt.

Session 12

Plain GB:

$$F^{(t)}(\boldsymbol{x}) = F^{(t-1)}(\boldsymbol{x}) + \epsilon h^{(t)}(\boldsymbol{x}) \,, \tag{7}$$

where $F^{(t-1)}$ is a model constructed at the previous iteration, $h^{(t)}(\boldsymbol{x}) \in \mathcal{H}$ is a weak learner chosen from some family of functionds $\mathcal{H}$, and ϵ is learning rate. The weak learner $h^{(t)}$ is usually chosen to approximate the negative gradient $-g^{(t)}(\boldsymbol{x}, y) := -\frac{\partial L(y,s)}{\partial s}\big|_{s=F^{(t-1)}(\boldsymbol{x})}$:

$$h^{(t)} = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}\big[\big(-g^{(t)}(\boldsymbol{x}, y) - h(\boldsymbol{x})\big)^2\big]. \tag{8}$$

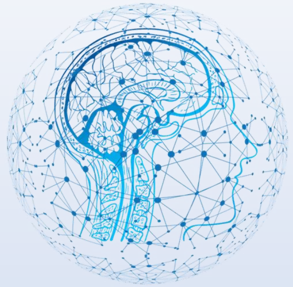
Stochastic Gradient Langevin Boosting:

$$h^{(t)} = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}\left[\left(-g^{(t)}(\boldsymbol{x}, y) - h(\boldsymbol{x}, \boldsymbol{\phi}) + \nu\right)^2\right], \nu \sim \mathcal{N}\left(0, \frac{2}{\beta\epsilon} I_{|\mathcal{D}|}\right) \,, \tag{11}$$

where β is the inverse diffusion temperature and $I_{|\mathcal{D}|}$ is an identity matrix. This random noise ν helps to explore the solution space in order to find the global optimum and the diffusion temperature controls the level of exploration. Second, the update $\boxed{7}$ is modified as:

$$F^{(t)}(\boldsymbol{x}) = (1 - \gamma\epsilon) F^{(t-1)}(\boldsymbol{x}) + \epsilon h^{(t)}(\boldsymbol{x}, \boldsymbol{\phi}^{(t)}) \,, \tag{12}$$

Relegated to Future



- Class imbalance.
- Outlier detection
- Out-of-Distribution (OoD) detection
- Uncertainty estimation
- Classifier calibration:
  - Logistic regression calibrated by-design.
  - Simple NNs already well-calibrated (but modern NNs extremely uncalibrated).
  - Not quite so useful when using CatBoost:
  - Plentiful datasets already fairly well calibrated,
  - Sparse datasets actually hopeless…
  - Calibration useful in various situations not encountered in the exercises (Naive Bayes classifiers, etc.)
- Interpretable ML: SHAP values