Machine learning

ML basics, linear models and sklearn/pandas.

Tommy Odland

Sonat

28. august 2020

Plan for the day

- 09:00: Lecture session 1
- 10:30: Problem session 1
- 11:30: Lunch
- 12:30: Lecture session 2
- 14:00: Problem session 2
- 16:00: Finished

Session 1 – Summary

- For our purposes, machine learning (ML) is essentially function approximation with the purpose of generalizing to unseen data.
- To create an ML algorithm, we need: (1) a function space, (2) a loss function and (3) an optimization algorithm.
- Linear regression finds weights $\mathbf{w}$ such that the predictions $\hat{y} = \sum_i w_i x_i = \mathbf{w}^\top \mathbf{x}$ are "close" to the true values y.
- The notion of "close" is formalized by minimizing a suitable norm of the error vector $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$, for instance the 2-norm (least squares).
- Feature engineering is the process of creating features ϕ_i from the original variables x_i, for instance using `BMI` instead of `weight` and `height` to predict `blood_pressure`. A linear model is linear in the features, but not necessarily the original variables.
- It's very easy to fool yourself. Use train/test/validation splits. Start with simple models. Look at the data. Balance theory and practice.

Session 2 – Summary

- Regularization involves penalizing model complexity in an effort to avoid overfitting the training data. In linear models, regularization means minimizing model fit plus model complexity:

$$\|\mathbf{y} - X\mathbf{w}\|_2^2 + \gamma \|\mathbf{w}\|_\ell^2$$

  - $\ell = 1$ is called LASSO (sparse **w**), $\ell = 2$ is called Ridge.
- Use one-hot encoding on nominal data. Aggregate relational data.
- Strategies for missing data: remove, use mean/median, impute, ...
- Generalized linear models predict $\hat{y} = f\left(\sum_i w_i x_i\right)$, where f is an activation function (e.g. sigmoid for logistic regression).
- ROC AUC is an example of a metric. The optimizer minimizes one loss function, but several metrics may be used to evaluate models.
- `scikit-learn` implements many algorithms in a unified API.
- Tree models are simple, powerful white-box models. A good model to learn more about. Boosting uses many trees in sequence.

References

The books are ordered by difficulty. The papers are all easy to read.

Books

- Chapters 1, 2, 3, 4, 6, 7 in "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Géron (also: checklist!)
- Chapters 3, 4 in "Pattern Recognition and Machine Learning" by Bishop
- Chapter 6 in "Convex Optimization" by Boyd et al
- Chapters 3, 4, 9, 10 in "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" by Hastie et al

Papers

- "A Few Useful Things to Know about Machine Learning" by Domingos
- "API design for machine learning software: experiences from the scikit-learn project" by Buitinck
- "Machine Learning: The High Interest Credit Card of Technical Debt" by Sculley et al.
- "Statistical Modeling: The Two Cultures" by Breiman