Wikipedia
The Free Encyclopedia

Feature engineering

Feature engineering or **feature extraction** or **feature discovery** is the process of extracting features (characteristics, properties, attributes) from raw data.[1] Due to deep learning networks, such as convolutional neural networks, that are able to learn it by itself, domain-specific- based feature engineering has become obsolete for vision and speech processing.

Other examples of features in physics include the construction of dimensionless numbers such as Reynolds number in fluid dynamics; then Nusselt number in heat transfer; Archimedes number in sedimentation; construction of first approximations of the solution such as analytical strength of materials solutions in mechanics, etc.[2]

Relevance

Features vary in significance.[3] Even relatively insignificant features may contribute to a model. Feature selection can reduce the number of features to prevent a model from becoming too specific to the training data set (overfitting).[4]

Explosion

Feature explosion occurs when the number of identified features grows inappropriately. Common causes include:

- Feature templates - implementing feature templates instead of coding new features
- Feature combinations - combinations that cannot be represented by a linear system

Feature explosion can be limited via techniques such as: regularization, kernel methods, and feature selection.[5]

Automation

Automation of feature engineering is a research topic that dates back to the 1990s.[6] Machine learning software that incorporates automated feature engineering has been commercially available since 2016.[7] Related academic literature can be roughly separated into two types:

- Multi-relational decision tree learning (MRDTL) uses a supervised algorithm that is similar to a decision tree.
- Deep Feature Synthesis uses simpler methods.

Multi-relational decision tree learning (MRDTL)

MRDTL generates features in the form of SQL queries by successively adding clauses to the queries. For instance, the algorithm might start out with

```
SELECT COUNT(*) FROM ATOM t1 LEFT JOIN MOLECULE t2 ON t1.mol_id = t2.mol_id GROUP BY
t1.mol_id
```

The query can then successively be refined by adding conditions, such as "WHERE t1.charge <= -0.392".

However, most MRDTL studies base implementations on relational databases, which results in many redundant operations. These redundancies can be reduced by using techniques such as tuple id propagation.[8][9] Efficiency can be increased by using incremental updates, which eliminates redundancies.[10]

Open-source implementations

There are a number of open-source libraries and tools that automate feature engineering on relational data and time series:

- **featuretools** is a Python library for transforming time series and relational data into feature matrices for machine learning.[11][12][13]
- **OneBM** or One-Button Machine combines feature transformations and feature selection on relational data with feature selection techniques.[14]

> [OneBM] helps data scientists reduce data exploration time allowing them to try and error many ideas in short time. On the other hand, it enables non-experts, who are not familiar with data science, to quickly extract value from their data with a little effort, time, and cost.[15]

- **getML community** is an open source tool for automated feature engineering on time series and relational data.[16][17] It is implemented in C/C++ with a Python interface.[18] It has been shown to be at least 60 times faster than tsflex, tsfresh, tsfel, featuretools or kats.[19]
- **tsfresh** is a Python library for feature extraction on time series data.[20] It evaluates the quality of the features using hypothesis testing.[21]
- **tsflex** is an open source Python library for extracting features from time series data.[22] Despite being 100% written in Python, it has been shown to be faster and more memory efficient than tsfresh, seglearn or tsfel.[23]
- **seglearn** is an extension for multivariate, sequential time series data to the scikit-learn Python library.[24]
- **tsfel** is a Python package for feature extraction on time series data.[25]
- **kats** is a Python toolkit for analyzing time series data.[26]

Deep feature synthesis

The deep feature synthesis (DFS) algorithm beat 615 of 906 human teams in a competition.[27][28]

Feature stores

The Feature Store is where the features are stored and organized for the explicit purpose of being used to either train models (by data scientists) or make predictions (by applications that have a trained model). It is a central location where you can either create or update groups of features created from multiple different data sources, or create and update new datasets from those feature groups for training models or for use in applications that do not want to compute the features but just retrieve them when it needs them to make predictions.[29]

A feature store includes the ability to store code used to generate features, apply the code to raw data, and serve those features to models upon request. Useful capabilities include feature versioning and policies governing the circumstances under which features can be used.[30]

Feature stores can be standalone software tools or built into machine learning platforms.

Alternatives

Feature engineering can be a time-consuming and error-prone process, as it requires domain expertise and often involves trial and error.[31][32] Deep learning algorithms may be used to process a large raw dataset without having to resort to feature engineering.[33] However, it's important to note that deep learning algorithms still require careful preprocessing and cleaning of the input data.[34] In addition, choosing the right architecture, hyperparameters, and optimization algorithm for a deep neural network can be a challenging and iterative process.[35]

See also

- Covariate
- Data transformation
- Feature extraction
- Feature learning
- Hashing trick
- Kernel method
- List of datasets for machine learning research
- Space mapping
- Instrumental variables estimation

References

1. "Machine Learning and AI via Brain simulations" (https://ai.stanford.edu/~ang/slides/DeepLearning-Mar2013.pptx). *Stanford University*. Retrieved 2019-08-01.
2. MacQueron C (2021). SOLID-LIQUID MIXING IN STIRRED TANKS : Modeling, Validation, Design Optimization and Suspension Quality Prediction (https://www.researchgate.net/publication/353947052) (Report). doi:10.13140/RG.2.2.11074.84164/1 (https://doi.org/10.13140%2FRG.2.2.11074.84164%2F1).
3. "Feature Engineering" (http://www.cs.princeton.edu/courses/archive/spring10/cos424/slides/18-feat.pdf) (PDF). 2010-04-22. Retrieved 12 November 2015.
4. "Feature engineering and selection" (http://www.cs.berkeley.edu/~jordan/courses/294-fall09/lectures/feature/slides.pdf) (PDF). Alexandre Bouchard-Côté. October 1, 2009. Retrieved 12 November 2015.

5. "Feature engineering in Machine Learning" (https://web.archive.org/web/20160304112056/https://ufal.mff.cuni.cz/~zabokrtsky/courses/npfl104/html/feature_engineering.pdf) (PDF). Zdenek Zabokrtsky. Archived from the original (https://ufal.mff.cuni.cz/~zabokrtsky/courses/npfl104/html/feature_engineering.pdf) (PDF) on 4 March 2016. Retrieved 12 November 2015.
6. Knobbe AJ, Siebes A, Van Der Wallen D (1999). "Multi-relational Decision Tree Induction" (https://link.springer.com/content/pdf/10.1007/978-3-540-48247-5_46.pdf) (PDF). *Principles of Data Mining and Knowledge Discovery*. Lecture Notes in Computer Science. Vol. 1704. pp. 378–383. doi:10.1007/978-3-540-48247-5_46 (https://doi.org/10.1007%2F978-3-540-48247-5_46). ISBN 978-3-540-66490-1.
7. "Its all about the features" (https://reality.ai/it-is-all-about-the-features/). *Reality AI Blog*. September 2017.
8. Yin X, Han J, Yang J, Yu PS (2004). "CrossMine: Efficient classification across multiple database relations". *Proceedings. 20th International Conference on Data Engineering*. pp. 399–410. doi:10.1109/ICDE.2004.1320014 (https://doi.org/10.1109%2FICDE.2004.1320014). ISBN 0-7695-2065-0. S2CID 1183403 (https://api.semanticscholar.org/CorpusID:1183403).
9. Frank R, Moser F, Ester M (2007). "A Method for Multi-relational Classification Using Single and Multi-feature Aggregation Functions". *Knowledge Discovery in Databases: PKDD 2007*. Lecture Notes in Computer Science. Vol. 4702. pp. 430–437. doi:10.1007/978-3-540-74976-9_43 (https://doi.org/10.1007%2F978-3-540-74976-9_43). ISBN 978-3-540-74975-2.
10. "How automated feature engineering works - The most efficient feature engineering solution for relational data and time series" (https://get.ml/resources/how-getml-works). Retrieved 2019-11-21.
11. "What is Featuretools?" (https://featuretools.alteryx.com/en/stable/). Retrieved September 7, 2022.
12. "Featuretools - An open source python framework for automated feature engineering" (https://www.featuretools.com). Retrieved September 7, 2022.
13. "github: alteryx/featuretools" (https://github.com/alteryx/featuretools). *GitHub*. Retrieved September 7, 2022.
14. Thanh Lam, Hoang; Thiebaut, Johann-Michael; Sinn, Mathieu; Chen, Bei; Mai, Tiep; Alkan, Oznur (2017-06-01). "One button machine for automating feature engineering in relational databases". arXiv:1706.00327 (https://arxiv.org/abs/1706.00327) [cs.DB (https://arxiv.org/archive/cs.DB)].
15. Thanh Lam, Hoang; Thiebaut, Johann-Michael; Sinn, Mathieu; Chen, Bei; Mai, Tiep; Alkan, Oznur (2017-06-01). "One button machine for automating feature engineering in relational databases". arXiv:1706.00327 (https://arxiv.org/abs/1706.00327) [cs.DB (https://arxiv.org/archive/cs.DB)].
16. "getML documentation" (https://docs.getml.com/latest/). Retrieved September 7, 2022.
17. "github: getml/getml-community" (https://github.com/getml/getml-community). *GitHub*. Retrieved September 7, 2022.
18. "github: getml/getml-community" (https://github.com/getml/getml-community). *GitHub*. Retrieved September 7, 2022.
19. "github: getml/getml-community" (https://github.com/getml/getml-community). *GitHub*. Retrieved September 7, 2022.
20. "tsfresh documentation" (https://tsfresh.readthedocs.io/en/latest). Retrieved September 7, 2022.
21. "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)" (https://www.researchgate.net/publication/324948288). Retrieved September 7, 2022.

22. "predict-idlab/tsflex" (https://github.com/predict-idlab/tsflex). *GitHub*. Retrieved September 7, 2022.
23. Van Der Donckt, Jonas; Van Der Donckt, Jeroen; Deprost, Emiel; Van Hoecke, Sofie (2022). "tsflex: Flexible time series processing & feature extraction" (https://www.sciencedirect.com/science/article/pii/S2352711021001904). *SoftwareX*. **17**: 100971. arXiv:2111.12429 (https://arxiv.org/abs/2111.12429). Bibcode:2022SoftX..1700971V (https://ui.adsabs.harvard.edu/abs/2022SoftX..1700971V). doi:10.1016/j.softx.2021.100971 (https://doi.org/10.1016%2Fj.softx.2021.100971). S2CID 244527198 (https://api.semanticscholar.org/CorpusID:244527198). Retrieved September 7, 2022.
24. "seglearn user guide" (https://dmbee.github.io/seglearn/user_guide.html). Retrieved September 7, 2022.
25. "Welcome to TSFEL documentation!" (https://tsfel.readthedocs.io/en/latest/). Retrieved September 7, 2022.
26. "github: facebookresearch/Kats" (https://github.com/facebookresearch/Kats). *GitHub*. Retrieved September 7, 2022.
27. "Automating big-data analysis" (https://news.mit.edu/2015/automating-big-data-analysis-1016). 16 October 2015.
28. Kanter, James Max; Veeramachaneni, Kalyan (2015). "Deep feature synthesis: Towards automating data science endeavors". *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. pp. 1–10. doi:10.1109/DSAA.2015.7344858 (https://doi.org/10.1109%2FDSAA.2015.7344858). ISBN 978-1-4673-8272-4. S2CID 206610380 (https://api.semanticscholar.org/CorpusID:206610380).
29. "What is a feature store" (https://www.featurestore.org/what-is-a-feature-store). Retrieved 2022-04-19.
30. "An Introduction to Feature Stores" (https://phaseai.com/resources/intro-to-feature-stores). Retrieved 2021-04-15.
31. "Feature Engineering in Machine Learning" (https://www.section.io/engineering-education/feature-engineering-in-machine-learning/). *Engineering Education (EngEd) Program | Section*. Retrieved 2023-03-21.
32. explorium_admin (2021-10-25). "5 Reasons Why Feature Engineering is Challenging" (https://www.explorium.ai/blog/5-reasons-why-feature-engineering-is-challenging/). *Explorium*. Retrieved 2023-03-21.
33. Spiegelhalter, D. J. (2019). *The art of statistics : learning from data* (https://www.worldcat.org/oclc/1064776283). [London] UK. ISBN 978-0-241-39863-0. OCLC 1064776283 (https://www.worldcat.org/oclc/1064776283).
34. Sarker IH (November 2021). "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8372231). *SN Computer Science*. **2** (6): 420. doi:10.1007/s42979-021-00815-1 (https://doi.org/10.1007%2Fs42979-021-00815-1). PMC 8372231 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8372231). PMID 34426802 (https://pubmed.ncbi.nlm.nih.gov/34426802).
35. Bengio, Yoshua (2012), "Practical Recommendations for Gradient-Based Training of Deep Architectures" (https://dx.doi.org/10.1007/978-3-642-35289-8_26), *Neural Networks: Tricks of the Trade*, Lecture Notes in Computer Science, vol. 7700, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 437–478, doi:10.1007/978-3-642-35289-8_26 (https://doi.org/10.1007%2F978-3-642-35289-8_26), ISBN 978-3-642-35288-1, S2CID 10808461 (https://api.semanticscholar.org/CorpusID:10808461), retrieved 2023-03-21

Further reading

- Boehmke B, Greenwell B (2019). "Feature & Target Engineering". *Hands-On Machine*

Learning with R. Chapman & Hall. pp. 41–75. ISBN 978-1-138-49568-5.

- Zheng A, Casari A (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly. ISBN 978-1-4919-5324-2.
- Zumel N, Mount (2020). "Data Engineering and Data Shaping". *Practical Data Science with R* (2nd ed.). Manning. pp. 113–160. ISBN 978-1-61729-587-4.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Feature_engineering&oldid=1170051454"