Feature engineering

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself [1].

Contents

Features

A feature is an attribute or property shared by all of the independent units on which analysis or prediction is to be done. Any attribute could be a feature, as long as it is useful to the model.

The purpose of a feature, other than being an attribute, would be much easier to understand in the context of a problem. A feature is a characteristic that might help when solving the problem.[2]

Importance

Features are important to predictive models and influence results [3].

It is asserted that feature engineering plays an important part of Kaggle competitions [4] and machine learning project's success or failure [5].

Process

The feature engineering process is:[6]

- Brainstorming or testing features;[7]
- Deciding what features to create;
- Creating features;
- Checking how the features work with your model;

- Improving your features if needed;
- Go back to brainstorming/creating more features until the work is done.

Relevance

A feature could be strongly relevant (i.e., the feature has information that doesn't exist in any other feature), relevant, weakly relevant (some information that other features include) or irrelevant.[8] Even if some features are irrelevant, having too many is better than missing those that are important. Feature selection can be used to prevent overfitting.[9]

Feature explosion

Feature explosion can be caused by feature combination or feature templates, both leading to a quick growth in the total number of features.

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

Feature explosion can be stopped via techniques such as: regularization, kernel method, feature selection.[10]

Automation

Automation of feature engineering is a research topic that dates back to at least the late 1990s.[11] The academic literature on the topic can be roughly separated into two strings: First, Multi-relational decision tree learning (MRDTL), which uses a supervised algorithm that is similar to a decision tree. Second, more recent approaches, like Deep Feature Synthesis, which use simpler methods.

Multi-relational decision tree learning (MRDTL) generates features in the form of SQL queries by successively adding new clauses to the queries.[12] 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".[13]

However, most of the academic studies on MRDTL use implementations based on existing relational databases, which results in many redundant operations. These redundancies can be reduced by using tricks such as tuple id propagation.[14][15] More recently, it has been demonstrated that the efficiency can be increased further by using incremental updates, which completely eliminates redundancies.[16]

In 2015, researchers at MIT presented the Deep Feature Synthesis algorithm and demonstrated its effectiveness in online data science competitions where it beat 615 of 906 human teams.[17][18] Deep Feature Synthesis is available as an open source library called Featuretools.[19] That work was followed by other researchers including IBM's OneBM[20] and Berkeley's ExploreKit.[21] The researchers at IBM stated that feature engineering automation "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."

See also

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

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. "Discover Feature Engineering, How to Engineer Features and How to Get Good at It - Machine Learning Mastery" (http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/). *Machine Learning Mastery*. Retrieved 2015-11-11.
3. "Feature Engineering: How to transform variables and create new ones?" (http://www.analyticsvidhya.com/blog/2015/03/feature-engineering-variable-transformation-creation/). *Analytics Vidhya*. 2015-03-12. Retrieved 2015-11-12.
4. "Q&A with Xavier Conort" (https://blog.kaggle.com/2013/04/10/qa-with-xavier-conort/). *kaggle.com*. 2013-04-10. Retrieved 12 November 2015.
5. Domingos, Pedro (2012-10-01). "A few useful things to know about machine learning" (http://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf) (PDF). *Communications of the ACM*. **55** (10): 78–87. doi:10.1145/2347736.2347755 (https://doi.org/10.1145%2F2347736.2347755).
6. "Big Data: Week 3 Video 3 - Feature Engineering" (https://www.youtube.com/watch?v=drUToKxEAUA). *youtube.com*.
7. Jalal, Ahmed Adeeb (January 1, 2018). "Big data and intelligent software systems" (https://content.iospress.com/articles/international-journal-of-knowledge-based-and-intelligent-engineering-systems/kes180383). *International Journal of Knowledge-based and Intelligent Engineering Systems*. **22** (3): 177–193. doi:10.3233/KES-180383 (https://doi.org/10.3233%2FKES-180383) – via content.iospress.com.
8. "Feature Engineering" (http://www.cs.princeton.edu/courses/archive/spring10/cos424/slides/18-feat.pdf) (PDF). 2010-04-22. Retrieved 12 November 2015.
9. "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.
10. "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.
11. Knobbe, Arno J.; Siebes, Arno; Van Der Wallen, Daniël (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. **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.
12. "A Comparative Study Of Multi-Relational Decision Tree Learning Algorithm". CiteSeerX 10.1.1.636.2932 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.636.2932).

13. Leiva, Hector; Atramentov, Anna; Honavar, Vasant (2002). "Experiments with MRDTL – A Multi-relational Decision Tree Learning Algorithm" (http://web.cs.iastate.edu/~honavar/Papers/kddmrdmpaperfinal.pdf) (PDF).
14. Yin, Xiaoxin; Han, Jiawei; Yang, Jiong; Yu, Philip S. (2004). "CrossMine: Efficient Classification Across Multiple Database Relations". *Proceedings. 20th International Conference on Data Engineering*. *Proceedings of the 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.
15. Frank, Richard; Moser, Flavia; Ester, Martin (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. **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.
16. "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.
17. "Automating big-data analysis" (https://news.mit.edu/2015/automating-big-data-analysis-1016).
18. 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)*. *IEEE International Conference on Data Science and Advanced Analytics*. pp. 1–10. doi:10.1109/DSAA.2015.7344858 (https://doi.org/10.1109%2FDSAA.2015.7344858). ISBN 978-1-4673-8272-4.
19. "Featuretools | An open source framework for automated feature engineering Quick Start" (https://www.featuretools.com/). *www.featuretools.com*. Retrieved 2019-08-22.
20. Hoang Thanh Lam; Thiebaut, Johann-Michael; Sinn, Mathieu; Chen, Bei; Mai, Tiep; Alkan, Oznur (2017). "One button machine for automating feature engineering in relational databases". arXiv:1706.00327 (https://arxiv.org/abs/1706.00327). Bibcode:2017arXiv170600327T (https://ui.adsabs.harvard.edu/abs/2017arXiv170600327T).
21. "ExploreKit: Automatic Feature Generation and Selection" (https://people.eecs.berkeley.edu/~dawnsong/papers/icdm-2016.pdf) (PDF).

Further reading

- Boehmke, Bradley; Greenwell, Brandon (2019). "Feature & Target Engineering". *Hands-On Machine Learning with R*. Chapman & Hall. pp. 41–75. ISBN 978-1-138-49568-5.
- Zheng, Alice; Casari, Amanda (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly. ISBN 978-1-4919-5324-2.
- Zumel, Nina; Mount, John (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=948384435"

This page was last edited on 31 March 2020, at 19:14 (UTC).