Search Data Science Centra Search

- Sign Up
- Sign In

Data Science Central® THE ONLINE RESOURCE FOR BIG DATA PRACTITIONERS

- HOME
- AI
- ML
- DL
- ANALYTICS
- STATISTICS
- BIG DATA
- DATAVIZ
- HADOOP
- PODCASTS
- WEBINARS
- FORUMS
- JOBS
- MEMBERSHIP
- GROUPS
- SEARCH
- CONTACT

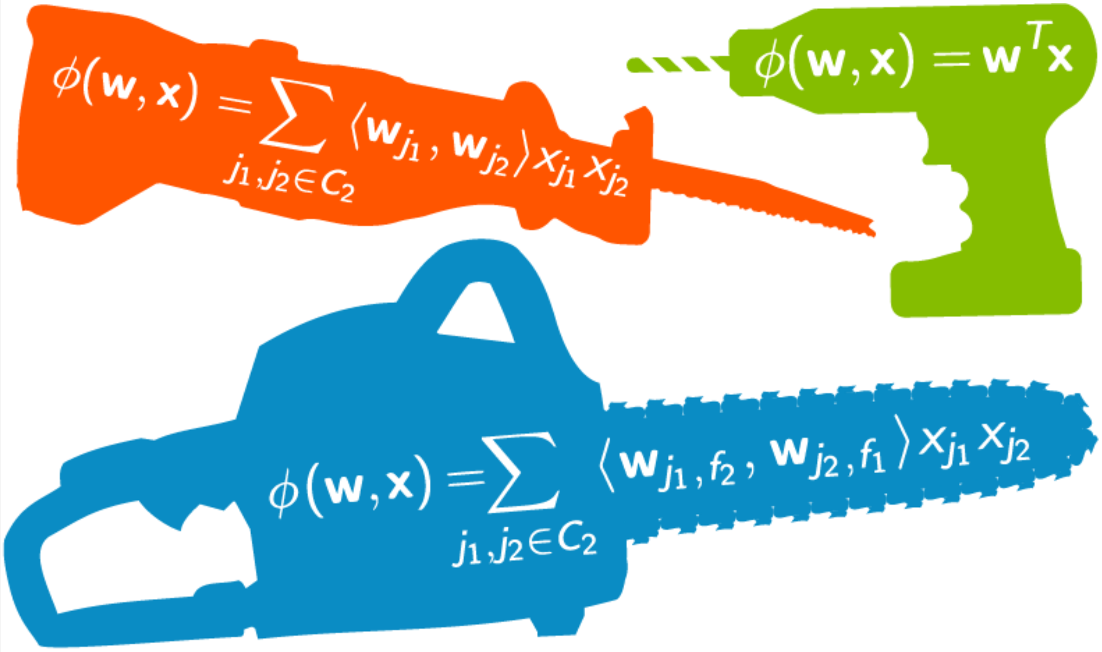
Subscribe to DSC Newsletter

- All Blog Posts
- My Blog
- Add

Feature Engineering: Data scientist's Secret Sauce !

- Posted by Ashish kumar on July 10, 2016 at 7:30am
- View Blog

originally posted by the author on Linkedin : Link


$\phi(\mathbf{w},\mathbf{x}) = \sum_{j_1,j_2 \in C_2} \langle \mathbf{w}_{j_1}, \mathbf{w}_{j_2} \rangle x_{j_1} x_{j_2}$
$\phi(\mathbf{w},\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
$\phi(\mathbf{w},\mathbf{x}) = \sum_{j_1,j_2 \in C_2} \langle \mathbf{w}_{j_1,f_2}, \mathbf{w}_{j_2,f_1} \rangle x_{j_1} x_{j_2}$

It is very tempting for data science practitioners to opt for the best known algorithms for a given problem.*However* It's not the algorithm alone , which can provide the best solution ; Model built on carefully engineered and selected features can provide far better results.

> "Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius -- and a lot of courage -- to move in the opposite direction."- Albert Einstein

The complex models are not easily interpretable and tougher to tune. Simpler algorithms, with better features or more data can perform far better than a weak assumption accompanied with a complex model.

Better features means flexibility, simpler models, better results. Presence of irrelevant features hurt generalization. Thus feature selection and feature engineering should not be considered as mutually exclusive activities and should be performed in conjunction to each other. With the help of an effective feature engineering process, we intend to come up with an effective representation of the data. The question arises, what is considered to be a good or bad representation?

Representation is as good as the information it contains.

1. single variable Basic transformations: x, x^2 ,sqrt x ,log x, scaling

2. If variable's distribution has a long tail, apply Box-Cox transformation (taking log() is a quick & dirty way).

3. One could also perform analysis of residuals or log-odds (for linear model) to check for strong nonlinearities.

4. Create a feature which captures the frequency of the occurrence of each level of the categorical variable. For high cardinality, this helps a lot. One might use ratio/percentage of a particular level to all the levels present.

5. For every possible value of the variable, estimate the mean of the target variable; use the result as an engineered feature.

6. Encode a variable with the ratio of the target variable.

7. Take the two most important variables and throw in second order interactions between them and the rest of the variables - compare the resulting model to the original linear one

8. if you feel your solutions should be smooth, you can apply a radial basis function kernel . This is like applying a smoothing transform.

9. If you feel you need covariates , you can apply a polynomial kernel, or add the covariates explicitly

10. High cardinality features : convert to numeric by preprocessing: out-of-fold average **two variable combinations**

11. Additive transformation

12. difference relative to baseline

13. Multiplicative transformation : interactive effects

14. divisive : scaling/normalisation

15. thresholding numerical features to get boolean values

16. Cartesian Product Transformation

17. Feature crosses: cross product of all features -- Consider a feature A, with two possible values {A1, A2}. Let B be a feature with possibilities {B1, B2}. Then, a feature-cross between A & B (let’s call it AB) would take one of the following values: {(A1, B1), (A1, B2), (A2, B1), (A2, B2)}. You can basically give these ‘combinations’ any names you like. Just remember that every combination denotes a synergy between the information contained by the corresponding values of A and B.

18. Normalization Transformation: -- One of the implicit assumptions often made in machine learning algorithms (and somewhat explicitly in Naive Bayes) is that the the features follow a normal distribution. However, sometimes we may find that the features are not following a normal distribution but a log normal distribution instead. One of the common things to do in this situation is to take the log of the feature values (that exhibit log normal distribution) so that it exhibits a normal distribution.If the algorithm being used is making the implicit/explicit assumption of the features being normally distributed, then such a transformation of a log-normally distributed feature to a normally distributed feature can help improve the performance of that algorithm.

19. Quantile Binning Transformation

20. whitening the data

21. Windowing -- If points are distributed in time axis, previous points in the same window are often very informative

22. Min-max normalization : does not necessarily preserve order

23. sigmoid / tanh / log transformations

24. Handling zeros distinctly – potentially important for Count based features

25. Decorrelate / transform variables

26. Reframe Numerical Quantities

27. Map infrequent categorical variables to a new/separate category.

28.Sequentially apply a list of transforms.

29. One Hot Encoding

30. Target rate encoding

Hash Trick Multivariate:

31. PCA

32. MODEL STACKING

33. compressed sensing

34..guess the average” or “guess the average segmented by variable X”

Projection : new basis

35. Hack projection:

- Perform clustering and use distance between points to the cluster center as a feature
- **PCA/SVD --** Useful technique to analyze the interrelationships between variables and perform dimensionality reduction with minimum loss of information (find the axis through the data with highest variance / repeat with the next orthogonal axis and so on , until you run out of data or dimensions; Each axis acts a new feature)

36.Sparse coding **--** choose basis : evaluate the basis based on how well you can use it to reconstruct the input and how sparse it is take some sort of gradient step to improve that evaluation

Most Popular Content on DSC

To not miss this type of content in the future, subscribe to our newsletter.

- Book: Statistics -- New Foundations, Toolbox, and Machine Learning Recipes
- Book: Classification and Regression In a Weekend - With Python
- Book: Applied Stochastic Processes
- Long-range Correlations in Time Series: Modeling, Testing, Case Study
- How to Automatically Determine the Number of Clusters in your Data
- New Machine Learning Cheat Sheet | Old one
- Confidence Intervals Without Pain - With Resampling
- Advanced Machine Learning with Basic Excel
- New Perspectives on Statistical Distributions and Deep Learning
- Fascinating New Results in the Theory of Randomness
- Fast Combinatorial Feature Selection

Other popular resources

- Comprehensive Repository of Data Science and ML Resources
- Statistical Concepts Explained in Simple English
- Machine Learning Concepts Explained in One Picture
- 100 Data Science Interview Questions and Answers
- Cheat Sheets | Curated Articles | Search | Jobs | Courses
- Post a Blog | Forum Questions | Books | Salaries | News

Archives: 2008-2014 | 2015-2016 | 2017-2019 | Book 1 | Book 2 | More

Follow us: Twitter | Facebook

Views: 18416

Like
20 members like this

Share Tweet Facebook

Like 45

- < Previous Post
- Next Post >

Comment

You need to be a member of Data Science Central to add comments!

Join Data Science Central

Comment by Harsh Pandya on June 25, 2018 at 8:08am

Amazing article. I have always wondered the best practices and ways for feature engineering. I was wondering if someone can point me to more resources on model stacking.

Comment by Pradeep Rao on July 17, 2016 at 2:46am

Thank you for this article. Each of the point you have mentioned is a topic in itself. A condensed view of all of feature engineering is an excellent reference material. Very useful article

Suggestion 5, specifically, sounds like "data leakage", or in other words, a self-fulfilling attribute since it's defined by the target variable...

Comment by Ammar Mohemmed on July 14, 2016 at 12:42pm

Academic research has been investigating feature selection and construction for long long time now and many phd theses were produced in this topic. It seems the commercial data scientists are not following what is been published !

Comment by Sander Stepanov on July 14, 2016 at 12:32pm

ery good

RSS

Welcome to
Data Science Central

Sign Up
or Sign In

Or sign in with:

RESOURCES

- Subscribe to DSC Newsletter
- Free Books
- Forum Discussions
- Cheat Sheets
- Jobs
- Search DSC
- DSC on Twitter
- DSC on Facebook

VIDEOS

Data Science Central | Webinar Series
From Pandas to Apache Spark™
SPONSORED BY
databricks
May 30, 2019
HOST: STEPHANIE GLEN - DSC

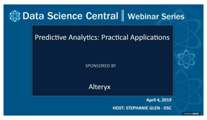
DSC Webinar Series: From Pandas to Apache Spark™

Added by Tim Matteson 0 Comments 1 Like

Data Science Central | Webinar Series
Making AI Work in the Real World:
How Real Companies Get Real Value with AI
SPONSORED BY
figure eight
March 12, 2019
HOST: RAFAEL KNUTH - DSC

DSC Webinar Series: Making AI Work in the Real World: How Real Companies Get Real Value with AI

Added by Tim Matteson 0 Comments 0 Likes


Data Science Central | Webinar Series
Predictive Analytics: Practical Applications
SPONSORED BY
Alteryx
April 4, 2019
HOST: STEPHANIE GLEN - DSC

DSC Webinar Series: Predictive Analytics: Practical Applications

Added by Tim Matteson 0 Comments 0 Likes