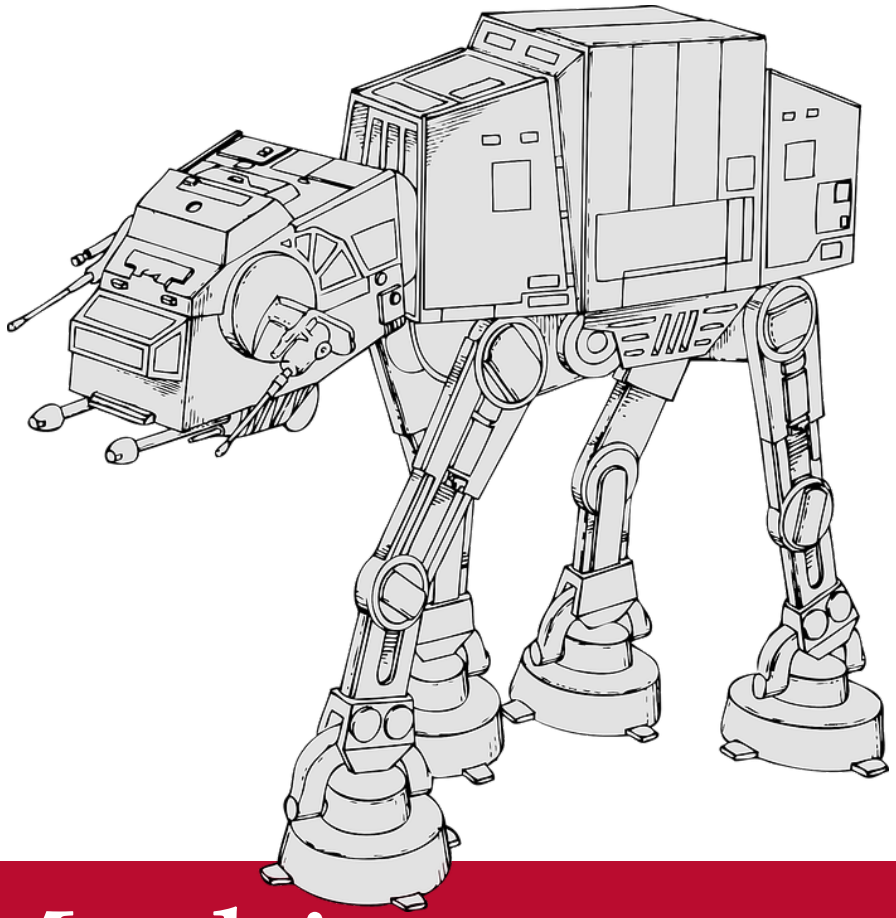
A Machine Learning Study Guide



Machine Learning Handbook

The Definitive Guide

ICS5110, class of 2018/9

L-Università ta' Malta

JEAN-PAUL EBEJER, DYLAN SEYCHELL, FRANCO CASSAR MANGHI

First printing, January 2019

Contents

Introduction

This book explains popular Machine Learning terms. We focus to explain each term comprehensively, through the use of examples and diagrams. The description of each term is written by a student sitting in for ICS5110 APPLIED MACHINE LEARNING[1] at the University of Malta (class 2018/2019). This study-unit is part of the MSc. in AI offered by the Department of Artificial Intelligence, Faculty of ICT.

[1] https://www.um.edu.mt/courses/studyunit/ICS5110

Synthetic Features

The problem

Training a machine learning algorithm requires inputting some form of training data. This training data comprises of all the features from which the algorithm learns from and builds a model. This input is often referred to as the training dataset.

Whilst in concept the above seems straightforward, it often transpires that the various data-points provided in the training dataset do not fit a structure that is easily understood by the algorithm. For this reason, an important pre-processing step is needed to:

1. Understand the original data well
2. Subsequently, if and where needed, generate synthetic features

If we look at the following example (Alberto et al., 2015): It contains a number of records used to train a spam / ham classifier for comments on a YouTube video.

Video	*Comment ID*	*Author*	*Date*	*Content*	*Class*
Psy	*LZQPQhLyRh9MSZYnf8djyk0gEF9BHDPYrrK-qCczIY8*	*Evgeny Murashkin*	*2013-11-08T17:34:21*	*just for test I have to say murdev.com*	*Spam*
Psy	*z13bgdvyluihfo11i22rgxxvhuvabzz1os04*	*Zielimeek21*	*2013-11-28T21:49:00*	*I'm only checking the views*	*Ham*
Psy	*z13kvpqqssa0hlryd04cc1dxeyynglljngk*	*Tasha Lucius*	*2014-01-19T13:25:56*	*2 billion....Coming soon*	*Ham*
Psy	*z12lg1vizrmsgxm3q23oij4aqrjxjdd1p*	*Holly*	*2014-11-06T13:41:30*	*Follow me on Twitter @mscalifornia95*	*Spam*

Table 1: Sample of four rows from the Psy dataset from the YouTube comment training dataset.

Table 2 describes each feature in the original unmodified dataset.

As one can see, there is very little input the machine learning algorithm can reliably take just from using the four features described above. One could easily realise this by asking oneself the following question (in plain English):

How can I describe the components of a comment well enough to decide whether it is probably spam or ham?

One can therefore summarise this problem paradoxically as: *Having enough data to solve the problem, but very little meta-data to actually understand it and solve it.*

Ways of solving the problem

A way of solving this problem is to apply a synthetic features approach, sometimes referred to as feature engineering. This is the generation of features derived from other existing features, in a way

Feature	*Description*
Video	*The video this comment was written for. The relevance depends whether the classification model is being built generically for all videos, or a per-video specific model is also considered.*
Comment ID	*Random comment ID generated by the YouTube comment board system. This probably has no impact on the final class.*
Author	*The author / account that generated the comment. This has relevance only if this account has a lot of spam comments.* *If that is the case, two things should happen, none of which are directly related to the machine learning algorithm:* *- Maintain a blacklist of accounts that are probable spam (if a particular author often has flagged comments).* *- Block such accounts.*
Date	*The date does not directly have a huge relevance on the classification of a comment.*
Content	*The comment body definitely has a big relevance in the classification result, however, can the whole sentence be easily understood by the algorithm as it is?*

Table 2: Description of each feature in the original unmodified YouTube comment dataset.

that can be more easily captured or understood by a machine learning algorithm (Li et al., 2013). It is a way of generating meta-data for the existing features in the original dataset.

In essence, the idea is to look at every available feature and for each determine the following:

- Does the feature contain more than one feature within it? If so, try exploding it into sub-features and test.
- Does the feature contain too little information for any relevance, but could benefit from adding some context to it? If so, attempt at looking at other features that might be related, and produce new features as a result, and test.
- For each of the above, the original feature(s) might not be relevant anymore and be entirely replaced by the newly generated synthetic features instead.

What or how an explosion of features or a composite of features is generated depends on the very specific nature of the components involved and there is no generic formula behind it that works without some additional specificness. For example, two pairs of geo-coordinates probably qualify in giving a distance feature, however the formula applied here is specific to the geo-coordinates domain.

There is not a one-size-fits-all approach but rather it is more of an iterative approach with new synthetic features being outputted per iteration, following which one then assesses whether it is enough to generate a reliable machine learning model from the new features or not.

Following below is a practical example of this technique, using the dataset described at the introduction of this chapter.

Analysing each feature

- Video

  - The video name / ID could be useful if a per-video classifier is also generated over and above the generic one. This together with other features could have some relevance.

- Comment ID
  - This feature does not have any relevance to the outcome whatsoever. It is a unique ID, built randomly, assigned to each comment. For this reason, it is out of scope for this discussion.

- Author and Date
  - As described earlier these two features independently do not have much of a direct impact on the outcome, however a synthetic feature could be generated which might have some form of effect on the outcome: A ratio of comment count over a time period for a particular author. The idea is to make it easier for the algorithm to detect a potential pattern related to volume over a typical short period, thus the definition of time period can be assigned via testing.

- Comment
  - The comment body is not an easy feature and it could grow into a number of features, however it is the most relevant input for this spam classifier. Quite a number of features could be exported from this comment, and most of them relate to natural language processing techniques (Cormack et al., 2007). For this reason, output quality could also vary based on the language in context. Some example features that could be extrapolated:

Synthetic Feature	*Scope / Description*
Language	*This depends on the availabilities of various NLP implementations for different languages, however one could have an indication of spam / non-spam probabilities based on the comment languages for each particular video.*
Readability score	*A readability score could be calculated per comment which gives an indication on the quality of such text. An example of such a score could be the* Flesch Reading Ease *score.*
Length (excl. stop words)	*Very short or very long comments might have a probabilistic impact on the outcome.*
Presence of account tags / URLs / emojis	*The presence of account tags (ex. a Twitter username), URLs or emojis could increase probability of the comment being spam.*

Table 3: Example of possible features that can be extracted from textual comments.

Updated feature / data set

Following the synthetic feature generation described above, the updated data set used as an example here would look as follows:

Looking at the output in table 4, the effect of synthetic features can immediately be appreciated, as with such new features more meaning is given to the original dataset.

Video	*Author Comments in last minute*	*Language*	*Readability*	*Length excl. stop words*	*Presence of account tags*	*Presence of URLs*	*Presence of emojis*	*Class*
Psy	*1*	*EN*	*94.3*	*3*	*No*	*Yes*	*No*	*Spam*
Psy	*1*	*EN*	*103*	*3*	*No*	*No*	*No*	*Ham*
Psy	*1*	*EN*	*83.3*	*3*	*No*	*No*	*No*	*Ham*
Psy	*1*	*EN*	*32.6*	*3*	*Yes*	*No*	*No*	*Spam*

Table 4: Updated sample of the four rows from the Psy dataset from the YouTube comment training dataset now containg the synthetic features.

Naturally the above contains just a sample, and one must experiment with:

- more or less synthetic features
- a further iteration of synthetic features from the generated features
- a much bigger data-set (the example above is too small to build a reliable classifier)
- perform feature selection (such as Principle Component Analysis) to identify the features that actually matter and remove extra noise

Therefore, employing a synthetic feature approach on your dataset as a pre-processing step, should in general give you positive results.

Bibliography

T C Alberto, J V Lochter, and T A Almeida. TubeSpam: Comment Spam Filtering on YouTube. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 138–143, dec 2015.

Gordon V Cormack, José María Gómez Hidalgo, and Enrique Puertas Sánz. Feature Engineering for Mobile (SMS) Spam Filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 871–872. ACM, 2007. ISBN 978-1-59593-597-7.

Jiefei Li, Xiaocong Liang, Weijie Ding, Weidong Yang, and Rong Pan. Feature Engineering and Tree Modeling for Author-paper Identification Challenge. In *Proceedings of the 2013 KDD Cup 2013 Workshop*, KDD Cup '13, pages 5:1—-5:8, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2495-3.

Index