Cyclo-Analysis: Data Mining and Analysis in an Online Cycling Community

Allan Whatmough
Birkbeck, University of London
MSc Data Analytics

April 2015

1 Introduction

Internet forums contain a large amount of potentially interesting information available for data mining and analysis. This offers opportunities for gaining insight into how online communities survive, and often, thrive.

In this study, I will base my research and experiments on the forum used by my local cycling club, Islington CC. *forum.islington.cc* is powered by Microcosm[1], an open-source forum and community CMS software system. The concepts expressed should be equally applicable to any type of forum, however.

2 Problem Description

Internet forums are one of the oldest forms of online community, and the software used to power them has remained largely unchanged in recent years. As such, the typical forum lacks features which make use of modern data analysis and mining techniques. Such techniques might include sentiment analysis and topic modelling, and can be used to provide forum users with useful information about the type of discussions which are taking place on the forum.

As online communities grow, the problem of finding useful and relevant information from a forum becomes even more pronounced. The difficulty of scaling online communities has been written about extensively. For example, in Communities, Audiences, and Scale[2], Clay Shirky states that "The inability of a single engaged community to grow past a certain size, irrespective of the technology, will mean that over time, barriers to community scale will cause a separation between media outlets that embrace the community model and stay small, and those that adopt the publishing model in order to accommodate growth." Similarly, in A Group Is Its Own Worst Enemy[3], Shirky advises that

[1] http://microcosm.cc
[2] http://shirky.com/writings/community_scale.html
[3] http://www.shirky.com/writings/herecomeseverybody/group_enemy.html

"human interaction, many to many interaction, doesn't blow up like a balloon. It either dissipates, or turns into broadcast, or collapses. So plan for dealing with scale in advance, because it's going to happen anyway."

3 Objectives

I intend to develop tools and produce visualisations to analyse the data within a forum. This will provide users and administrators with information about the discussions taking place within the forum. Such tools and visualisations should prove particularly useful when a community reaches a certain size, at which point the volume of conversation is so great that manual categorisation and analysis of the data becomes difficult.

Drawing on topics from cloud computing, information retrieval, relational/non-relational database technology, and the Semantic Web, I will develop a system which can perform the following tasks:

- Classification of forum posts into top-level categories, based on which subsection within the forum the post is most likely to belong to.
- Further classification of forum posts into more fine-grained topics, using topic modelling. For example, it should be possible to identify posts which are 'classified ads', that is, posts which are advertising items for sale.
- Similar to the above, but to address a common issue faced by moderators of forums, posts can automatically be categorised into spam/not spam categories, as described in [1].
- Perform sentiment analysis of forum posts. For example, individual posts (or individual sentences within posts) can be described as positive or negative based on their content.
- Perform named entity recognition on forum posts in order to extract information about entities, such as identifying particular bike manufacturers.
- If time permits, using information gained from named entity recognition, it will be possible to extend a given forum system to produce additional markup which describes the structure of a post, using Schema.org[4], for example. Schema.org can be used to structure the data and identify people, places, and things, amongst other types of data.
- Provide information about individual users based on the topics they tend to discuss. Each user can be represented by a histogram of topics, and it will be possible to visualise the most frequently discussed topics for each user.

[4] http://schema.org/

- Group similar users together based on the topics they tend to discuss. These users can be clustered and visualisations will then be produced to illustrate user similarity.
- If time permits, tagging functionality will be added to the forum software, allowing users to manually tag forum posts themselves. This could be used as a way of improving the classification accuracy within the system, since the tags can be used to train my classifier. In addition, the tags can be used for evaluation purposes, to compare how my classifiers perform against data which has been labelled by real users.

4 Existing Forum Software

There are a number of existing forum software packages available, including Discourse, Simple Machines, Vanilla, and vBulletin. None of these provide any support for analysis of the forum data based on machine learning or data analysis techniques, however.

I have chosen to base my study on the Microcosm forum software for a variety of reasons. Unlike the forum software mentioned above, it is developed using a client-server architecture. This makes it more straightforward to modify and extend the code, sine there is a greater separation of concerns. In addition, the Microcosm server provides a rich API, which makes it simple to extract forum data. This is an important task and must be done prior to performing any analysis of the data.

Finally, it should be noted that Microcosm has been described as a "community CMS", since it has features required by a community, including support for posting discussion topics in addition to other types of content, such as events. In this study, I will only consider discussion topics and their associated comments and replies, however.

5 Background Research/Proposed Methodology

Supervised vs Unsupervised Learning

Two major types of machine learning are supervised and unsupervised learning. In this study, I will consider both types. Some of the forum data is in a sense already labelled, depending on the sub-forum a given post is contained in. I can use this label to perform supervised learning, for example, by using a naive Bayes classifier. In addition, unsupervised learning algorithms such as kNN (k-Nearest Neighbours) and LDA (Latent Dirichlet allocation) can be used to cluster forum posts and make predictions about a post's topics, respectively.

Classification

There are a number of well-researched classification techniques that I will make use of in this project. One of the simplest approaches is to use a naive Bayes classifier. This assumes that the terms within the forum posts (which can also be referred to as 'documents') are conditionally independent of each other. In practice, this is not true, hence why this type of classification is referred to as 'naive', but the classifier may still perform well on the data. As pointed out by [3], the conditional independence assumption is equivalent to the 'bag of words' model of text classification, which states that the order of the words within a document is irrelevant. What is relevant, however, is the frequency of each word in the document [6]. More specifically, TF-IDF is a useful measure of how important a term is within a corpus [2, 3]. This weights the terms according to how frequently they occur within individual documents, but also takes into account how frequently the term occurs within the entire corpus. TF-IDF is highest when a given term occurs many times within a small number of documents [3]. It has been argued that the assumptions made in naive Bayes classification are excessively naive, and there are ways in which these issues can be addressed [4]. In particular, the multinomial naive Bayes model does not accurately reflect the frequency distribution of terms in real documents. Instead, the distribution of terms within the document collection more closely resembles a power law distribution. This concept is related to *Zipf's law* [2, 3], which states that the most frequent term appears in the collection roughly twice as often as the second most frequent term, and so on. If time permits, I will attempt to improve the performance of the text classifiers by applying some of the ideas proposed in [4], such as performing length normalisation of the documents and applying TF-IDF transforms.

In the Islington Cycling Club forum, posts are already categorised into one of four categories, depending on which of the four sub-forums the post belongs to. A naive Bayes classifier can then be trained on this data, and it will then be possible to automatically assign future posts to the most appropriate category/sub-forum based on the content of the post.

A popular Python software package for machine learning is scikit-learn[5]. This package includes various tools which will prove useful for performing data analysis, including implementations of many popular algorithms (including naive Bayes).

Topic Modelling

There are a number of software packages available which can be used to perform topic modelling. A popular one which I intend to use in this project is Gensim[6], which includes support for models such as LSA (Latent Semantic Analysis) and LDA. Using these models, it is possible to automatically assign forum posts to topics and compute the similarity between different posts.

[5] http://scikit-learn.org/
[6] https://radimrehurek.com/gensim/

Sentiment Analysis

It is possible to split forum posts into sentences, and then perform sentiment analysis on the individual sentences. The Python library TextBlob[7] will be helpful for this task. It should be noted that sentiments for a particular topic are likely to be a mixture of different sentiments, both positive and negative, and include varying degrees of each. It might therefore be necessary to aggregate the sentiments across a particular topic in order to summarise this information.

Visualisation

To make it easier to interpret the data, visualisations will be generated. There are a variety of open-source tools which can be used to produce attractive and engaging visualisations. For example, D3.js[8] is a JavaScript library which can be used to develop interactive visualisations, and Seaborn[9] is a Python library based on matplotlib.

A simple example of a visualisation which can be generated is a tag cloud [2]. This illustrates the relative frequency of terms which are in use on the forum, and is an easy way to see which topics of discussion are most popular.

Data Pre-processing and Stemming

Microcosm provides an API which makes it easy to access the posts contained within the forum.

Before analysing the data, it will be necessary to perform some pre-processing. Since the forum posts are stored as Markdown[10], this might include removing links and references to images. Following the pre-processing and tokenisation steps, by which unnecessary information is removed and each post is split into its individual tokens, stemming should be performed. Stemming reduces inflectional forms of a word to a common base form [3]. As pointed out by [5], this makes sense since words with the same base form refer to the same concept.

There are a number of open-source tools available which can be used to stem words. For example, the Python NLTK[11] package can be used in the following way:

```
>>> from nltk.stem import SnowballStemmer
>>> stemmer = SnowballStemmer('english')
>>> print stemmer.stem('cycling')
cycl
>>> print stemmer.stem('cycle')
cycl
```

[7] http://textblob.readthedocs.org/
[8] http://d3js.org/
[9] http://stanford.edu/~mwaskom/software/seaborn/
[10] http://daringfireball.net/projects/markdown/
[11] http://www.nltk.org/

Other terms are stemmed in similar ways, for example: 'pedalling' → 'pedal', 'braking' → 'brake', and 'unclipping' → 'unclip'.

In the above example, the Snowball[12] stemmer is used. There are many other types of stemming algorithm available, however.

Finally, there are tools available for cleaning and normalising text which might prove useful in this project. One such tool is Annotate.io[13] which can learn from training data examples and produce cleaned versions of unseen text. This could be used to normalising the various different forms of title which people use when posting classified adverts. For example, 'FS: Giant bike' and 'For sale: Giant bike' are equivalent, and should be classified in the same way.

6 Project Plan

In order to ensure that the project is completed successfully, it will be necessary to break the project down into various smaller tasks. These are summarised below, along with planned timescales:

- April - May 2015: Further research into the problem, including experimental work with the algorithms provided by libraries such as scikit-learn on the data.
- April - June: Implementation of a test version of the Microcosm system for development purposes. Real data from the Islington CC forum will be loaded into the system.
- June - August: Development of functionality as described in the project objectives. This will be an interactive process, involving testing and evaluation of the techniques used. Functionality which meets the requirements will be integrated into the test version of the forum. During these months, the final project report will be written.
- August - September: Detailed evaluation of the project. This might involve using tools such as Palmetto[14] to evaluate the quality of the topics discovered using topic modelling, for example. In the first week of September, any final changes to the project report will be made.

References

[1] Paul Graham. A plan for spam. 2002.

[2] Mark Levene. *An Introduction to Search Engines and Web Navigation*. Wiley, 2010.

[12] http://snowball.tartarus.org/texts/introduction.html
[13] http://annotate.io/
[14] http://aksw.org/Projects/Palmetto.html

[3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. 2008.

[4] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. 2003.

[5] Willi Richert and Luis Pedro Coelho. *Building Machine Learning Systems with Python*. 2013.

[6] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2003.