Thoughtland: Natural Language Descriptions for Machine Learning n-dimensional Error Functions

Pablo Ariel Duboue
Les Laboratoires Foulab
999 du College
Montreal, Quebéc
pablo.duboue@gmail.com

Abstract

This demo showcases Thoughtland, an end-to-end system that takes training data and a selected machine learning model, produces a cloud of points via cross-validation to approximate its error function, then uses model-based clustering to identify interesting components of the error function and natural language generation to produce an English text summarizing the error function.

1 Introduction

For Machine Learning practitioners of supervised classification, the task of debugging and improving their classifiers involves repeated iterations of training with different parameters. Usually, at each stage the trained model is kept as an opaque construct of which only aggregate statistics (precision, recall, etc.) are investigated. Thoughtland (Duboue, 2013) improves this scenario by generating Natural Language descriptions for the error function of trained machine learning models. It is a pipeline with four components:

(1) A cross-validation step that uses a machine algorithm from a given learning library run over a given dataset with a given set of parameters. This component produces a cloud of points in n-dimensions, where $n = F + 1$, where F is the number of features in the training data (the extra dimension is the error value). (2) A clustering step that identifies components within the cloud of points. (3) An analysis step that compares each of the components among themselves and to the whole cloud of points. (4) A verbalization step that describes the error function by means of the different relations identified in the analysis step.

2 Structure of the Demo

This demo encompasses a number of training datasets obtained from the UCI Machine Learning repository (attendees can select different training parameters and see together the changes in the text description). It might be possible to work with some datasets provided by the attendee at demo time, if they do not take too long to train and they have it available in the regular Weka ARFF format.

A Web demo where people can submit ARFF files (of up to a certain size) and get the different text descriptions is will also be available at **http://thoughtland.duboue.net** (Fig. 1). Moreover, the project is Free Software[1] and people can install it and share their experiences on the Website and at the demo booth.

3 An Example

I took a small data set from the UCI Machine Learning repository, the Auto-Mpg Data[2] and train on it using Weka (Witten and Frank, 2000). Applying a multi-layer perceptron with two hidden layers with three and two units, respectively, we achieve an accuracy of 65% and the following description:

> There are four components and eight dimensions. Components One, Two and Three are small. Components One, Two and Three are very dense. *Components Four, Three and One are all far from each other.* The rest are all at a good distance from each other.

When using a single hidden layer with eight units we obtain an accuracy 65.7%:

[1] https://github.com/DrDub/Thoughtland.

[2] http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/

Proceedings of the 14th European Workshop on Natural Language Generation, pages 200–201,
Sofia, Bulgaria, August 8-9 2013.

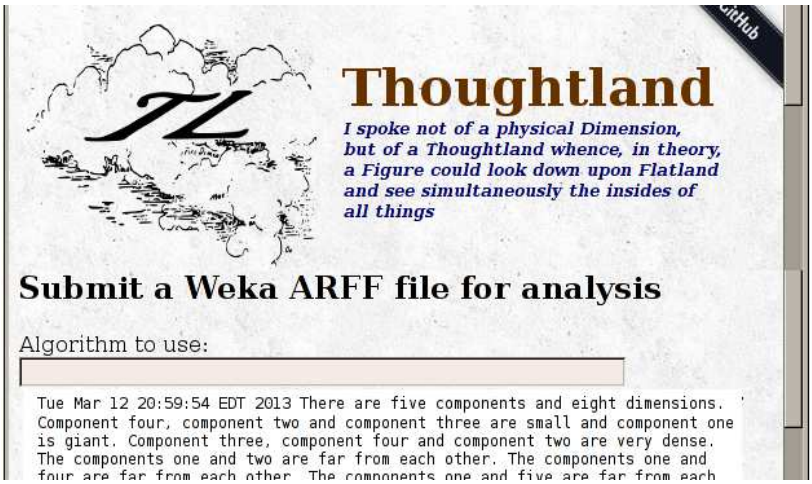

Figure 1: Web Interface to Thoughtland (composite).

> There are four components and eight dimensions. Components One, Two and Three are small. Components One, Two and Three are very dense. *Components Four and Three are far from each other.* The rest are all at a good distance from each other.

As both descriptions are very similar (we have emphasized the difference, which in the first case is also an example of our clique-based aggregation system), we can conclude that the two systems are performing quite similarly. However, if we use a single layer with only two units, the accuracy lowers to 58% and the description becomes:

> There are five components and eight dimensions. Components One, Two and Three are small and Component Four is giant. Components One, Two and Three are very dense. Components One and Four are at a good distance from each other. Components Two and Three are also at a good distance from each other. Components Two and Five are also at a good distance from each other. The rest are all far from each other.

4 Final Remarks

Thoughtland follows the example of Mathematics, where understanding high dimensional objects is an everyday activity, thanks to a mixture of formulae and highly technical language. It's long term goal is to mimic these types of descriptions automatically for the error function of trained machine learning models.

The problem of describing n-dimensional objects is a fascinating topic which Throughtland just starts to address. It follows naturally the long term interest in NLG for describing 3D scenes (Blocher et al., 1992).

Thoughtland is Free Software, distributed under the terms of the GPLv3+ and it is written in Scala, which allow for easy extension in both Java and Scala and direct access to the many machine learning libraries programmed in Java. It contains a straightforward, easy to understand and modify classic NLG pipeline based on well understood technology like McKeown's (1985) schemata and Gatt and Reiter's (2009) SimpleNLG project. This pipeline presents a non-trivial NLG application that is easy to improve upon and can be used directly in classroom presentations.

References

A. Blocher, E. Stopp, and T. Weis. 1992. ANTLIMA-1: Ein System zur Generierung von Bildvorstellungen ausgehend von Propositionen. Technical Report 50, University of Saarbrücken, Informatik.

P.A. Duboue. 2013. On the feasibility of automatically describing n-dimensional objects. In *ENLG'13*.

A. Gatt and E. Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proc. ENLG'09*.

K.R. McKeown. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

Ian H. Witten and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers.