ML4PG: Machine learning for Proof General

Jónathan Heras and Ekaterina Komendantskaya
{jonathanheras,katya}@computing.dundee.ac.uk

May 31, 2013

Contents

1 Using ML4PG

To illustrate the use of ML4PG, we will use the file `ml4pg.v` which can be find in the same folder of this manual. This file contains various lemmas about natural numbers and lists.

1.1 Getting started

Open the file `ml4pg.v` using emacs. The Proof General interface is the usual one, but it includes a new option in the Coq menu called ML4PG.

If you select this option, the interface asks you if you are developing your proofs using the plain Coq style or the SSReflect style, in this case we select the Coq mode (c). Subsequently, the interface asks you if you want to extract the information associated with the lemmas which have been previously developed in this library. In this case, we select no (c). Once that this is done, the Proof General interface is extended with a new menu called *Statistics* and two buttons, see Figure 2.

1.2 Extracting feature vectors

Feature vectors can be extracted in two different ways:

- During the development of the proofs. To this aim, you have to use the shortcut Ctrl-C Ctrl-M to process the next proof command.
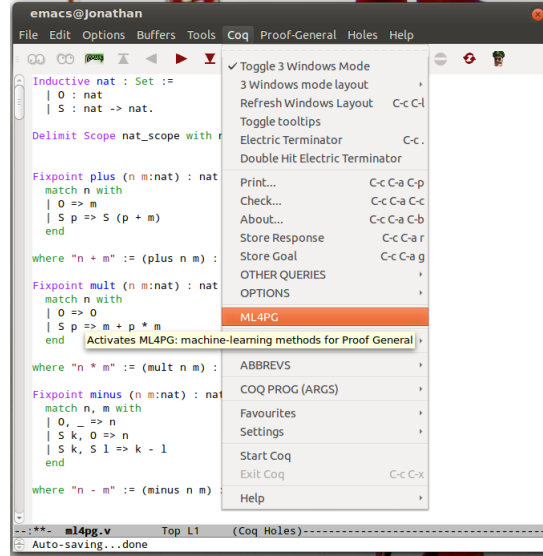
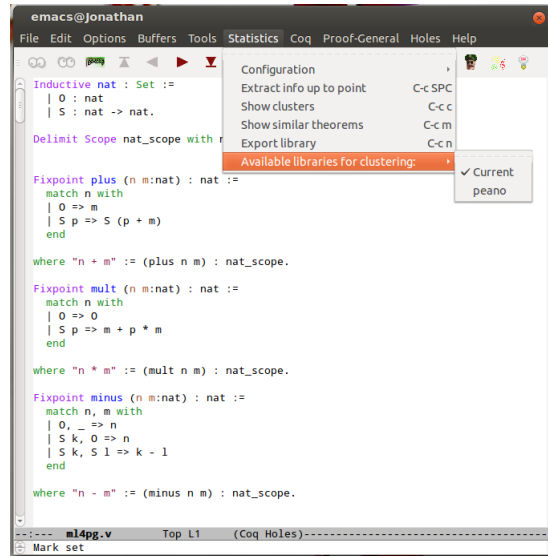Figure 1: Proof General with the ML4PG option.

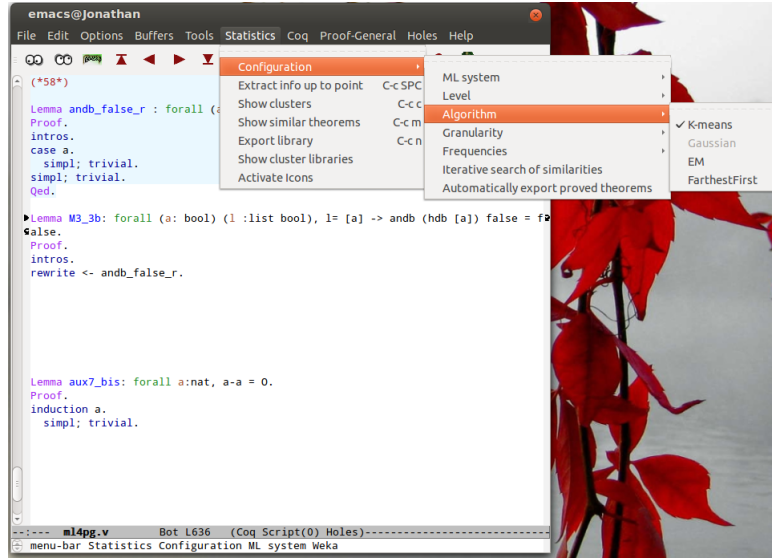Figure 2: ML4PG interface with all the options active.

Figure 3: The ML algorithms menu.

- Several proofs at the same time. If you want to extract the feature vectors of several proofs, go to the last proof and use the shortcut Ctrl-C Space. You can also use the *Extract info up to point* option of the statistics menu.

Go to the end of `emacs ml4pg.v` file; there, you can see two unfinished proofs: `M3_3b` and `aux7_bis`. Put the cursor at the end of the proof of Lemma `andb_false_r` and use the shortcut Ctrl-C Space or the *Extract info up to point* option of the statistics menu. In this way the information associated with each proof will be extracted and you will be able to use it to obtain proof clusters (groups of similar proofs).

Now, let us explain the functionality of the options included in the Statistics menu.

1.3 Configuration menu

The different options to configure the Machine-learning environments were detailed in [2]. All those options can be accessed from the Configuration submenu of the Statistics menu, see Figure 2.

Algorithms: The user can select different algorithms to obtain proof similarities (all of them behave similar, see [1]). ML4PG offers different algorithms, see Figure 3.

In the case of MATLAB; there are three algorithms available: K-means and Gaussian. In the case of Weka, the algorithms which are available are: K-means, EM and FarthestFirst.
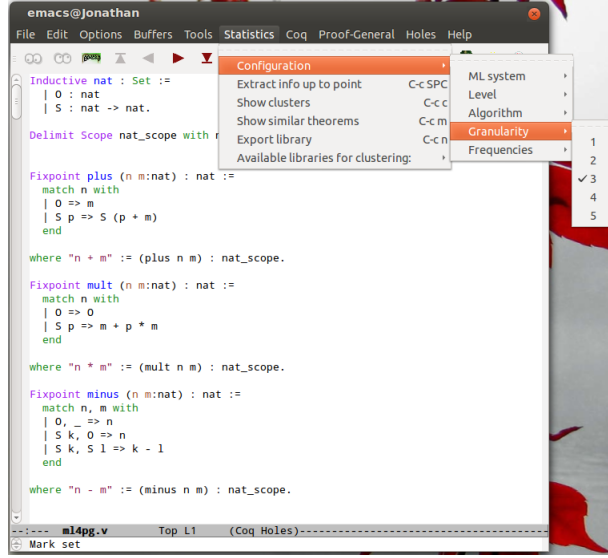
Figure 4: ML4PG granularity menu.

Granularity: In the machine learning literature, there exists a number of heuristics to determine this optimal number of clusters. We used them as an inspiration to formulate our own algorithm for ML4PG, tailored to the interactive proofs. It takes into consideration the size of the proof library and an auxiliary parameter – called granularity. This parameter is used to calculate the optimal number of proof clusters, the process to calculate this optimal number was described in [2]. The user decides the granularity in ML4PG menu (see Figure 4), by selecting a value between 1 and 5, where 1 stands for a low granularity (producing big and general clusters) and 5 stands for a high granularity (producing small and precise clusters).

Frequencies: Clustering techniques divide data into n groups of similar objects (called clusters), where the value of n is a "learning" parameter provided by the user together with other inputs to the clustering algorithms. Increasing the value of n means that the algorithm will try to separate objects into more classes, and, as a consequence, each cluster will contain examples with higher correlation. The frequencies of clusters can serve for analysis of their reliability. Results of one run of a clustering algorithm may differ from another, even on the same data set. This is due to the fact that clustering algorithms randomly choose examples to start from, and then, form clusters relative to those examples. However, it may happen that certain clusters are found repeatedly – and frequently – in different runs; then, we can use these frequencies to determine the reliable clusters. The frequencies can be determined using the threshold presented in Figure 5, a detailed description of this parameter was given in [2].
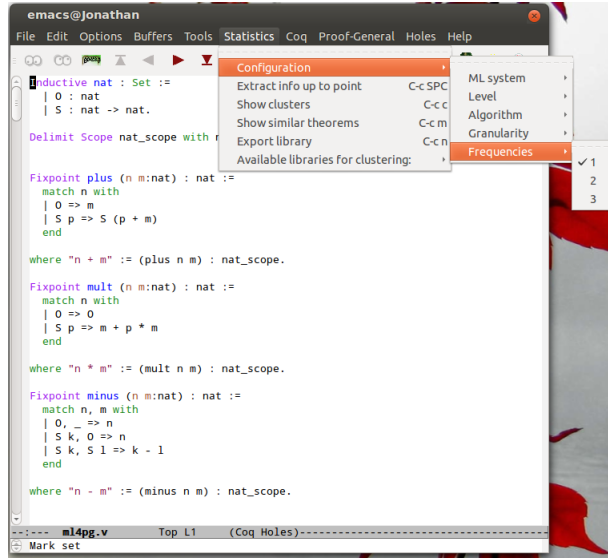
4

Figure 5: ML4PG frequencies menu.

1.4 Show clusters

The option *Show Clusters* of the Statistics menu shows clusters when a library is clustered irrespective of the current proof goal. An example using the `ml4pg.v` library with the options:

- Algorithm: K-means,
- Granularity: 3,
- Frequencies: 1.

is shown in Figure 6.

This functionality can also invoked using the second right most button of the Proof General toolbar.

1.5 Show similar theorems

The example above shows one mode of working with ML4PG: that is, when a library is clustered irrespective of the current proof goal. However, it may be useful to use this technology to aid the interactive proof development. In which case, we can cluster libraries relative to a few initial proof steps for the current proof goal. An example using the `ml4pg.v` library with the options:

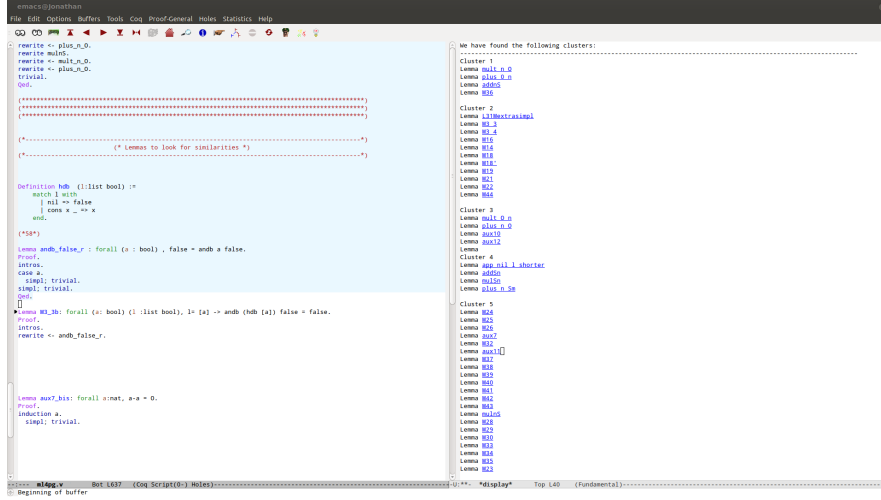- Algorithm: FarthestFirst,
- Granularity: 2,

Figure 6: Clusters for the ml4pg library. The Proof General window has been split into two windows positioned side by side: the left one keeps the current proof script, and the right one shows the clusters. If the user clicks on the name of a theorem showed in the right screen, such a window is split horizontally and a brief description of the selected theorem is shown.

- Frequencies: 2.

and with the few steps included about the proof of M3_3b is shown in Figure 7.

This functionality can also invoked using the right most button of the Proof General toolbar.

1.6   Export Library

Using the Export library option, the user can export the library for further use (see Figure 8) with the Available libraries for clustering option.

References

[1] J. Heras and E. Komendantskaya. Ml4pg case studies. 2013.

[2] E. Komendantskaya, J. Heras, and G. Grov. Machine learning in proof general: interfacing interfaces. 2012.
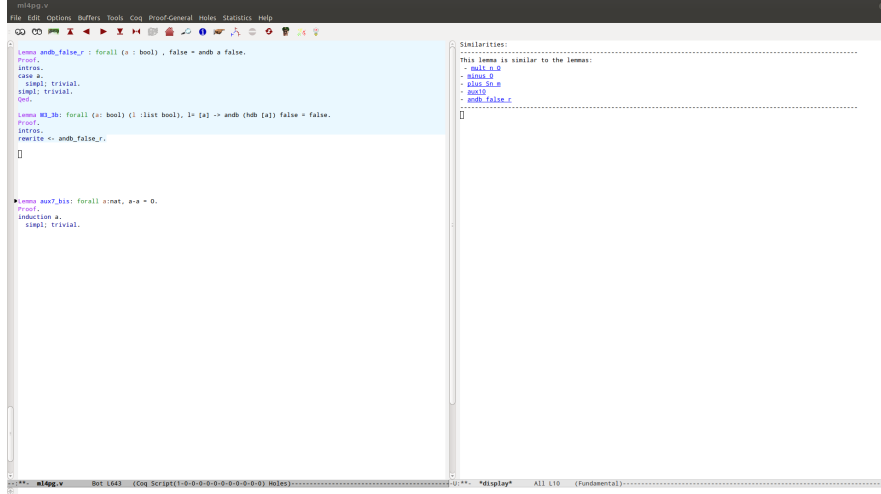
6

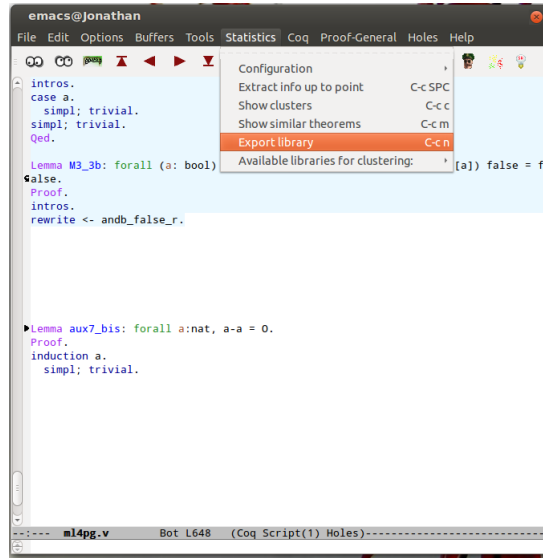Figure 7: On the right side, several suggestions provided by ML4PG. If the user clicks on the name of one of the suggested lemmas, a brief description about it is shown.

Figure 8: ML4PG export menu.