ML4PG: Machine learning for Proof General

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

May 30, 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

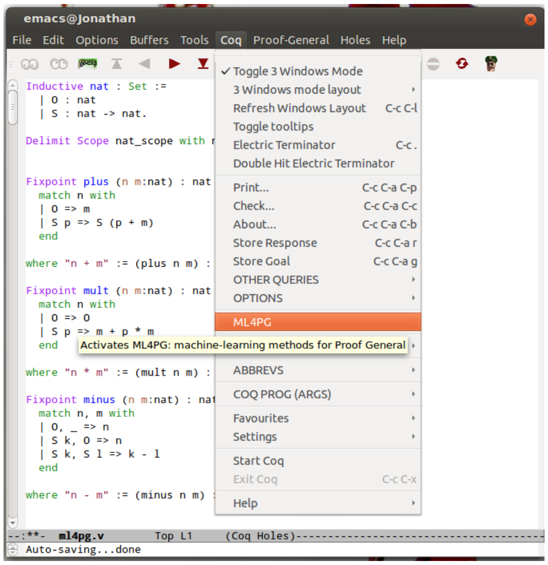
Open the file `ml4pg.v` using emacs. The Proof General interface is the usual one, but it includes a new option in the Coq menu called ML4PG.

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

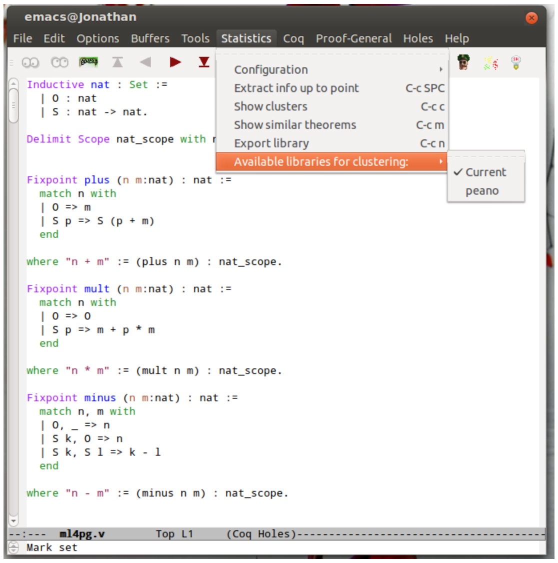
Feature vectors can be extracted in two different ways:

- 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.



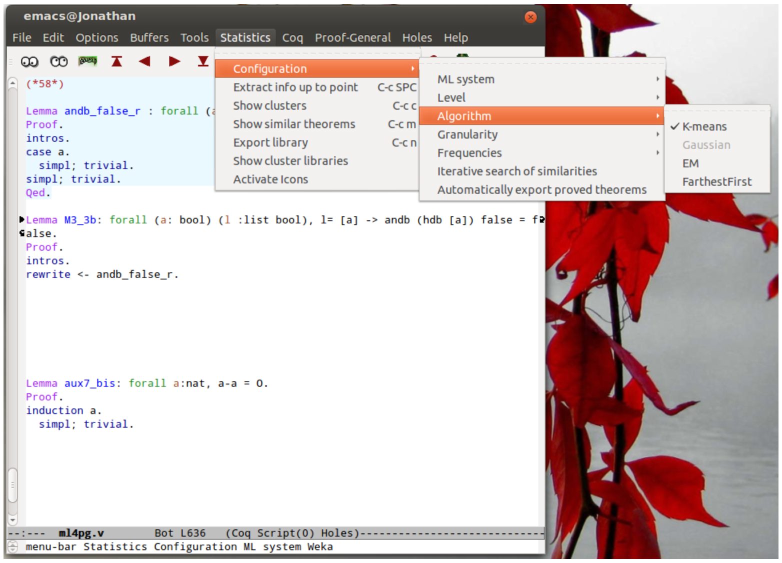
```
Inductive nat : Set :=
  | O : nat
  | S : nat -> nat.

Delimit Scope nat_scope with r

Fixpoint plus (n m:nat) : nat
  match n with
  | O => m
  | S p => S (p + m)
  end

where "n + m" := (plus n m) :

Fixpoint mult (n m:nat) : nat
  match n with
  | O => O
  | S p => m + p * m
  end

where "n * m" := (mult n m) :

Fixpoint minus (n m:nat) : nat
  match n, m with
  | O, _ => n
  | S k, O => n
  | S k, S l => k - l
  end

where "n - m" := (minus n m) :
--:**-  ml4pg.v        Top L1      (Coq Holes)--------------------------------------
Auto-saving...done
```

Figure 1: Proof General with the ML4PG option.


emacs@Jonathan
File Edit Options Buffers Tools Statistics Coq Proof-General Holes Help
Configuration
Extract info up to point C-c SPC
Show clusters C-c c
Show similar theorems C-c m
Export library C-c n
Available libraries for clustering:
✓ Current
peano

```
Inductive nat : Set :=
  | O : nat
  | S : nat -> nat.

Delimit Scope nat_scope with r


Fixpoint plus (n m:nat) : nat :=
  match n with
  | O => m
  | S p => S (p + m)
  end

where "n + m" := (plus n m) : nat_scope.

Fixpoint mult (n m:nat) : nat :=
  match n with
  | O => O
  | S p => m + p * m
  end

where "n * m" := (mult n m) : nat_scope.

Fixpoint minus (n m:nat) : nat :=
  match n, m with
  | O, _ => n
  | S k, O => n
  | S k, S l => k - l
  end

where "n - m" := (minus n m) : nat_scope.
--:---  ml4pg.v        Top L1      (Coq Holes)--------------------------------------
Mark set
```

Figure 2: ML4PG interface with all the options active.



```
(*58*)

Lemma andb_false_r : forall (a
Proof.
intros.
case a.
  simpl; trivial.
simpl; trivial.
Qed.

Lemma M3_3b: forall (a: bool) (l :list bool), l= [a] -> andb (hdb [a]) false = f
alse.
Proof.
intros.
rewrite <- andb_false_r.




Lemma aux7_bis: forall a:nat, a-a = 0.
Proof.
induction a.
  simpl; trivial.
```

```
--:---  ml4pg.v      Bot L636   (Coq Script(0) Holes)----------------------------
menu-bar Statistics Configuration ML system Weka
```

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).

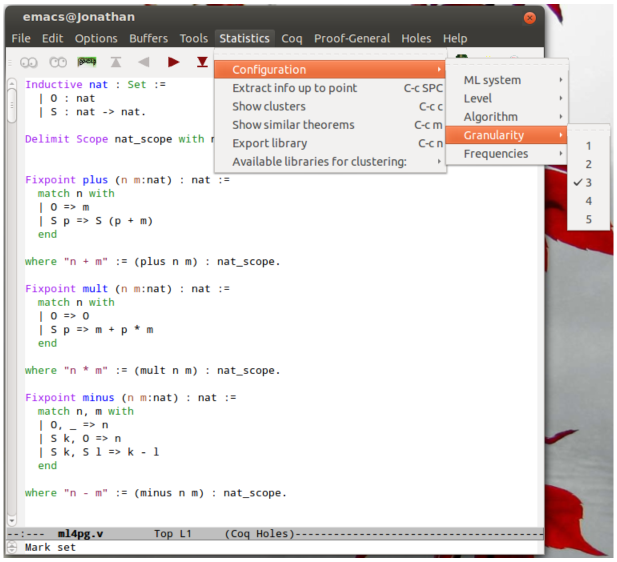
Now, let us explain the functionality of the options included in the Statistics menu.

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 **??**.

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.

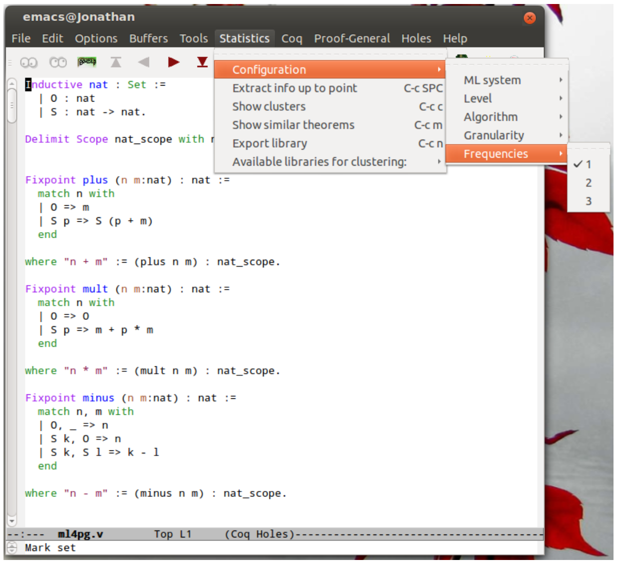


```
Inductive nat : Set :=
  | O : nat
  | S : nat -> nat.

Delimit Scope nat_scope with r

Fixpoint plus (n m:nat) : nat :=
  match n with
  | O => m
  | S p => S (p + m)
  end

where "n + m" := (plus n m) : nat_scope.

Fixpoint mult (n m:nat) : nat :=
  match n with
  | O => O
  | S p => m + p * m
  end

where "n * m" := (mult n m) : nat_scope.

Fixpoint minus (n m:nat) : nat :=
  match n, m with
  | O, _ => n
  | S k, O => n
  | S k, S l => k - l
  end

where "n - m" := (minus n m) : nat_scope.

--:---  ml4pg.v       Top L1     (Coq Holes)------------------------------------
Mark set
```

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].



emacs@Jonathan
File Edit Options Buffers Tools Statistics Coq Proof-General Holes Help
Configuration
Extract info up to point C-c SPC
Show clusters C-c c
Show similar theorems C-c m
Export library C-c n
Available libraries for clustering:
ML system
Level
Algorithm
Granularity
Frequencies
✓ 1
2
3

```
Inductive nat : Set :=
  | O : nat
  | S : nat -> nat.

Delimit Scope nat_scope with r

Fixpoint plus (n m:nat) : nat :=
  match n with
  | O => m
  | S p => S (p + m)
  end

where "n + m" := (plus n m) : nat_scope.

Fixpoint mult (n m:nat) : nat :=
  match n with
  | O => O
  | S p => m + p * m
  end

where "n * m" := (mult n m) : nat_scope.

Fixpoint minus (n m:nat) : nat :=
  match n, m with
  | O, _ => n
  | S k, O => n
  | S k, S l => k - l
  end

where "n - m" := (minus n m) : nat_scope.

--:---  ml4pg.v        Top L1     (Coq Holes)------------------------------------
Mark set
```

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.

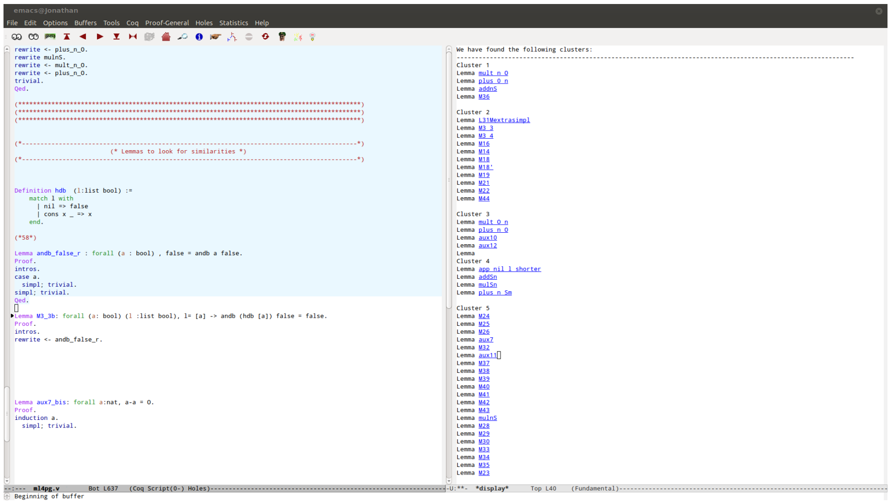
is shown in Figure 6.

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,



```
rewrite <- plus_n_O.
rewrite mulnS.
rewrite <- mult_n_O.
rewrite <- plus_n_O.
trivial.
Qed.

(*****************************************************************************************)
(*****************************************************************************************)
(*****************************************************************************************)


(*---------------------------------------------------------------------------------------*)
                          (* Lemmas to look for similarities *)
(*---------------------------------------------------------------------------------------*)



Definition hdb  (l:list bool) :=
    match l with
      | nil => false
      | cons x _ => x
    end.

(*58*)

Lemma andb_false_r : forall (a : bool) , false = andb a false.
Proof.
intros.
case a.
  simpl; trivial.
simpl; trivial.
Qed.

Lemma M3_3b: forall (a: bool) (l :list bool), l= [a] -> andb (hdb [a]) false = false.
Proof.
intros.
rewrite <- andb_false_r.





Lemma aux7_bis: forall a:nat, a-a = 0.
Proof.
induction a.
  simpl; trivial.
```

```
We have found the following clusters:
------------------------------------------------------------------------------------------------------
Cluster 1
Lemma mult_n_O
Lemma plus_O_n
Lemma addnS
Lemma M36

Cluster 2
Lemma L31Mextrasimpl
Lemma M3_3
Lemma M3_4
Lemma M16
Lemma M14
Lemma M18
Lemma M18'
Lemma M19
Lemma M21
Lemma M22
Lemma M44

Cluster 3
Lemma mult_O_n
Lemma plus_n_O
Lemma aux10
Lemma aux12
Lemma
Cluster 4
Lemma app_nil_l_shorter
Lemma addSn
Lemma mulSn
Lemma plus_n_Sm

Cluster 5
Lemma M24
Lemma M25
Lemma M26
Lemma aux7
Lemma M32
Lemma aux11
Lemma M37
Lemma M38
Lemma M39
Lemma M40
Lemma M41
Lemma M42
Lemma M43
Lemma mulnS
Lemma M28
Lemma M29
Lemma M30
Lemma M33
Lemma M34
Lemma M35
Lemma M23
```

```
--:---  ml4pg.v        Bot L637    (Coq Script(0-) Holes)------------------------------------------
-U:**-  *display*      Top L40     (Fundamental)---------------------------------------------------
Beginning of buffer
```

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.

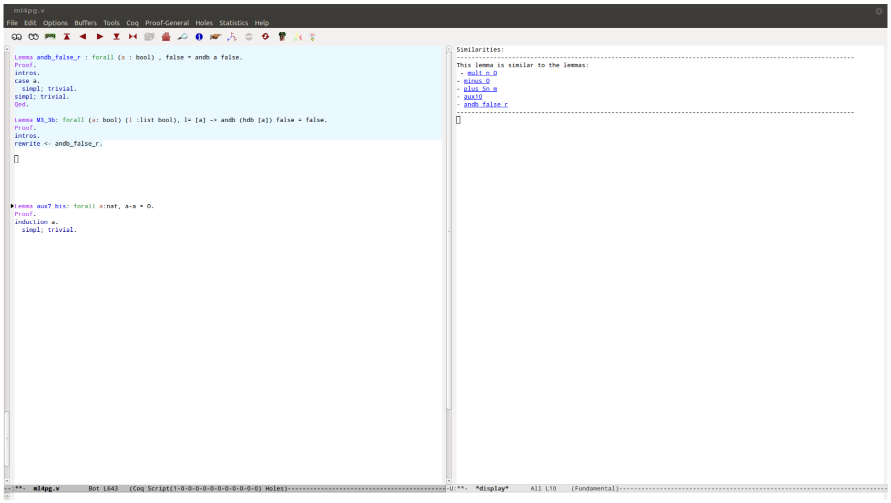
1.6 Export Library

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.

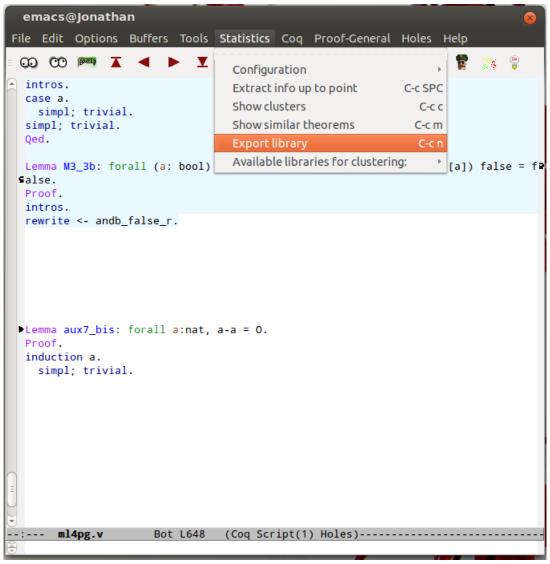

ml4pg.v
File Edit Options Buffers Tools Coq Proof-General Holes Statistics Help

```
Lemma andb_false_r : forall (a : bool) , false = andb a false.
Proof.
intros.
case a.
  simpl; trivial.
simpl; trivial.
Qed.

Lemma M3_3b: forall (a: bool) (l :list bool), l= [a] -> andb (hdb [a]) false = false.
Proof.
intros.
rewrite <- andb_false_r.




Lemma aux7_bis: forall a:nat, a-a = 0.
Proof.
induction a.
  simpl; trivial.
```

```
Similarities:
--------------------------------------------------------------------------------
This lemma is similar to the lemmas:
 - mult_n_O
- minus_O
- plus_Sn_m
- aux10
- andb_false_r
--------------------------------------------------------------------------------
```

```
--:**-  ml4pg.v       Bot L643   (Coq Script(1-0-0-0-0-0-0-0-0-0-0) Holes)-------
-U:**-  *display*      All L10    (Fundamental)-------
```

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.


emacs@jonathan
File Edit Options Buffers Tools Statistics Coq Proof-General Holes Help
Configuration
Extract info up to point C-c SPC
Show clusters C-c c
Show similar theorems C-c m
Export library C-c n
Available libraries for clustering:

```
intros.
case a.
  simpl; trivial.
simpl; trivial.
Qed.

Lemma M3_3b: forall (a: bool) [a]) false = f
alse.
Proof.
intros.
rewrite <- andb_false_r.




Lemma aux7_bis: forall a:nat, a-a = 0.
Proof.
induction a.
  simpl; trivial.
```

```
--:---  ml4pg.v       Bot L648    (Coq Script(1) Holes)-----------------------------
```

Figure 8: ML4PG export menu.