Lynx User Manual

Nicola Di Mauro

November 8, 2011

Contents

1 Introduction

`Lynx` is a system for learning probabilistic models combining relational features. It can be for both relational sequence learning and relational learning.

The following papers provides a description of the system:

- N. Di Mauro, T.M.A. Basile, S. Ferilli, and F. Esposito. *Optimizing Probabilistic Models for Relational Sequence Learning.* In 19th International Symposium on Methodologies for Intelligent Systems, pp. 240-249, Springer, 2011.
- N. Di Mauro, T.M.A. Basile, S. Ferilli, and F. Esposito. *Feature Construction for Relational Sequence Learning.* Technical Report, arXiv:1006.5188, pp. 1-15, 2010.

2 Installation

`Lynx` is distributed in source code, and it includes Prolog and C files. It may be downloaded from `http://www.di.uniba.it/~ndm/lynx/`. `Lynx` needs YAP

Prolog. This version was tested with YAP 6.2.0 (i686-linux). YAP Prolog is available for download at `http://www.dcc.fc.up.pt/~vsc/Yap`.

Compile `Lynx`

- download lynx-1.3.tgz
- decomplress it
- cd lynx-1.3
- `make`

3 Syntax

The input data is stored in three files. The file `.m` contains the training data in form of model (sequences). The file `.val` contains the test data. The file `.bk` contains a given background knowledge. The file `.l` contains the language bias and settings. Each file starts with the problem name.

3.1 `foo.m` and `foo.val`

Each example in the training/test file is represented as follows:

```
begin(ex1, pos).
  p(a,b).
  q(b,c).
  p(b,c).
end.

begin(ex2, neg).
  p(b,c).
  q(b,e).
  p(e,c).
end.
```

The predicate `begin/2` starts the model/sequence. Its first argument is the example ID, while the second argument denotes the class label of the example. `Lynx` is a multiclass learner, you can specify how many class labels you want. The model ends with the `end/0` predicate.

In order to describe sequences, you should use fluent predicates. A fluent predicate contains an argument typed as event. Furthermore, events are connected with a `next/2`. You can use multiple types of `next` predicate, in order to descrive multi dimensional sequences.

```
begin(ex1, pos).
  p(e1,a,b).
  next_o(e1,e2).
  p(e2,b,c).
end.

begin(ex2, neg).
  p(e1,b,c).
  next_o(e1,e2).
  p(e2,e,c).
end.
```

3.2 foo.l

3.2.1 Type and mode declarations

This file contains the language bias described with types and modes.

```
type(molecule(m)).
mode(molecule(+)).

type(bond(m,a,a,c)).
mode(bond(+,#,#,#)).

type(atom(m,a,x,y,z)).
mode(atom(+,#,#,#,#)).
```

For each type, if you wanto to have ground predicates in the model, you can specify the values of each type used in the previous declaration.

```
values(x,[br,c,cl,f,h,i,n,o,s]).
values(c,[1,2,3,4,5,7]).
values(y,[22,195,3,27,38,40,29,10,32,1,14,35,45,8,28,41,16]).

v(bond(_,_,_,_)).
v(bon(_,_,_,_)).
```

For speedup mining you can specify all the possible ground predicates that may be occur.

```
v(atm(_,_,c,22,_)).
v(atm(_,_,c,195,_)).
v(atm(_,_,h,3,_)).
v(atm(_,_,c,27,_)).
v(atm(_,_,n,38,_)).
v(atm(_,_,o,40,_)).
```

This avoud to test, for instance, the predicate `atm(_,_,o,27,_)`.

3.2.2 Settings

`:- set_lynx_flag('$minfreq',k).`
k is the minimum frequency of each pattern for sequence (default 1)

`:- set_lynx_flag('$maxsize',k).`
k is the maximum length of the patterns (default 5)

`:- set_lynx_flag('$minclassrecall', k).`
k is the minimun percentage of class examples a pattern should cover, k $\in [0.0, 1.0]$ (default 0.1)

`:- set_lynx_flag('$maxfollowat', k).`
the maximum distance for the follow at (default 2)

`:- set_lynx_flag('$lynx_verbose', 1).`
the verbosity level, 0 or 1 (default 0)

`:- set_lynx_flag('$mincover', k).`
each example should be covered by at least k patterns (default 1)

`:- set_lynx_flag('$unbalanced_data', true).`
true if the data are class-unbalanced (default false)

`:- set_lynx_flag('$object_identity',true).`
true if the Object Identity should be used for subsuption (default false)

```
%dimension(next_s).
%dimension(next_2).
%dimension(next_3).
```

3.2.3 Constraints

```
sametype([start_sequence(C),end_sequence(C)]).

onlyone([p0(X),p1(X)]).

negconstraint([bon(\_,B,B,\_)]).
negconstraint([atm(A,B,\_,\_,\_),atm(A,B,\_,\_,\_)]).
negconstraint([bon(A,B,C,\_),bon(A,C,B,\_)]).
negconstraint([bon(A,B,C,\_),bon(A,B,C,\_)]).

posconstraint([).

key([m(\_)]).
```

4 Commands

`lynx_i(TRAINFILE).`
perform learning on `TRAINFILE`

`lynx_i(TRAINFILE,TESTFILE).`
perform learning on `TRAINFILE` and test on `TESTFILE`

`lynx_t(TESTFILE).`
test on `TESTFILE`

`lynx_flags.`
show all settings

`lynx_l(MODELFILE,TESTFILE).`
load a classification model from MODELFILE and then test on TESTFILE

`get_lynx_flag(FLAG,VALUE).`
get current `VALUE` for `FLAG`

`set_lynx_flag(FLAG,VALUE).`
set `VALUE` for `FLAG`

`halt.`
quit YAP

`help.`
show the help

5 Licence

The Lynx Software is made available under the terms and conditions of the Artistic License 2.0. LICENSEE shall acknowledge University of Bari "Aldo Moro" as the provider of the Software.

Artistic License 2.0

Copyright (c) 2000-2006, The Perl Foundation.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

This license establishes the terms under which a given free software Package may be copied, modified, distributed, and/or redistributed. The intent is that the Copyright Holder maintains some artistic control over the development of that Package while still keeping the Package available as open source and free software.

You are always permitted to make arrangements wholly outside of this license directly with the Copyright Holder of a given Package. If the terms of this license do not permit the full use that you propose to make of the Package, you should contact the Copyright Holder and seek a different licensing arrangement.

Definitions

"Copyright Holder" means the individual(s) or organization(s) named in the copyright notice for the entire Package.

"Contributor" means any party that has contributed code or other material to the Package, in accordance with the Copyright Holder's procedures.

"You" and "your" means any person who would like to copy, distribute, or modify the Package.

"Package" means the collection of files distributed by the Copyright Holder, and derivatives of that collection and/or of those files. A given Package may consist of either the Standard Version, or a Modified Version.

"Distribute" means providing a copy of the Package or making it accessible to anyone else, or in the case of a company or organization, to others outside of your company or organization.

"Distributor Fee" means any fee that you charge for Distributing this Package or providing support for this Package to another party. It does not mean licensing fees.

"Standard Version" refers to the Package if it has not been modified, or has been modified only in ways explicitly requested by the Copyright Holder.

"Modified Version" means the Package, if it has been changed, and such changes were not explicitly requested by the Copyright Holder.

"Original License" means this Artistic License as Distributed with the Standard Version of the Package, in its current version or as it may be modified by The Perl Foundation in the future.

"Source" form means the source code, documentation source, and configuration files for the Package.

"Compiled" form means the compiled bytecode, object code, binary, or any other form resulting from mechanical transformation or translation of the Source form.

Permission for Use and Modification Without Distribution

(1) You are permitted to use the Standard Version and create and use Modified Versions for any purpose without restriction, provided that you do not Distribute the Modified Version.

Permissions for Redistribution of the Standard Version

(2) You may Distribute verbatim copies of the Source form of the Standard Version of this Package in any medium without restriction, either gratis or for a Distributor Fee, provided that you duplicate all of the original copyright notices and associated disclaimers. At your discretion, such verbatim copies may or may not include a Compiled form of the Package.

(3) You may apply any bug fixes, portability changes, and other modifications made available from the Copyright Holder. The resulting Package will still be considered the Standard Version, and as such will be subject to the Original License.

Distribution of Modified Versions of the Package as Source

(4) You may Distribute your Modified Version as Source (either gratis or for a Distributor Fee, and with or without a Compiled form of the Modified Version) provided that you clearly document how it differs from the Standard Version, including, but not limited to, documenting any non-standard features, executables, or modules, and provided that you do at least ONE of the following:

(a) make the Modified Version available to the Copyright Holder of the Standard Version, under the Original License, so that the Copyright Holder may include your modifications in the Standard Version.

(b) ensure that installation of your Modified Version does not prevent the user installing or running the Standard Version. In addition, the Modified Version must bear a name that is different from the name of the Standard Version.

(c) allow anyone who receives a copy of the Modified Version to make the Source form of the Modified Version available to others under

(i) the Original License or

(ii) a license that permits the licensee to freely copy, modify and redistribute the Modified Version using the same licensing terms that apply to the copy that the licensee received, and requires that the Source form of the Modified Version, and of any works derived from it, be made freely available in that license fees are prohibited but Distributor Fees are allowed.

Distribution of Compiled Forms of the Standard Version or Modified Versions without the Source

(5) You may Distribute Compiled forms of the Standard Version without the Source, provided that you include complete instructions on how to get the Source of the Standard Version. Such instructions must be valid at the time of your distribution. If these instructions, at any time while you are carrying out such distribution, become invalid, you must provide new instructions on demand or cease further distribution. If you provide valid instructions or cease distribution within thirty days after you become aware that the instructions are invalid, then you do not forfeit any of your rights under this license.

(6) You may Distribute a Modified Version in Compiled form without the Source, provided that you comply with Section 4 with respect to the Source of the Modified Version.

Aggregating or Linking the Package

(7) You may aggregate the Package (either the Standard Version or Modified Version) with other packages and Distribute the resulting aggregation provided that you do not charge a licensing fee for the Package. Distributor Fees are permitted, and licensing fees for other components in the aggregation are permitted. The terms of this license apply to the use and Distribution of the Standard or Modified Versions as included in the aggregation.

(8) You are permitted to link Modified and Standard Versions with other works, to embed the Package in a larger work of your own, or to build stand-alone binary or bytecode versions of applications that include the Package, and Distribute the result without restriction, provided the result does not expose a direct interface to the Package.

Items That are Not Considered Part of a Modified Version

(9) Works (including, but not limited to, modules and scripts) that merely extend or make use of the Package, do not, by themselves, cause the Package to be a Modified Version. In addition, such works are not considered parts of the Package itself, and are not subject to the terms of this license.

General Provisions

(10) Any use, modification, and distribution of the Standard or Modified Versions is governed by this Artistic License. By using, modifying or distributing the Package, you accept this license. Do not use, modify, or distribute the Package, if you do not accept this license.

(11) If your Modified Version has been derived from a Modified Version made by someone other than you, you are nevertheless required to ensure that your Modified Version complies with the requirements of this license.

(12) This license does not grant you the right to use any trademark, service mark, tradename, or logo of the Copyright Holder.

(13) This license includes the non-exclusive, worldwide, free-of-charge patent license to make, have made, use, offer to sell, sell, import and otherwise transfer the Package with respect to any patent claims licensable by the Copyright Holder that are necessarily infringed by the Package. If you institute patent litigation (including a cross-claim or counterclaim) against any party alleging that the Package constitutes direct or contributory patent infringement, then this Artistic License to you shall terminate on the date that such litigation is filed.

(14) Disclaimer of Warranty: THE PACKAGE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS "AS IS' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES. THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT ARE DISCLAIMED TO THE EXTENT PERMITTED BY YOUR LOCAL LAW. UNLESS REQUIRED BY LAW, NO COPYRIGHT HOLDER OR CONTRIBUTOR WILL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING IN ANY WAY OUT OF THE USE OF THE PACKAGE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.