

## Machine learning for ICALL

### L700: Automatic Analysis of Learner Language

Autumn 2008

## Machine learning and what it's good for

- ▶ Machine learning is the process of a computer learning from data
  - ▶ Break the problem space down into a classification problem, i.e., a way of assigning a category
  - ▶ Break down the input into features describing the instance

Machine learning could be extremely useful to detecting and diagnosing errors

- ▶ Teach a machine learner what errors look like → Problem: lack of annotated corpora to train on
- ▶ Teach a machine learner what correct language looks like and look for aberrations from this correct model

## Different classification techniques

A sampling of software which can be used to do classification tasks:

- ▶ PyML (Machine Learning in Python): <http://pymml.sourceforge.net/>
  - ▶ SVMs, *k*-nn, ridge regression; Requires NumPy package
- ▶ mlpy (Machine learning for python): <https://mlpy.fbk.eu/>
  - ▶ SVM, SRDA, FDA, PDA, NN; Requires NumPy package
- ▶ Weka: <http://www.cs.waikato.ac.nz/ml/weka/>
  - ▶ Java-based; Decision trees, Naive Bayes, logistic regression, SVMs, IB/*k*-nn, ...
- ▶ SVM<sup>light</sup>: <http://svmlight.joachims.org/>
- ▶ Maximum Entropy-related software: <http://homepages.inf.ed.ac.uk/s0450736/maxent.html>
- ▶ ...

## Memory-based learning (MBL)

Memory-based learning (MBL) is one way of doing machine learning, a form of nearest neighbors classifying:

- ▶ all training instances are stored in a database
- ▶ new instances are compared to those in the database to see which are the closest
  - ▶ they are given the most likely label from the nearest neighbors stored in memory

There are a variety of ways to score neighbors.

## Obtaining TiMBL

1. Go to: <http://ilk.uvt.nl/timbl/>
2. Click on the Download link → this will give you a file `timbl-6.1.2.tar.gz`
  - ▶ A very useful document on this page is the Reference Guide
3. Unpack the file in a place you want to keep it
4. Go through the directions in the INSTALL file to install timbl

## Running TiMBL

If I type `timbl` at the terminal, I get something like this:

```
TimBL 6.0(Release) (c) ILK 1998 - 2007.
Tilburg Memory Based Learner
Induction of Linguistic Knowledge Research Group
Tilburg University / University of Antwerp
Fri Nov 14 14:22:53 2008
```

```
usage: Timbl -f data-file [-t test-file]
or see: Timbl -h
        for all possible options
```

In other words, to run timbl, do this:

```
timbl -f training-data -t testing-data
```

## Exploring TiMBL's options

You can run timbl with a host of different options. See chapter 6 of the reference guide.

Main options:

- ▶ `-a` sets the classification algorithm
- ▶ `-m` sets the distance metric (and can also be used to ignore features or define features as numeric)
- ▶ `-w` sets the feature weighting possibilities
- ▶ `-k` sets the number of nearest neighbors

Many of the output options can be useful, in that they will show more of what happened inside timbl in order to classify instances

## Testing different options

Just a note on exploring the space of options:

- ▶ You'll want to clearly separate your data into disjoint sets:
  - ▶ Training data (80-90%)
  - ▶ Development data (5-10%)
  - ▶ Testing/Evaluation data (5-10%)
- ▶ Explore the space of options on the *development* data
  - ▶ Once you've optimized the model and have reasons behind the optimizations, then you can see whether it works on the testing data

## Data formats

There are a variety of possible formats. The main ones are:

- ▶ Column format: white space between features
- ▶ C4.5 format: commas between features

If you need to refer to a space or a comma in your data, you should recode it as something else, e.g., `<COMMA>`

The last feature you put on a line is the class, i.e., the value you're trying to predict

## Experimenting with TiMBL

1. Create a directory to experiment in: from `timbl-6.1.2/demos/`, copy `dimin.train` and `dimin.test`
2. Run timbl and walk through the output
3. Try getting different output options, such as adding the distance of the nearest neighbor to the output file or generating a confusion matrix
4. Try different learning options, such as ignoring features; using MVDM for certain features; etc.
5. Try concatenating some of the features and changing the input file by adding this new concatenated feature

## Integrating TiMBL into programs

Information on the C++ API is at:

- ▶ [http://ilk.uvt.nl/downloads/pub/papers/Timbl\\_6.1\\_API.pdf](http://ilk.uvt.nl/downloads/pub/papers/Timbl_6.1_API.pdf)

You could also try putting timbl into a python program:

- ▶ <http://ilk.uvt.nl/~sander/software/python-timbl.html>

This will allow you to classify individual instances without relearning every time.