EDLIN Technical Documentation

Introduction

Edlin is a collection of machine learning algorithms comprising a large number of state-of-the-art methods for classification and sequence tagging. Even though at their core they are general machine-learning approaches (perceptrons, logistic regression), the implementation is optimized for NLP learning tasks:

- inputs are represented as sparse document-term matrices
- parallel computation is used whenever possible (in order to deal with very large datasets)
- specific evaluation metrics such as Precision/Recall/F are being reported
- appropriate feature selection methods are added in order to reduce dimensionality, etc.

Edlin consists of four sub-projects(Basics, Edlin-Wrapper, Mallet-Wrapper and Feature Extraction). Below find technical details on each of the sub-projects. The source can be found here.

Edlin Basics

Edlin Basics is the core of the tool, containing all ML algorithms divided into two general groups: classification and sequence (tagging).

Algorithms for classification

All algorithms are used for multi-class classification. Typically, for each input example, the raw model output is a list of scores, one for each class. As a final prediction, the class with largest score is chosen.

Maxent

Maximum Entropy (Maxent) is essentially multi-class logistic regression. It was first adapted and applied to NLP tasks by Berger, et al (1996) and Della Pietra, et al. 1997 (Adam L. Berger , Stephen A. Della Pietra , Vincent J. Della Pietra, A Maximum Entropy approach to Natural Language Processing, Journal of Computational Linguistics, 1996, vol. 22, pp. 39-71). Maxent is a linear model, where the posterior class probabilities are modelled as a linear combination of the input features. In order to fit the model weights to the training observations, a loglikelihood loss function is maximized. The maximization is carried out by some gradient ascent method. The output of the maxent model for a given example is a list of posterior class probabilities, out of which the largest is chosen as prediction.

Several parameters for maxent can be selected, such that the most appropriate and efficient model is trained to suite a particular dataset:

- Gradient method:
  - Gradient ascent
  - Conjugate gradient
  - Stochastic gradient ascent (fast)
- Parallelization:
  - An approach to multithreaded maxent by Mann et al. (2009), (Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models)
- Modified objective for targeted optimization of particular Precision/Recall trade-off:
  - We implemented a weighted likelihood objective that allows for optimizing a specific F_beta, for a given beta, which means that we can specify a desired Precision/ Recall trade-off. In practice, we can therefore train models that have very high Precision, or very high Recall, at the expense of the complementary measure.
  - Main publication: Dimitroff et al. Weighted maximum likelihood as a convenient shortcut to optimize the F-measure of maximum entropy classiers, RANLP 2013
- Regularization:
  - L1 regularization is often used in practice for sparse models and reducing overfitting. An L1-regularized maxent can also serve

as feature selection procedure.

Perceptron

The Perceptron is a very old algorithm for training a linear model invented by F. Rosenblatt (Rosenblatt, Frank (1957), The Perceptron--a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory). It is an online algorithm, meaning that it is trained by observing one input example at a time, by updating upon mistakes.

- Parallelization
  - The online training of the perceptron allows for easy parallelization. Batches of data are fed independently to several 'temporary' perceptrons and then they are averaged into a single model. The procedure is repeated for a number of epochs (see "Distributed Training Strategies for the Structured Perceptron", McDonald et al. (2010))
  - For many datasets, the perceptron is the fastest algorithm of the entire EDLIN collection.
  - For speed reasons, it is preferred for learning large datasets.
- Sigmoid perceptron
  - We implemented a modification of the perceptron that allows for a probabilistic output. Here is a draft of the paper.

Naive Bayes

The Naive Bayes is a linear classifier that is widely used in NLP. It is estimating the posterior class probabilities based on term frequencies, using the Bayes formula. Naive Bayes models have a strong assumption of independence of features, which often does not hold and makes the model underperform.

Algorithms for sequence tagging (or labelling)

Sequence tagging is a typical NLP task, where classification of all components of a sequence is done at once, as opposed to splitting into many separate instances. If classified together, sequence context features ensure better performance. A classical example of sequence tagging is *part-of-speech tagging*, where the best global assignment of a set of part of speech labels is inferred.

Conditional random fields (CRF)

The CRF algorithm has been proposed in Lafferty, J., McCallum, A., Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data, Proc. 18th International Conf. on Machine Learning. .

Perceptron

The structured perceptron, with parallelization is implemented after : McDonald et al. (2010), (Distributed Training Strategies for the Structured Perceptron)

Algorithms for feature selection

NLP datasets are characterized by a large number of features, sometimes order of magnitudes higher than the number of training samples available. In order to avoid overfitting, feature selection can be used prior to or during model training. We have a large number of approaches to feature selection:

- Filter by Fisher test (association between feature and outcome), either keeping a percent of features, or keeping the features yielding small enough p-value.
- Filter by mutual information (between feature and outcome), either keeping a percent of features, or keeping the features yielding large enough mutual information
- A feature induction algorithm, described here: (Tolosi et. al. 2013 A Feature Induction Algorithm with Application to Named Entity Disambiguation. RANLP 2013)

Feature Extraction module

A module for feature extraction: classification instances are produced automatically, as features are extracted from documents using a set of Groovy rules.

Edlin-Wrapper(for GATE)

Edlin-Wrapper wraps the algorithms of Edlin, so that they can be used in GATE for multiple information extraction purposes.
The algorithms are wrapped as ProcessingResources and LanguageResources and can be applied directly in a pipeline.
More about [Edlin-Wrapper]

Mallet-Wrapper(for GATE)

Mallet-Wrapper wraps the algorithms of Mallet, so that they can be used in GATE for multiple information extraction purposes. The algorithms are wrapped as ProcessingResources and LanguageResources and can be applied directly in a pipeline.