

# 1 MaxEnt source code review

Source code review of Tsuruoka's C++ library for maximum entropy classification (see <http://www.nactem.ac.uk/tsuruoka/maxent/>).

## 1.1 Introduction

On his website, [Tsuruoka](#) proposed a C++ library for maximum entropy classification. In order to get a better and deeper understanding of implementation details, I propose here a simple code review. Those notes are primary destined for my personal use and reflect my current understanding. I propose them here, in case where it could help someone.

## 1.2 Outline

Using [Rexdep](#), we can obtain the following directed graph of dependencies:

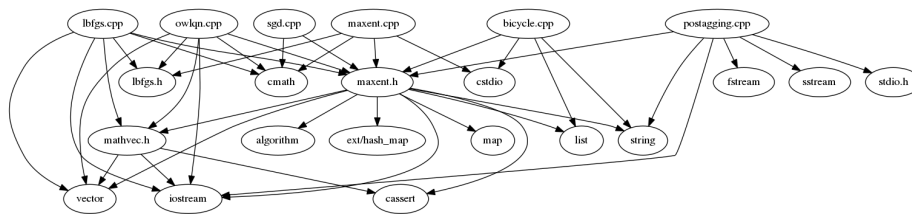


Figure 1: Dependencies

The core of the library are the header `maxent.h`, and its implementation `maxent.cpp`. In this core, the fundamental structures and functions of the learning task are defined (training set, computing of objective, etc.). The following section will discuss some of these aspects. Besides that, three optimisations methods of the objective are given:

- `lbfgs.cpp` : the very efficient [BFGS algorithm](#) in limited memory version;
- `owlqn.cpp` : a [variant of BFGS](#) for L1-regularized models
- `sgdl.cpp` : the classic [stochastic gradient descent](#)

Note that, due to their shared nature, `lbfgs.cpp` and `owlqn.cpp` include both the header `lbfgs.h`.

Two examples are proposed and include only the core, `maxent.h`:

- `bicycle.cpp` : binary classification between cars and bicycles given simple features. This example is made to just figure out the basic principles;

- `postaggin.cpp` : the classic task which aims at identifying the right part-of-speech (POS) for a given token. A little dataset of 200 sentences is given to train and test the models;

The `mathvec.h` is a data structure which extends the possibilities of `std::vector<double>` : the operators are overloaded and some facilities are give for dot product and vector projection. Those elements are extensively used in `lbfgs.cpp`, `owlqn.cpp` and `maxent.h`.

This article is divided in three parts:

- The core
- Optimization methods
- Postagging example

### 1.3 The core

### 1.4 Optimization methods

### 1.5 POSTagging