CS 769 Course Project

Efficiently Maintaining Conditional Random Fields

Xixuan (Aaron) Feng, and Guangde Chen

Summary. Real-world systems, such as isWiki, Alibaba, Citeseer, Kylin and YAGO, have a growing need of statistical information extraction (IE) programs. Motivated by this need, our work concerns the efficiency of maintaining the underlying model of statistical IE over incoming training examples. In real-world settings, new examples often need to be incorporated. In order to keep the underlying model up to date, state-of-the-art statistical IE systems have to repeat the process of parameter estimation from scratch over each batch of new examples, which is usually time-consuming. To understand this problem, we present, evaluate, and analyze four maintenance strategies of the most popular statistical IE, conditional random fields (CRF). We also study how regularization affects these strategies.

1. Introduction

Information extraction (IE) has been widely used in the industry and academia. Well-known examples are system T in IBM and DBLife in University of Wisconsin, and both of them use rule-based IE. In the discussion with both, we found out that one of major concerns of using statistical methods is flexibility and efficiency. However, advantages that come with statistical methods are not negligible. HP Labs isWiki, Microsoft Academic Search, etc. are using conditional random fields (CRFs) for their IE system. In this work, we consider a problem of deploying CRF for real-world IE systems, efficiently maintaining underlying model with new training examples constantly being added.

Problem Definition Given an initial CRF training set S_0, let $m*_0 = train(S_0)$ be the optimal model estimated using all examples in S_0. After each constant period of time, constant number C new examples are inserted into the training pool. Let $S_1, S_2, S_3, ...$ denote consecutive snapshots of the training pool, $|S_i| - |S_{i-1}| = C$. For each S_i, our goal is to estimate a model m_i that is close enough to $m*_i$ as efficiently as possible.

2. Related Work

The efficiency of deploying statistical information extraction (IE) has been studied a lot recently in the database community [1, 2, 3]. But none of them has discussed cases that the underlying statistical model can potentially change.

The problem of on-line learning and sparsity are discussed by Langford [4], but only linear models not graphical models like conditional random fields (CRF) are handled. Our implementation use incremental gradient descent [5] and regularization with penalty terms [6].

Concept drifting [7] has very related motivation, but it focuses on generating new models that can capture the new concepts that do not exist in the old population, while we don't have such assumptions. And efficiency problems are not well discussed.

3. Methods

3.1. Retrain From Scratch

3.2. Incrementally Refine Previous Model

4. Experiments

The whole data set has been divided into three subsets: training set (sample size 3K), new set (sample size 936), and test set (sample size 5K). Several experiments have been implemented as follows.

4.1. Experiment 1

In this experiment, samples are randomly selected from training set to train the model and the loss value is calculated based on the test set. We start from 10 samples,and recruit 10 new samples every incremental step until the samples get to 150. For each run, the iteration is fixed at 50 and the experiment has been duplicated four times. The average loss values and training times are in the Table 1.

Table 1. The loss values and training time in different sample sizes; All are the average of 4 duplicates.

Sample sizes	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
Loss values	30752	25029	22655	21006	20274	19239	18933	18295	17793	17552	17278	16914	16797	16676	16391
Training time(sec.)	66	73	76	83	87	91	95	98	102	106	108	112	115	118	122

As we can see from Figure 1, the loss values deceases when the sample size increases. One important thing we observed is that the loss values tend to converge after a certain sample size, which suggests that a small subset of samples could be enough to train the model. There is no surprise in Figure 2, that the training time is increasing as the sample sizes increase. Therefore, for the computational time of sake, it is necessary to consider a small subset of samples to train the model instead of using all the samples.

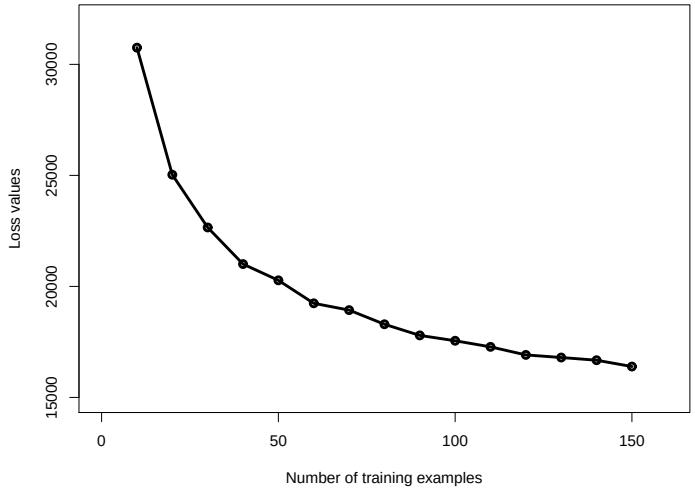

Fig. 1. The loss values v.s. different sample sizes

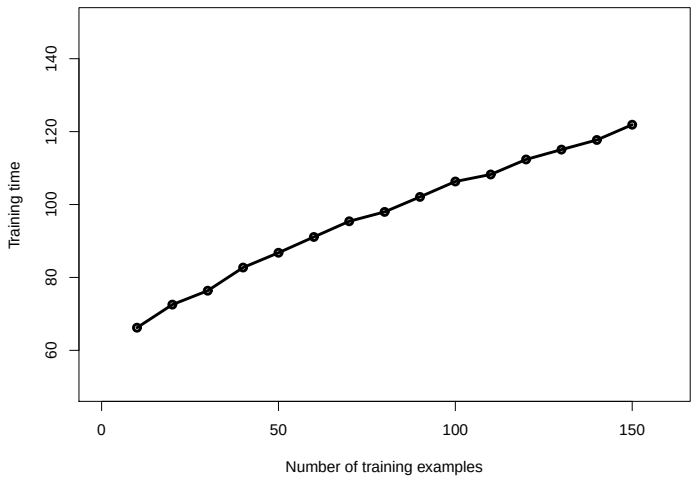

Fig. 2. The training time v.s. different sample sizes

4.2. Experiment 2

Above experiment, the iteration is fixed. As we all know, the iteration will affect the experiment not only on the computational time, but also on the accuracy to train the model. A new experiment has been set up as follows. For a curtain subset of samples, we combines a new examples at each time. We use the estimated parameters to calculate the loss values from the test set. It is shown that in Figure 3, the loss values tend to decrease with the increase of new examples.

4.3. Experiment 3

In this experiment, at each run, we have 11 examples which include 1 new example and 10 other examples randomly selected from the previous example pool. For each run, the initial parameters are from the previous parameters (in the first run, we set all parameters as zero), then we run the gradient descent to estimate the parameters. Figure 4 shows that as more and more new examples arrives, the loss values tend to be stable as a surprise. This clearly demonstrates that the proposal is totally correct and our problem is to find a good way to sample the examples from previous example pool as we can see that there is some fluctuation at each run.

5. New results

5.1. Experiment 1

5.2. Experiment 2

5.3. Experiment 3

5.4. Experiment 4

6. Conclusions

7. References

[1] M. Wick, A. McCallum, and G. Miklau, "Scalable probabilistic databases with factor graphs and mcmc", PVLDB, 3(1), 2010.

[2] D. Z. Wang, M. J. Franklin, M. Garofalakis, J. M. Hellerstein, and M. L. Wick, Hybrid in-database inference for declarative information extraction, SIGMOD-11.

[3] F. Chen, X. Feng, C. R and M. Wang, "Optimizing Statistical Information Extraction Programs Over Evolving Text", ICDE-2012, to appear.

[4] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient", NIPS-08.

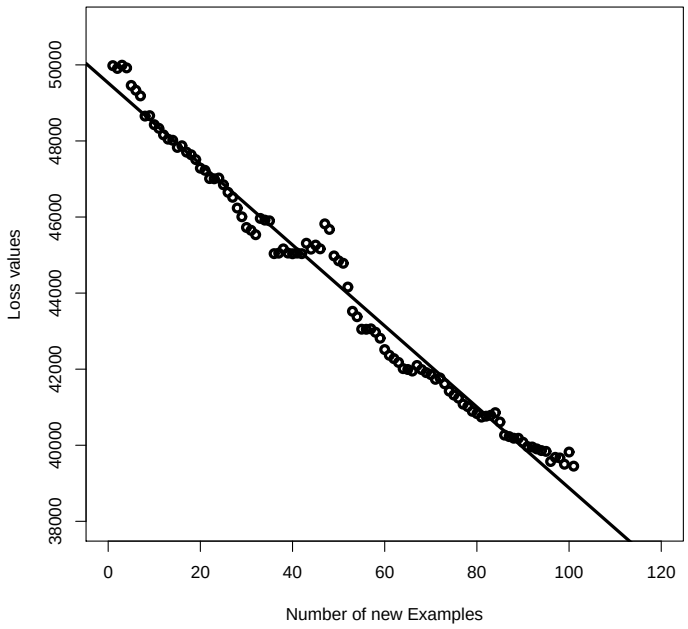

Fig. 3. The loss values v.s. number of new examples

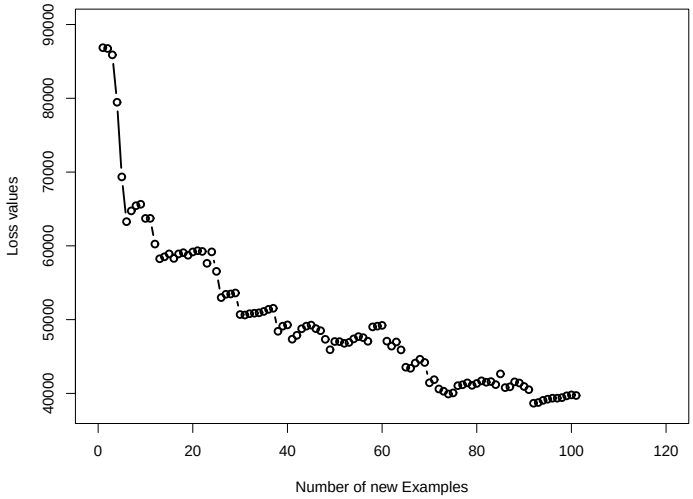

Fig. 4. The loss values v.s. number of new examples

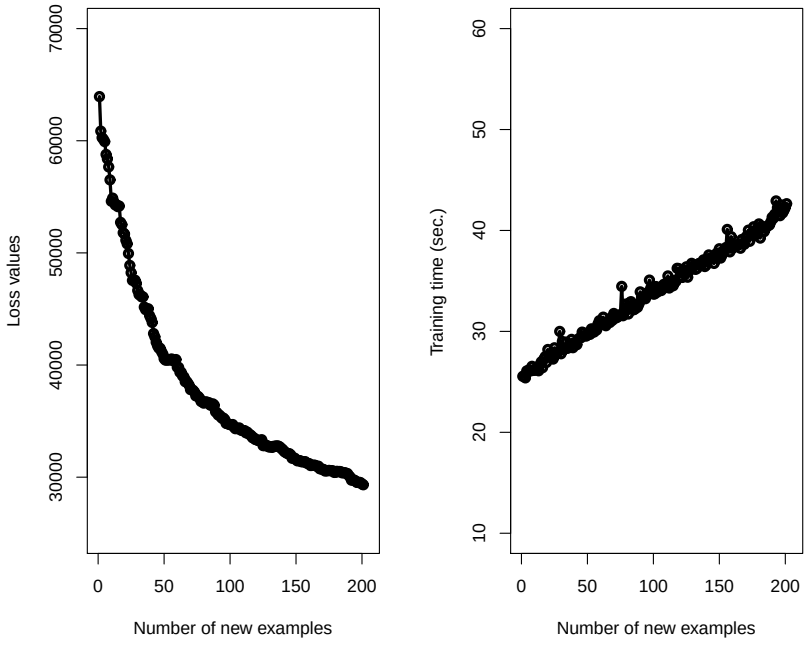

Fig. 5. The loss values and training time v.s. number of new examples. Each time when a new example arrives, we train the model with **all the examples** and from **initial zero parameters**

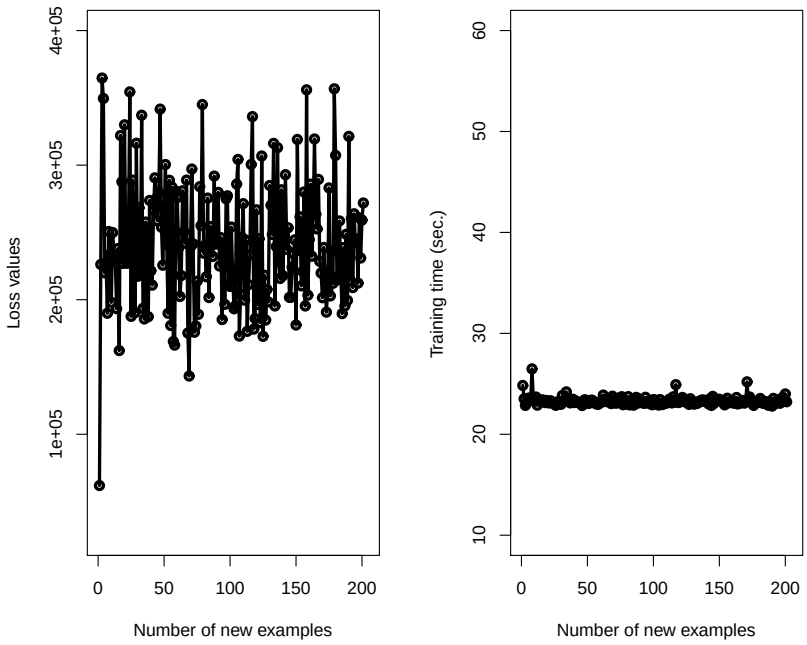

Fig. 6. The loss values and training time v.s. number of new examples. Each time when a new example arrives, we train the model with **the new example only** and from **initial zero parameters**

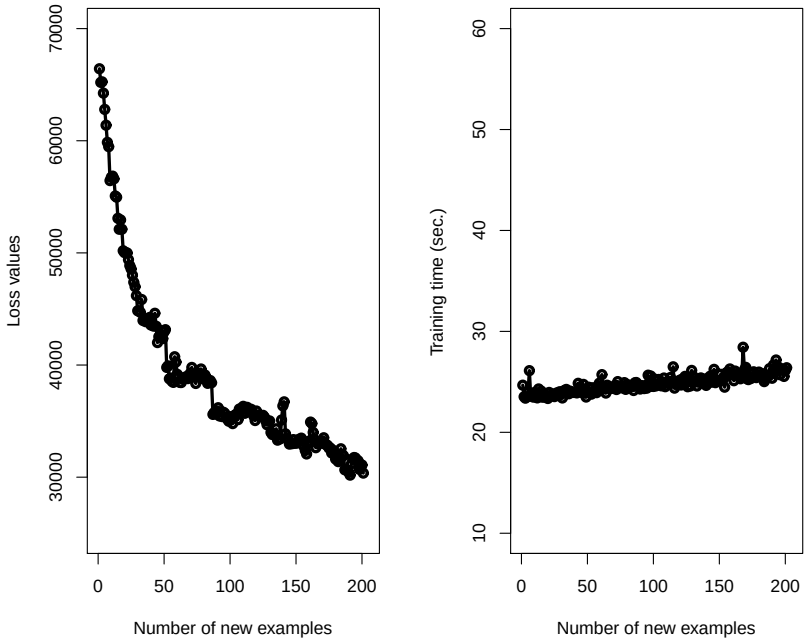

Fig. 7. The loss values and training time v.s. number of new examples. Each time when a new example arrives, we train the model with **the new example only** and from **previous parameters**

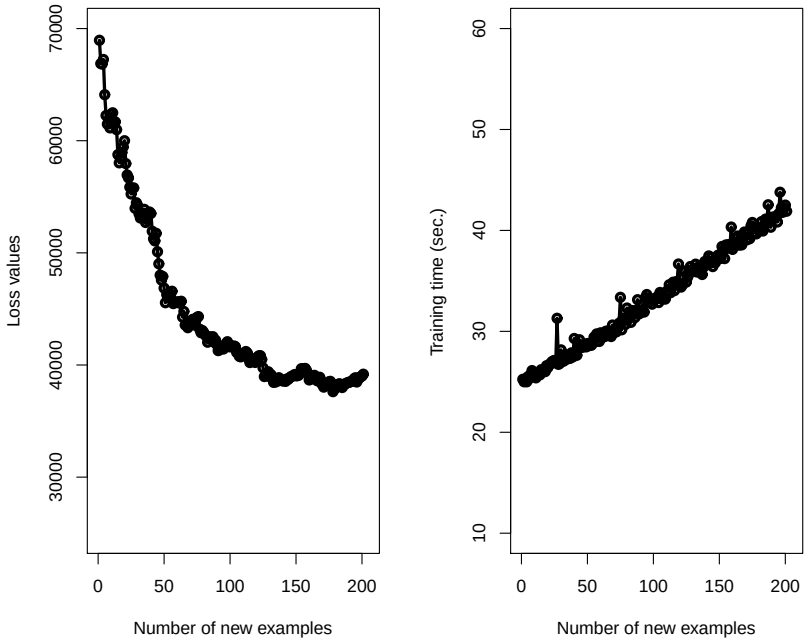

Fig. 8. The loss values and training time v.s. number of new examples. Each time when a new example arrives, we train the model with **all the examples** and from **previous parameters**

[5] D. P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. Technical report, Laboratory for Information and Decision Systems, 2010.

[6] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty", ACL-09.

[7] A. Tsymbal, "The problem of concept drift: Denitions and related work", Technical Report TCD-CS-2004-15, Trinity College Dublin, 2004.