CRF versus SVM-Struct for Sequence Labeling

S. Sathiya Keerthi SELVARAK@YAHOO-INC.COM
Yahoo! Research, Santa Clara, USA

S. Sundararajan SSRAJAN@YAHOO-INC.COM
Advanced Technology Group, Yahoo! SDC India Pvt Ltd, Bangalore, India

Abstract

A recent comparison of various algorithms for sequence labeling by Nguyen and Guo indicated that SVM-struct has much superior generalization performance than CRFs. In this short report we point out that the above difference mainly arises because that comparison employed different softwares that use different internal feature functions. When the two methods are compared using identical feature functions they do turn out to have quite close peak performance.

1. Introduction

Sequence labeling consists of assigning a sequence of labels (states) $y = \{y_t\}_{t=1}^T$ to a sequence of inputs $x = \{x_t\}_{t=1}^T$. (T can depend on each instance of the sequence.) Each y_t takes on a value from $\{1, \ldots, n_c\}$ denoting n_c classes. To do this, structured output methods such as Conditional Random Fields (CRFs) (Sutton and McCallum, 2006) and SVM-struct (Tsochantaridis et al, 2005) form a vector of feature functions $\phi(x, y)$ (which use the sequential structure, e.g. $\phi(x, y) = \sum_t \phi_t$ where ϕ_t is the feature vector associated with the t-th token and its sorrounding information; see section 2) and a scoring function $s(x, y; w) = w^T\phi(x, y)$. The parameter vector w is learnt by using a training set T of (x, y) pairs and optimizing

$$\min_w \frac{\lambda}{2}\|w\|^2 + \sum_{(x,y)\in T} L(x, y; w) \qquad (1)$$

where λ is a regularization constant that is chosen via validation. CRFs use the negative log-likelihood loss function, $L(x, y; w) = -\log[\exp s(x, y; w)/Z]$. Here Z is the partition function $Z = \sum_{\bar{y}} \exp s(x, \bar{y}; w)$ where $\bar{y}$ runs over all possible sequences, i.e. $\bar{y} = \{\bar{y}_t\}_{t=1}^T$ and $\bar{y}_t \in \{1, \ldots, n_c\}$. SVM-struct uses the margin based loss function, $L(x, y; w) = \max_{\bar{y}} s(x, \bar{y}; w) + \Delta(\bar{y}, y) - s(x, y; w)$ where $\Delta(\bar{y}, y)$ is the loss associated with classifying y as $\bar{y}$ and it is usually taken to be the Hamming loss, i.e. the number of tag differences between y and $\bar{y}$.

Recently Nguyen and Guo (2007) compared a set of sequence labeling algorithms that includes CRFs and SVM-struct. Results on two datasets seem to indicate that SVM-struct has much better generalization performance than CRFs. The comparison was done using the *Mallet* software (McCallum, 2002) for CRFs and the SVM^{hmm}software (Herbst and Joachims) for SVM-struct. These softwares employ different feature functions $\phi(x, y)$. The aim of this report is to point out that it is this difference in feature functions that mainly caused the performance difference between CRFs and SVM-struct in the evaluation done by Nguyen and Guo. To show this we wrote our own code for both CRFs and SVM-struct with the ability to uniformly toggle various types of feature functions. Doing experiments on the same two datasets (POS and OCR) used by Nguyen and Guo as well as an additional dataset (CORA) we find that when the right choices are made for the feature functions (which can be done via validation) the generalization performances of CRFs and SVM-struct are quite close. The fact that the choice of feature functions has a crucial effect on performance has also been pointed out before, e.g. in Peng and McCallum, 2004. Section 2 discusses feature functions and section 3 details the experiments and results.

2. Feature Functions

Let us discuss feature functions associated with a training example, (x, y), with $x = \{x_t\}_{t=1}^T$ and $y = \{y_t\}_{t=1}^T$. We can add two dummy boundary states,

This is a Yahoo Research Technical Report.

$y_0 = s$ (start state) and $y_{T+1} = e$ (end state). At each t a bunch of feature functions (ϕ_t) are formed. The following are some examples of feature functions.

- *First order:* $\phi(y_t, x_t)$ (for $1 \leq t \leq T$)
- *Second order forward:* $\phi(y_{t-1}, y_t, x_{t-1})$ (for $2 \leq t \leq T$)
- *Second order backward:* $\phi(y_{t-1}, y_t, x_t)$ (for $2 \leq t \leq T$)
- *Boundary:* $\phi(s, y_1, x_1)$ and $\phi(y_T, e, x_T)$
- *Token-independent first order:* $\phi(y_t)$ (for $1 \leq t \leq T$)
- *Token-independent second order:* $\phi(y_{t-1}, y_t)$ (for $2 \leq t \leq T$)
- *Token-independent boundary:* $\phi(s, y_1)$ and $\phi(y_T, e)$

Typically the ϕ are simple boolean functions. For example if x_t^j (the j-th element of the token input vector x_t) is a boolean variable then a suitable first order feature function is $(y_t = i) \wedge (x_t^j = 1)$, where i denotes the i-th class. Only feature functions that 'occur' in the training set need to be considered. For example, if we take the token-independent boundary feature case, if class 1 never occurs as the starting token label, i.e., y_1, we do not assign a feature function for $\phi(s, y_1 = 1)$.

Some combinations of the above-mentioned feature functions for which we have carried out empirical study are as follows.

- $\boldsymbol{F_1}$: Use first order and token-independent first order functions.
- $\boldsymbol{F_2}$: Use $\boldsymbol{F_1}$ and token-independent second order functions.
- $\boldsymbol{F_3}$: Use second order forward functions.
- $\boldsymbol{F_4}$: Use second order backward functions.
- $\boldsymbol{F_5}$: Use all seven types of feature functions listed earlier.

$\boldsymbol{F_1}$ is the plain *multiclass* case and corresponds to treating the tagging problem purely using token properties without using any sequence information. With this choice of features a CRF becomes a maximum entropy model and SVM-struct becomes the Crammer-Singer multiclass SVM model (Crammer and Singer, 2001). The performances of the methods with $\boldsymbol{F_1}$ form a good baseline. On datasets having informative sequential structure the methods are expected to do significantly better than this baseline performance when sequence related second order features are included.

Nguyen and Guo, 2007 mention $\boldsymbol{F_2}$ in equation (1) of their paper. This is the set of features used by Herbst and Joachims, 2007 in their SVM-struct code.

Mallet (McCallum, 2002) seems to use $\boldsymbol{F_3}$ only.

3. Experiments

We use three datasets: POS, OCR and CORA. The first two datasets are exactly as used in Nguyen and Guo, 2007. CORA is the *Cora Information Extraction* dataset taken from *http://www.cs.umass.edu/˜mccallum/code-data.html.* For this dataset token features of the types described in Peng and McCallum, 2004 are used. Each dataset is partitioned randomly into three parts for training, validation and testing. For each (method, dataset) combination w is obtained by solving (1) on the training set and tuning the regularization constant λ using the validation set. Then, at the best λ, w is again obtained by training it on the combined training and validation sets and the resulting model is tested on the testing set. As in Nguyen and Guo, 2007, *Average Loss* (the mean, over examples, of the fraction of wrongly labeled tokens in each example) is used for measuring performance. For POSB and OCR we used exactly the same (training, validation, testing) partitions used by Nguyen and Guo. For POSB Nguyen and Guo used only one random partition, but tried various training data sizes: 500, 1000, 2000, 4000 and 8000. For the sake of doing quick experiments we used only the first three training set sizes. For OCR, all ten partitions used by Nguyen and Guo are used. For CORA we formed our own ten random partitions, each having (training, validation, testing) sizes of (320, 80, 100). For OCR and CORA which have ten partitions the reported results are the mean values of *Average Loss* over the ten partitions.

For each dataset and method (i.e. CRF and SVM-struct) we try $\boldsymbol{F_1}$, $\boldsymbol{F_2}$, $\boldsymbol{F_3}$, $\boldsymbol{F_4}$ and $\boldsymbol{F_5}$. One could also use the validation set to choose the right feature combination. It turns out that this is very effective for all three datasets. In fact, the best choice of feature combination chosen via the validation set correlated perfectly with the choice that has the best test set performance.

Table 1 compares CRF and SVM-Struct on the POS dataset when different combinations of features are used. The nearly best performance of $\boldsymbol{F_1}$ shows that sequence information is only very mildly useful for this

Table 1. Average Loss of algorithms on POS dataset in %

Feature set	Algorithm	Train Size 500	1000	2000
$\boldsymbol{F_1}$	SVM-Struct	8.79	7.28	5.93
	CRF	9.05	7.17	6.09
$\boldsymbol{F_2}$	SVM-Struct	8.38	7.15	5.63
	CRF	8.84	7.08	5.83
$\boldsymbol{F_3}$	SVM-Struct	13.86	10.62	8.23
	CRF	16.45	12.36	9.66
$\boldsymbol{F_4}$	SVM-Struct	13.54	10.18	8.15
	CRF	16.96	12.25	9.33
$\boldsymbol{F_5}$	SVM-Struct	9.89	7.73	6.14
	CRF	12.25	8.73	6.87

Table 2. Average Loss of algorithms on OCR dataset

Feature set	Algorithm	Mean average loss on 10 runs
$\boldsymbol{F_1}$	SVM-Struct	0.2667
	CRF	0.2790
$\boldsymbol{F_2}$	SVM-Struct	0.1924
	CRF	0.1997
$\boldsymbol{F_3}$	SVM-Struct	0.3923
	CRF	0.4092
$\boldsymbol{F_4}$	SVM-Struct	0.3769
	CRF	0.3978
$\boldsymbol{F_5}$	SVM-Struct	0.1957
	CRF	0.2113

POS task. Inclusion of *token-independent second order* features, i.e. $\boldsymbol{F_2}$, enhances the performance a little bit. $\boldsymbol{F_2}$ gives the best performance for both, CRF and SVM-struct and the peak performances of these methods are close. The other detailed features seem unhelpful. Since the Mallet software uses $\boldsymbol{F_3}$ this explains why it had poor performance in the comparison done by Nguyen and Guo. Though, with the best choice of features ($\boldsymbol{F_2}$) the performances of SVM-struct and CRF are close, SVM-struct seems to be less affected when a wrong set of features are used.

On the OCR dataset (see Table 2), again $\boldsymbol{F_2}$ gives the best performance for both methods and their peak performances are close. Even for other $\boldsymbol{F_i}$, the performances of CRF and SVM-struct are well-matched. SVM-struct with $\boldsymbol{F_1}$ and SVM-struct with $\boldsymbol{F_2}$ are slightly better than their counterparts, i.e. *SVM-Multiclass* and *SVM-HMM* of Nguyen and Guo, 2007. However, for some reason, CRF with $\boldsymbol{F_3}$ is much worse than the *CRF (Mallet)* result reported in Nguyen and Guo, 2007.

The CORA dataset seems to make the best use of sequential information of the tokens. Table 3 gives the results. $\boldsymbol{F_5}$ that uses all features gives the best performance for both CRF and SVM-struct. The peak performances of these methods are again close. SVM-struct seems to make much better use of $\boldsymbol{F_2}$ than CRF. We need to investigate why this is so.

Overall, it is clear that, if the proper set of feature functions are chosen via validation, then the performances of CRF and SVM-struct are quite close. This is the main take-home point of this short report.

Table 3. Average Loss of algorithms on CORA dataset

Feature set	Algorithm	Mean average loss on 10 runs
$\boldsymbol{F_1}$	SVM-Struct	0.4037
	CRF	0.3673
$\boldsymbol{F_2}$	SVM-Struct	0.0810
	CRF	0.1170
$\boldsymbol{F_3}$	SVM-Struct	0.0869
	CRF	0.0860
$\boldsymbol{F_4}$	SVM-Struct	0.1072
	CRF	0.1151
$\boldsymbol{F_5}$	SVM-Struct	0.0514
	CRF	0.0542

References

Crammer, K., and Singer, Y. (2001) On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2, 265-292.

Herbst, E., and Joachims, T. (2007) SVM^{hmm} Sequence tagging with structural support vector machines. *http://svmlight.joachims.org/*.

McCallum, A. (2002) Mallet: A machine learning for language kit. *http://mallet.cs.umass.edu*.

Nguyen, N., and Guo, Y. (2007) Comparisons of sequence labeling algorithms and extensions. *International Conference on Machine Learning*.

Peng, F., and McCallum, A. (2004) Accurate Information Extraction from Research Papers using Conditional Random Fields. *HLT-NAACL*.

Sutton, C., and McCallum, A. (2006) An Introduction to Conditional Random Fields for Relational Learning. In *Introduction to Statistical Relational Learning*, Eds. L. Getoor and B. Taskar, MIT Press.

Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y. (2005) Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research, 6*, 1453-1484.