Multitask Learning for Sequence Labeling Tasks

Anonymous

Affiliation

Abstract. In this paper, we present a learning method for sequence labeling tasks in which each example sequence has multiple label sequences. Our method learns multiple models, one model for each label sequence. Each model computes the joint probability of all label sequences given the example sequence. Although each model considers all label sequences, its primary focus is only one label sequence, and therefore, each model becomes a task-specific model, for the task belonging to that primary label. Such multiple models are learned *simultaneously* by facilitating the learning transfer among models through *explicit parameter sharing.* We experiment the proposed method on two applications and show that our method significantly outperforms the state-of-the-art method.

Keywords: multitask learning, multilabel learning, label dependency

1 Introduction

Sequence labeling is an important task that finds applications in many areas such as bio-informatics, natural language processing (NLP), speech recognition, image processing etc. Depending upon the underlying sequence labeling task, labels are assigned to the tokens present in the sequence. Often in many domains, multiple labeling tasks need to be specified for the same sequence, i.e., multiple task specific labels are assigned to each token in the same sequence. For example, in NLP domain, words in a sentence can be labeled with their Part of Speech (POS) tags as well as the phrase chunks [13]. In another domain, a customer-care center might be interested in labeling the textual conversations between a customer and customer-care agent with *resolution status of a product specific issue* as well as the semantic tone of conversations, a.k.a. *dialogue act* [10].

Multiple tasks formed on one sequence, typically, tend to have intrinsic inter-label correlations. For instance, in customer-care domain, customers typically have complaint in their tone while describing their issue with a certain product. Incidentally, issues status `open` tends to have correlation with dialogue class `COMPLAINT`. In such a setting where each token in a sequence has multiple labels and these multiple labels exhibit correlations, it is important that the learning algorithm takes advantage of these correlations. If we define a task as learning from pairs of example sequence and its corresponding label sequence, then we can cast learning multiple label sequences as multitask sequence labeling learning problem. In machine learning, Multitask Learning(MTL) is a well known problem to learn various related tasks simultaneously [7,15,14,4,9,5]. In MTL, most of the work has focused on classification or regression problems, with very little work on sequence labeling problem. In addition, most of the MTL methods are not especially designed for *our* multitask setting, i.e., an example sequence has multiple label sequences. Any method especially designed for multiple label sequences setting should exploit the dependencies among labels. One recent work that exploits the label dependencies is the work of [12]. In this work, authors build a model called factorial CRF (appropriate for sequence labeling tasks) that is a *combined* CRF model *implicitly* learned on multiple tasks. In contrast to this method, our proposed method exploits the correlations present in multiple label sequences *explicitly* that not only improves upon the factorial CRF but also leads to a flexible framework for multitask sequence learning.

In this work, we extend the MTL setting to the sequence labeling problem with multiple label sequences. Our method — based on CRFs — not only exploits label dependencies but also learns multiple tasks simultaneously by *explicitly* sharing parameters. In our method, we learn one model for each task[1]. Each task has two factors (as opposed to one factor in CRFs), one factor corresponding to *all* labels (we call it *label dependency factor*), and other factor corresponding to task-specific *primary* label (we call it *task-specific factor*). Since the factor corresponding to *all* labels appear in all tasks, we facilitate the learning transfer among tasks by keeping the parameters corresponding to this factor same across all tasks. We show through a variety of experiments on two different data sets that such a model outperforms the state-of-the-art method. Note that learning from multiple labels is typically done in two ways: (1) build one single model that incorporates factors of all label sequences and example sequence, i.e., complete dependency and no independent learning (2) build multiple CRF-like *independent* models with no learning transfer among models, i.e., complete independent learning, and no dependency among labels. Our proposed method is a middle ground between these two extremes, and provides the best of both worlds. Because of a task-specific factor, it allows model to learn independently, and at the same time, because of label-dependency factor, it allows learning to transferred among all tasks.

In this work, we also propose a variation of this method. This variation allows one to control the amount of transfer among multiple tasks. Experimental results of this variation show further improvement. One of the main advantages of MTL is its relatively less reliance on task specific parameter tuning (rather draws benefit from other tasks), making it robust and applicable to wider set of applications. In our experiment, we spend little to no efforts on hyper-parameter tuning, and our the main improvement comes from the MTL paradigm, a natural method for learning.

[1] A task definition is expanded to include all labels, however, each task has one *primary* label sequence, and other label sequences are considered *secondary.*

2 Background and Problem Description

We first define CRFs. CRFs [6] are undirected graphical models that model the conditional probability of a label sequence given an observed example sequence. Let $\mathcal{G}$ be an undirected graphical model over random variable sequences $\mathbf{x}$ and $\mathbf{y}$, i.e. $\mathbf{x} = (x_1, x_2, \dots x_T)$ is the sequence of observed entities (e.g. words in a sentence) that we want to label with $\mathbf{y} = (y_1, y_2, \dots y_T)$. $(\mathbf{x}, \mathbf{y})$. In the undirected graph $\mathcal{G}$, let $\mathcal{C} = \{C_1, C_2 \dots\}$ be the set of cliques contained in the graph, then, given such a graph defined on example-label pair, the conditional probability $p(\mathbf{y}|\mathbf{x})$ is given by $p(\mathbf{y}|\mathbf{x}, \theta) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \Phi(\mathbf{y}_c, \mathbf{x}_c|\theta)$, where Φ is the potential function defined over a clique. For example, in a specific case of linear chain CRF, these potential functions are defined over cliques (x_t, y_{t-1}, y_t). Here θ is the parameter and $Z(\mathbf{x})$ is the normalization function.

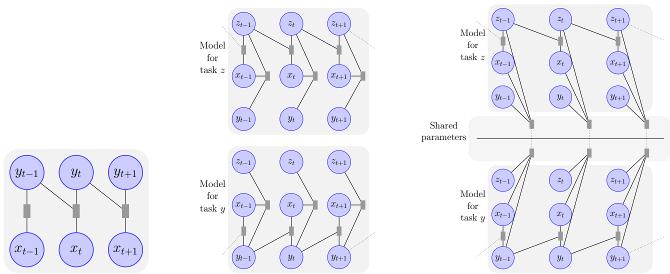
In multitask sequence labeling problem, we are given multiple label sequences for each example sequence, i.e., in addition to $\mathbf{y} = (y_1, y_2, \dots y_T)$ (as defined for CRFs), we have $\mathbf{z} = (z_1, z_2, \dots z_T)$ as another set of label sequence for $\mathbf{x}$. For simplicity, we only consider two types of label sequences, however, it is straightforward to extend our approach to more than two labeling sequences (see Definition 1). Thus our training examples for the entire task become triplets of $\{\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}_{i=1}^n$. Therefore, the multiple sequence labeling problem can be formalized as modeling conditional density $p(\mathbf{y}, \mathbf{z}|\mathbf{x})$.

3 Our Approach

In this section, we first describe a basic approach. In our basic but novel approach, we begin by providing a middle ground between two extremes (i.e. one single fully dependent model [12,8] and two fully independent CRF-like models), where both tasks are modeled independently but at the same time, one task draws benefit from other task through label dependencies. We model $p(\mathbf{y}, \mathbf{z}|\mathbf{x})$ by considering two types of cliques: task-specific clique i.e. $\Phi(y_{t-1}, y_t, x_t)$ and common clique $\Phi(y_t, z_t, x_t)$. Here task-specific clique provides the independence while the common clique provides the benefit from other labels. As we shall see later, such a model provides *better discriminating power* than the models that consider all types of cliques [12,8]. Given such two types of cliques, the conditional probability of both label sequences given the example sequence can be written as:

$$p^y(\mathbf{y}, \mathbf{z}|\mathbf{x}, \theta^y, \psi^y) = \frac{1}{U^y(\mathbf{x})} \prod_{t=1}^{T} \Big(\underbrace{\Phi(y_{t-1}, y_t, x_t|\theta^y)}_{\text{task(y) factor}} \Big) \Big(\underbrace{\Phi(y_t, z_t, x_t|\psi^y)}_{\substack{\text{label dependency} \\ \text{factor}}} \Big) \quad (1)$$

Although (1) provides the probability of both the labels, i.e., $(\mathbf{y}, \mathbf{z})$, conditioned on $\mathbf{x}$, there is no clique that depends on adjacent z labels, i.e., z_t, z_{t-1}. Thus though incorporating partial information from other label z, the above model still focuses on the task y. One can define a similar model for task z by replacing y with z. Note that in the above models each type of clique has its own separate set of parameters, i.e. task y has its parameters θ^y and ψ^y and the task z has its own parameters θ^z and ψ^z. We call this model UNSHARED model. A pictorial representation of this UNSHARED model is shown in Figure 1. The above models can be optimized (and inferenced) using the standard machinery used in CRF since these models are exactly the same as CRF except an additional clique.



(a) CRF (b) UNSHARED (c) SHARED

Fig. 1: Graphical model representations of CRF, UNSHARED and SHARED models. Note the common factors in the SHARED model, above and below the horizontal line. These factors are defined over the same random variables and share the parameters. Also note that the shared version has two separate graphical models with shared parameters and should not be confused with one model.

Below we define the generalized UNSHARED multilabel model, i.e., there can be any number of labels with arbitrary dependencies among them.

Definition 1 *Let* $\mathbf{x}$ *be an observed example sequence with* $\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_k$ *its multiple label sequences. Let* $\mathcal{C}_t$ *be the set of cliques denoting the possible interactions among labels at time* t *(i.e., interaction among labels* $\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_k$*), then, the* UNSHARED *multilabel model is a set of task-specific models where each task-specific model (for task* $\mathbf{y}_l$*) is defined as:*

$$p^{y_l}(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k | \mathbf{x}, \theta^{y_l}, \psi^{y_l}) = \frac{1}{Z(\mathbf{x})} \Big(\prod_{t=1}^{T} \Phi(y_{l,t-1}, y_{l,t}, x_t | \theta^{y_l}) \Big) \Big(\prod_{t=1}^{T} \prod_{c \in \mathcal{C}_t} \Phi(\mathbf{y}_c, x_t | \psi^{y_l}) \Big)$$

3.1 Shared Models

Although more accurate than the existing methods (CRF and factorial CRF) (see experiments), this method does not take advantage of the multitask nature of the problem, as both models have their own separate set of parameters, and there is no learning transfer between these models. We exploit the multitask nature of the problem and facilitate learning transfer by sharing the parameters corresponding to the common clique in both models. Sharing parameters to facilitate learning transfer is a well-known practice in multitask learning [9,5,2,4,3]. In other words, we make $\psi^y = \psi^z = \psi$. We call this formulation SHARED model. A pictorial representation of this SHARED model is shown in Figure 1. Now for the clarity and follow up discussion, we write the formulation for task y (similar for task z) in terms of corresponding feature functions (under SHARED model):

$$p^y(\mathbf{y}, \mathbf{z}|\mathbf{x}, \theta^y, \psi) = \frac{1}{U^y(\mathbf{x})} \prod_{t=1}^{T} \exp\Big(\sum_{k} \Big(\theta^y_k f_k(y_{t-1}, y_t, x_t) + \psi_k f_k(y_t, z_t, x_t) \Big) \Big). \quad (2)$$

Next we construct our objective function to fit data to these models. We take four specific approaches to define objective function as described below.

Joint Optimization: We hypothesize that although each of these models are sufficient to learn the labels for both tasks independently, it will be advantageous to learn them simultaneously. Consequently, we define a joint model that is the product of both models We maximize the likelihood of the data under this model, i.e., find the parameters by optimizing the joint log likelihood. This is equivalent to minimizing *cumulative* loss of both models on the training data. To reduce the overfitting, we use Gaussian prior with zero mean and unit variance on all parameters. The log likelihood of the data with this modeling approach can be written as: $\ell(\theta^y, \theta^z, \psi) = \sum_{i=1}^{n} \log p^y \left(\mathbf{y}^{(i)}, \mathbf{z}^{(i)}|\mathbf{x}^{(i)}, \theta^y, \psi\right) + \log p^z \left(\mathbf{y}^{(i)}, \mathbf{z}^{(i)}|\mathbf{x}^{(i)}, \theta^z, \psi\right) - \frac{\eta^y}{2}\|\theta^y\|^2 - \frac{\eta^z}{2}\|\theta^z\|^2 - \frac{\eta^o}{2}\|\psi\|^2$. Note that the joint likelihood function $\ell(\theta^y, \theta^z, \psi)$ is convex in all its parameters i.e. θ^y, θ^z, and ψ; and hence can be optimized by a number of techniques. In our implementation, we use L-BFGS for parameter optimization, and belief propagation and Viterbi for inferences.

3.2 Variance Models

We also propose a variation of the above model where the shared parameters is further broken into two parts: one task specific while other common. We hypothesize that the whole label dependency factor may not be common to both tasks, but only a part of it. As we shall see shortly that it will bring flexibility in the model, allowing one to control the amount of transfer among different tasks. Along the lines of [9], we believe that the parameters corresponding to the label dependency factor lie around a common set of parameters having their own variance specific to task. With this assumption, the common set of parameters ψ can be written as: $\psi^y = \psi^o + \nu^y$. Now, ψ^o is the part that is common to all tasks while ν^y is the task specific part. This is to indicate that there might be a component of ψ that is only specific to that task when considering parameters ψ. The log likelihood under this modeling paradigm with proper prior on each parameter can be written as: $\ell_y(\theta^y, \nu^y, \psi^o) = \sum_{i=1}^{n} \log p^y(\mathbf{y}^{(i)}, \mathbf{z}^{(i)}|\mathbf{x}^{(i)}, \theta^y, \nu^y, \psi^o) - \frac{\eta^y}{2}\|\theta^y\|^2 - \frac{\lambda}{2}\|\nu^y\|^2 - \frac{\eta^o}{2}\|\psi^o\|^2$. We emphasize here the importance of the factor λ/η^o which acts as a interpolating factor interpolating between a unshared model and a completely shared model. Note that $\lambda/\eta^o \to 0 \implies \psi^o \to 0$ while $\lambda/\eta^o \to \infty$ forces θ^y and ν^y to go to zero. Under this variance model, similar to the previous models, there can be two ways to model the data likelihood: one is jointly and other alternative.

4 Experiments

In this section, we describe the datasets, our experimental methodology, and report results.

4.1 Dataset

We evaluate and report our results on two datasets. The first dataset comes from an electronic conversation medium over social media (twitter). The example set is borrowed from real conversations (chat) between customers and customer care agents for a particular telecommunication carrier. Two specific tasks are designed in this case where the chat sentences are labeled for (1) nature of dialogue between customer and agent (namely *Dialogue Act*), and (2) nature of the state of the issue being discussed by customer and agent (namely *Issue Status*). We employed 3 annotators for labeling each sentence present in the conversations. Each conversation is treated as a sequence example akin to a sentence in the first dataset. For first task, sentences are annotated from 12 label such as Complaint, answer, acknowledgement, response, apology etc. For second task, sentences are annotated with 4 labels: Open Issue, Issue Resolved, Change Medium of Communication, and Issue Closed. We take 291 annotated conversations with a total of 3072 sentences with 10.6 sentences per conversation. We append frequent bigrams, emoticons, punctuation and standard word features such as capitalization etc.

In order to show the effectiveness of our method beyond issue-status and dialogue act prediction problems, we also experiment with a second dataset. This second dataset corresponds to a noun phrase chunking and POS tagging tasks, and comes from a CoNLL 2000 shared-task [2]. We take a smaller set of the original data set primarily because MTL only makes sense when single task learning (STL) is not sufficient (i.e. it is difficult). This difficulty of STL can be attributed to two main reasons– one, there are not enough labeled examples, and second, the problem itself is a difficult problem despite being enough labeled examples. The CoNLL dataset violates both of these conditions, i.e., there are enough labeled examples, and these labeled examples give a very good accuracy i.e., in the range of 99%. So in order to make the MTL applicable here, we increase the difficulty of the problem by reducing the size of labeled data. The smaller dataset consists of total 350 sentences containing 8785 individual tokens as examples. We split the data into 150 train and 200 test examples. In this dataset, two tasks correspond to the NP chunking and part-of-speech (POS) tagging. The idea is to get performance improvement by learning from these two tasks simultaneously. This dataset is also used in the baseline method by Sutton et al., [12]. For the sake of completeness, we also ran our experiments on full dataset, and all methods performed between 98% and 99%.

[2] Publicly available at [13] `http://mallet.cs.umass.edu/grmm/data`

4.2 Models Comparisons

We use following models for comparisons. Among these models, one is baseline, other models are ours, with different variations.

- **Factorial CRF[12]:** We use this as our primary baseline.
- **Unshared model:** Both tasks have their own separate parameters (See Definition 1).
- **JOSP:** (Jointly Optimized Shared Parameters) This is the shared model where parameters are learned by optimizing the joint likelihood.
- **AOSP:** (Alternatively Optimized Shared Parameters) This is the shared model but in contrast to the joint optimization, here parameter are learned in an alternative fashion, i.e., we split the joint likelihood into two parts, one for each task and optimize the parameters alternatively. ψ is still a common set of parameters among both tasks however we do not optimize the joint likelihood.
- **JOVM:** (Jointly Optimized Variance Model) Variance model as defined in Section 3.2 but parameters are learned by optimizing the joint likelihood.
- **AOVM:** (Alternatively Optimized Variance Model) Variance model as defined in Section 3.2 but parameters are learned alternatively.

4.3 Results

We use accuracy as our metric of evaluation. Here we define accuracy as fraction of correctly labeled tokens in sequences present in the test set. It is important to note that we report the accuracy from their respective models i.e., each model gives labels for all tasks but we take the labels from the model that is specific to that task (as described in Section 3.1). The results for the two datasets are presented in Table 1. We vary the training size and report the results. All reported results are averaged over 10 random runs, and their means and standard deviations are reported. For the baseline, we use the code provided by the authors. All the hyper-parameters are tuned via cross validation with 10 folds.

From these results we draw multiple conclusions: (1) In general, learning tasks together in MTL setting —either directly or using variance method— helps. All results show significant improvement over factorial CRF. This improvement is higher when there are fewer labeled examples. (2) Though in some cases, MTL (Shared model and Variance model) helps over factorial CRF but learning them independently (Unshared model) helps even more. e.g. Issue Status task. This establishes the fact that not all tasks improve from MTL. In fact, it shows that in multiple tasks, one task can benefit from other tasks while another cannot.

From the accuracy figures, it can be inferred that the Task 1 is harder than Task 2 for both datasets. The results reported show that the accuracy improvements are greater for Task 1 compared to Task 2. For difficult tasks, results show that learning both tasks independently (Unshared model) hurts. Learning them together through explicit parameter sharing gives significant improvement over Unshared or factorial CRF. This observation along with the observation that

MTL improvement is higher when there are fewer labeled examples, provide evidence in support of the hypothesis about the applicability of MTL, i.e., MTL is applicable when the underlying problem is difficult, either inherently or because of the scarcity of labeled examples. The results are not as clear for Task 2, but still, in these tasks, results indicate that one should use MTL – either learn all tasks together through *explicit* parameter sharing (Shared model or Variance model) or not share anything at all (UNSHARED MODEL). Partial sharing (one task structure) as in factorial CRF gives inferior results.

Table 1: Experimental results for MTL for CoNLL and Social Conversations datasets

Dataset	Task	%Train	MTL					DCRF
			JOVM	AOVM	JOSP	AOSP	Unshared	
CoNLL	POS Tagging	(30%)	86.0 ± 1.3	**86.3 ± 1.2**	83.7 ± 1.6	84.1 ± 1.4	77.9 ± 1.2	81.6 ± 1.4
	(Task 1)	(60%)	91.5 ± 0.5	**91.6 ± 0.4**	90.7 ± 0.5	90.8 ± 0.6	85.7 ± 0.4	88.2 ± 0.5
	NP Chunking	(30%)	**89.0 ± 0.4**	88.8 ± 0.9	88.5 ± 1.1	88.7 ± 0.9	88.8 ± 0.9	87.5 ± 0.8
	(Task 2)	(60%)	91.5 ± 0.5	**91.6 ± 0.3**	91.3 ± 0.5	91.4 ± 0.3	91.5 ± 0.4	90.7 ± 0.4
Social Conversation	Dialogue Act	(30%)	**51.4 ± 2**	50.7 ± 1.4	45.3 ± 2	50.5 ± 2	45.6 ± 2.0	48.9 ± 1.1
	(Task 1)	(60%)	56.7 ± 2.6	**56.9 ± 1.8**	55.7 ± 2.8	56.6 ± 1.6	52.1 ± 1.9	53.9 ± 1.2
	Issue Status	(30%)	**77.2 ± 0.9**	76.6 ± 0.8	74.4 ± 2.9	76.5 ± 1.1	**77.2 ± 1.1**	76.0 ± 1.4
	(Task 2)	(60%)	80.3 ± 1.1	80.5 ± 1.2	80.8 ± 1.5	80.0 ± 1.1	**80.9 ± 0.6**	79.4 ± 0.5

5 Conclusion

In this paper, we have presented a novel method for learning from multiple sequence labeling tasks. Unlike the previous methods, our method models each task as one single model, but still transfer the learning from other tasks through parameters sharing, thus finding the sweet spot between one single model and multiple independent models. We have shown through various experiments on two datasets that our method consistently outperforms the one of the best methods for such tasks, especially in cases when tasks are relatively harder and there are few labeled examples.

References

1. Agarwal, A., Iii, H.D., Gerber, S.: Learning multiple tasks using manifold regularization. In: Advances in neural information processing systems. pp. 46–54 (2010)
2. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: NIPS '06 (2006)
3. Argyriou, A., Evgeniou, T., Pontil, M., Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. In: Machine Learning. press (2007)
4. Argyriou, A., Micchelli, C.A., Pontil, M., Ying, Y.: A spectral regularization framework for multi-task structure learning. In: NIPS '08 (2008)
5. Jacob, L., Bach, F., Vert, J.P.: Clustered multi-task learning: A convex formulation. In: NIPS '08 (2008)
6. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)

7. Liu, Q., Liao, X., Carin, H.L., Stack, J.R., Carin, L.: Semisupervised multitask learning. IEEE 2009 (2009)
8. Marcheggiani, D., Täckström, O., Esuli, A., Sebastiani, F.: Hierarchical multi-label conditional random fields for aspect-oriented opinion mining. In: Advances in Information Retrieval, pp. 273–285. Springer (2014)
9. Micchelli, C.A., Pontil, M.: Regularized multi-task learning. In: KDD 2004. pp. 109–117 (2004)
10. Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Van Ess-Dykema, C., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., Meteer, M.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Comput. Linguist.
11. Sutton, C., McCallum, A.: Composition of conditional random fields for transfer learning. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 748–754. Association for Computational Linguistics (2005)
12. Sutton, C., McCallum, A., Rohanimanesh, K.: Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. The Journal of Machine Learning Research 8, 693–723 (2007)
13. Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the conll-2000 shared task: Chunking. In: Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7. pp. 127–132. Association for Computational Linguistics (2000)
14. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with dirichlet process priors. J. Mach. Learn. Res. 8, 35–63 (2007)
15. Yu, K., Tresp, V., Schwaighofer, A.: Learning gaussian processes from multiple tasks. In: ICML '05 (2005)