A Discriminative Latent Variable Model for Online Clustering

Rajhans Samdani, Google Research RAJHANS@GOOGLE.COM
Kai-Wei Chang, University of Illinois KCHANG10@ILLINOIS.EDU
Dan Roth, University of Illinois DANR@ILLINOIS.EDU

Abstract

This paper presents a latent variable structured prediction model for discriminative supervised clustering of items called the Latent Left-linking Model (L^3M). We present an online clustering algorithm for L^3M based on a feature-based item similarity function. We provide a learning framework for estimating the similarity function and present a fast stochastic gradient-based learning technique. In our experiments on coreference resolution and document clustering, L^3M outperforms several existing online as well as batch supervised clustering techniques.

1. Introduction

Many machine learning applications require clustering of items in an online fashion, e.g. detecting network intrusion attacks (Guha et al., 2003), detecting email spam (Haider et al., 2007), and identifying topical threads in text message streams (Shen et al., 2006). Many clustering techniques use pairwise similarities between items to drive a batch or an online algorithm. Learning the similarity function and performing online clustering are challenging tasks.

This paper addresses these challenges and presents a novel discriminative model for online clustering called the Latent Left-Linking Model (L^3M). L^3M assumes that for data items arriving in a given order, to cluster an item i, it is sufficient to consider only the previous items (i.e. items considered before i.) This assumption is suitable for many clustering applications, especially when the items arrive as a data stream. More specifically, L^3M is a feature-based probabilistic structured prediction model, where each item can *link* to a previous item with a certain probability. L^3M expresses the probability of an item joining a previously formed cluster as the sum of the probablities of multiple links connecting that item to the items inside that cluster. We present an efficient online inference (or clustering) procedure for L^3M. L^3M admits a latent variable learning framework, which we optimize using a fast online stochastic gradient technique.

We present experiments on coreference resolution and document clustering. Coreference resolution is a popular and challenging Natural Language Processing (NLP) task, that involves clustering denotative noun phrases in a document where two noun phrases are co-clustered if and only if they refer to the same entity. We consider document clustering as the task of clustering a collection of textual items (like emails or blog posts) based on criteria like common authorship or common topic.

We compare L^3M to supervised clustering techniques — some of these techniques are online (Haider et al., 2007; Bengtson & Roth, 2008) and some are batch algorithms that need to consider all the items together (Mccallum & Wellner, 2003; Finley & Joachims, 2005; Yu & Joachims, 2009). L^3M outperforms all the competing baselines. Interestingly, it outperforms batch clustering techniques (which are also computationally slower e.g. Correlation Clustering is NP hard (Bansal et al., 2002).) Consequently, we conduct further experiments to discern if L^3M is benefitting from better modeling or from exploiting a natural ordering of the items (e.g. noun phrases in a document.)

2. Notation and Pairwise Classifier

Notation: Let d be an item set i.e. a set of items to be clustered. Let m_d denote the number of items in d, e.g. in coreference, d is a document and m_d is the number of noun phrases in d. We refer to items using their indices, which range from 1 to m_d. A cluster c of items is a subset of $\{1, \ldots, m_d\}$. A *clustering* $\mathcal{C}$ for an item set d partitions the set of all items, $\{1, \ldots, m_d\}$, into disjoint clusters. However, instead of representing $\mathcal{C}$ as a set of subsets of $\{1, \ldots, m_d\}$, for notational convenience, we represent $\mathcal{C}$ as a binary function with $\mathcal{C}(i, j) = 1$ if items i and j are co-clustered in $\mathcal{C}$, otherwise 0.

Pairwise classifier: We use a pairwise scoring function indicating the compatibility or similarity of a pair of items as the basic building block for clustering. In particular, for any two items i and j, we produce a pairwise compatibility

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

score w_{ij} using features extracted from i and j, $\phi(i,j)$, as

$$w_{ij} = \mathbf{w} \cdot \phi(i,j) \ , \tag{1}$$

where $\mathbf{w}$ is a weight vector to be estimated during learning. The feature-set consists of different features indicative of the compatibility of items i and j. E.g. in document clustering, these features could be the cosine similarity, difference in time stamps of i and j, the set of commons words, etc. The pairwise approach is very popular for discriminative supervised clustering tasks like coreference resolution and email spam clustering (Mccallum & Wellner, 2003; Finley & Joachims, 2005; Haider et al., 2007; Bengtson & Roth, 2008; Yu & Joachims, 2009; Ng, 2010). Also notably, this pairwise feature-based formulation is more general and flexible than metric learning techniques (Xing et al., 2002) as it can express concepts (e.g. cosine similarity) that cannot be expressed using distances in a metric space.

3. Probabilistic Latent Left-linking Model

In this section, we describe our Latent Left-Linking Model (L^3M) for online clustering of items based on pairwise links between the items. First we will describe our modeling assumptions and the resulting probabilistic model, then we will elaborate on the underlying latent variables in our model, and then finally we will discuss the clustering (or inference) and learning algorithms.

3.1. L^3M: Model Specification and Discussion

Let us suppose that we are considering the items $1, \ldots, m_d$ in order. For intuitive illustration, assume that the items are streaming from right-to-left and item 1 is the leftmost item (i.e. is considered first.) To simplify the notation, we introduce a dummy item with index 0, which is to the left (i.e. appears before) of all the items and has $\phi(i,0) = \emptyset$, and consequently, similarity $w_{i0} = 0$ for all *actual* items $i > 0$. For a given clustering $\mathcal{C}$, if an item i is not co-clustered with any previous actual item $j, 0 < j < i$, then we assume that i links to 0 and $\mathcal{C}(i,0) = 1$. In other words, $\mathcal{C}(i,0) = 1$ *iff* i is the first actual item of a cluster in $\mathcal{C}$. However, such an item i is **not** considered to be co-clustered with 0 as that would incorrectly imply, by transitivity, that all the items $(1, \ldots, m_d)$ are co-clustered. In particular, for any valid clustering, item 0 is always in a singleton dummy cluster, which is eventually discarded.

3.1.1. Model Specification:

L^3M is specified by three simple modeling assumptions on probabilistic links between items:

1. **Left-Linking:** Each item i can only (probabilistically) link to an antecedent item j on its left (i.e. j occurs before i or $j < i$), thereby creating a *left-link*, $j \leftarrow i$.
2. **Independence of Left-links:** The event that item i links to an antecedent item j is independent of the event that any item $i', i' \neq i$, has a left-link to some item j'.
3. **Probabilistic Left-link:** For an item set d, the probability of an item $i \geq 1$ linking to an item j to its left $(0 \leq j < i)$, $P[j \leftarrow i; d, \mathbf{w}]$, is given by

$$Pr[j \leftarrow i; d, \mathbf{w}] = \frac{\exp\left(\frac{w_{ij}}{\gamma}\right)}{\sum\limits_{0 \leq k < i} \exp\left(\frac{w_{ik}}{\gamma}\right)} = \frac{\exp\left(\frac{w_{ij}}{\gamma}\right)}{Z_i(d, \mathbf{w}, \gamma)} \ , \tag{2}$$

where, recall that $w_{ij} = \mathbf{w} \cdot \phi(i,j)$ is the similarity between i and j, $Z_i(d, \mathbf{w}, \gamma) = \sum_{0 \leq k < i} \exp(\frac{w_{ik}}{\gamma})$ is the normalization and $\gamma \in (0, 1]$ is a tunable temperature parameter. In previous works (Pletscher et al., 2010; Schwing et al., 2012), use of temperature in discriminative models has been restricted to entropy reduction in learning. We, on the other hand, extend it to explicitly create a probabilistic model, which leads to a very general and flexible inference (or clustering) algorithm.

Note that our modeling assumptions have an obvious similarity to the Distance Dependent Chinese Restaurant Process (CRP) model for *generative* clustering (Blei & Frazier, 2011). However, L^3M, to the best of our knowledge, is the first *supervised discriminative* model to generalize this idea to the use of arbitrary pairwise features with learned weights. In Sec. 3.1.3, we will show that L^3M in fact is a latent variable structured prediction model.

3.1.2. Likelihood of a Clustering in L^3M

In this section, we compute the likelihood $Pr[\mathcal{C}; d, \mathbf{w}]$ of generating a clustering $\mathcal{C}$ for items in an item set d, given $\mathbf{w}$. This probability will shed more insight into our model and will also help in performing likelihood based learning. Due to the Assumptions 1 and 2, we can express $Pr[\mathcal{C}; d, \mathbf{w}]$ as the product of the probabilities of each item i connecting to its left in a manner consistent with $\mathcal{C}$: $Pr[\mathcal{C}; d, \mathbf{w}] = \prod_{i=1}^{m_d} Pr[\mathcal{C} \triangleleft i; d, \mathbf{w}]$, where $Pr[\mathcal{C} \triangleleft i; d, \mathbf{w}]$ is the probability that item $i \geq 1$ connects to its left as per $\mathcal{C}$ i.e. the probability that i links only to those antecedent items j that have $\mathcal{C}(i,j) = 1$. $Pr[\mathcal{C} \triangleleft i; d, \mathbf{w}]$ is simply given by the sum of probabilities of item i connecting to only those items before i that are co-clustered with i in $\mathcal{C}$:

$$Pr[\mathcal{C} \triangleleft i; d, \mathbf{w}] = \sum_{0 \leq j < i} Pr[j \leftarrow i; d, \mathbf{w}] \mathcal{C}(i,j) = \frac{Z_i(\mathcal{C}; d, \mathbf{w}, \gamma)}{Z_i(d, \mathbf{w}, \gamma)}; \tag{3}$$

$Z_i(\mathcal{C}; d, \mathbf{w}, \gamma) = \left(\sum_{0 \leq j < i} \exp\left(\frac{w_{ij}}{\gamma}\right) \mathcal{C}(i,j)\right)$ being the unnormalized measure of connecting as per clustering $\mathcal{C}$, and $Z_i(d, \mathbf{w}, \gamma)$ is defined in Eq. (2). Using (3), we obtain

the likelihood of clustering $\mathcal{C}$ as

$$Pr[\mathcal{C}; d, \mathbf{w}] = \prod_{i=1}^{m_d} Pr[\mathcal{C} \triangleleft i; d, \mathbf{w}] = \prod_{i=1}^{m_d} \frac{Z_i(\mathcal{C}; d, \mathbf{w}, \gamma)}{Z_i(d, \mathbf{w}, \gamma)} = \prod_{i=1}^{m_d} \frac{\left(\sum_{0 \le j < i} \exp\left(\frac{w_{ij}}{\gamma}\right) \mathcal{C}(i,j)\right)}{\left(\sum_{0 \le j < i} \exp\left(\frac{w_{ij}}{\gamma}\right)\right)} \quad . \tag{4}$$

3.1.3. L^{3}M AS A LATENT-VARIABLE STRUCTURED PREDICTION MODEL

In this section, we present an alternative way of explaining L^{3}M, which exposes the underlying latent variables. We consider a special tree structure over items in item set d, which we call a *Left-Linking Tree*. A left-linking tree is a tree connecting items $1, \ldots, m_d$, where the parent of each item is on its left (i.e. considered before) in the item set. More formally, a valid left-linking tree for item set d can be represented as a set of edges $\mathbf{z} = \{(i,j) | 0 \le j < i \le m_d\}$, such that $\forall i \in \{1, \ldots, m_d\}$, $\exists$ a unique $j \in \{0, \ldots, i-1\}$ (to the left of i) such that $(i,j) \in \mathbf{z}$ and $\nexists k \in \{i, \ldots, m_d\}, (i,k) \in \mathbf{z}$. Trivially, a left-linking tree is always rooted at the dummy item 0.

For an item set d, let $\mathcal{Z}_d$ represent the set of all valid left-linking trees. We define a probability distribution over these trees, where the probability of a left-linking tree $\mathbf{z}$ is given by a Gibbs distribution based on the sum of the weights of edges in that tree: $Pr[\mathbf{z}; d, \mathbf{w}] = (\frac{1}{T(d,w,\gamma)} \exp(\frac{1}{\gamma}(\sum_{(i,j)\in\mathbf{z}} w_{ij})))$, where $T(d,w,\gamma) = \sum_{\mathbf{z}\in\mathcal{Z}_d} \exp(\frac{1}{\gamma}(\sum_{(i,j)\in\mathbf{z}} w_{ij}))$ is the partition function.

Lets assume that a left-linking tree is a latent underlying link structure between the items such that the clustering we observe is a result of taking the transitive closure of the subtrees rooted at the dummy item 0. Thus, trivially, a given left-linking tree results in a unique clustering. However, a given clustering can have multiple consistent left-linking trees as many left-linking trees can result in the same clustering after taking transitive closure. Given a clustering $\mathcal{C}$, let $\mathcal{Z}_d^{\mathcal{C}} = \{\mathbf{z} \in \mathcal{Z}_d | \forall (i,j) \in \mathbf{z}, \mathcal{C}(i,j) = 1\}$ refer to the set of all left-linking trees consistent with $\mathcal{C}$. Now consider the following model where we express the probability of a clustering $\mathcal{C}$ — the variable of interest — as the sum of the probabilities of all left-linking trees (the latent variables) consistent with $\mathcal{C}$:

$$Pr'[\mathcal{C}; d, \mathbf{w}] = \sum_{\mathbf{z}\in\mathcal{Z}_d^{\mathcal{C}}} Pr[\mathbf{z}; d, \mathbf{w}] \tag{5}$$

$$= \frac{1}{T(d,w,\gamma)} \sum_{\mathbf{z}\in\mathcal{Z}_d^{\mathcal{C}}} \exp\left(\frac{1}{\gamma}\left(\sum_{(i,j)\in\mathbf{z}} w_{ij}\right)\right) \quad .$$

The following theorem shows that the above model is exactly the same as the L^{3}M model:

Theorem 1 *The probability of a clustering expressed in Eq.* (5) *is the same as probability of clustering for L^{3}M as expressed in Eq.* (4)*, i.e.* $Pr'[\mathcal{C}; d, \mathbf{w}] = Pr[\mathcal{C}; d, \mathbf{w}]$.

The proof is presented in the supplement. This implies that L^{3}M indeed is a latent variable structured prediction model that marginalizes the left-linking trees as latent variables.

3.2. Approximate Online Clustering in the Latent Left-Linking Model

The goal of clustering or inference in L^{3}M is to cluster a set of items, given $\mathbf{w}$. We present a greedy online clustering algorithm, where each new item either joins an existing cluster or starts a new cluster. The probability that item i joins a previously formed cluster c, $Pr[c \odot i; d, \mathbf{w}]$, is simply the sum of the probabilities of i linking to the items inside c:

$$Pr[c \odot i; d, \mathbf{w}] = \sum_{j\in c, 0\le j<i} Pr[j \leftarrow i; d, \mathbf{w}] = \sum_{j\in c, 0\le j<i} \frac{\exp\left(\frac{1}{\gamma}\left(\mathbf{w} \cdot \phi(i,j)\right)\right)}{Z_i(d, \mathbf{w}, \gamma)} \quad . \tag{6}$$

Based on Eq. (6), we follow an online clustering algorithm: as each item i arrives, sequentially add it to a previously formed cluster $\arg\max_c Pr[c \odot i; d, \mathbf{w}]$. If the $\arg\max$ cluster is the singleton cluster with the dummy item 0 (and unnormalized measure 1), then i starts a new cluster (and is not included in the dummy cluster.) The greedy approach is not exact i.e. there exist cases where this algorithm does not give the most probable clustering (as per Eq. (4).) However, the sequential nature of this algorithm is suitable for online clustering and it works very well empirically.

The Case of $\gamma = 0$: Noting that l_p norm approaches the max norm as $p \to \infty$, as γ approaches zero, the probability $P[j \leftarrow i; d, w]$ in Eq. (2) in the limit approaches a Kronecker delta function that assigns probability 1 to the *max-scoring item* $j = \arg\max_{0\le k<i} w_{ik}$ (assuming no ties), and 0 to other items else (Pletscher et al., 2010; Samdani et al., 2012). Similarly, as $\gamma \to 0$, $Pr[c \odot i; d, \mathbf{w}]$ in Eq. (6) approaches a Kronecker delta function centered on the cluster containing the max-scoring item. Thus, for the rest of the paper, we abuse the notation and use the expressions in Eq. (2), (5), and (6) for all $\gamma \in [0, 1]$, where for $\gamma = 0$, they are assumed to be replaced by the appropriate Kronecker delta distributions.

The resulting clustering procedure for $\gamma = 0$ can effectively consider only one left link (the max-scoring left-link) per item. Consequently, our online inference algorithm for $\gamma = 0$ becomes what we call the *Max-Left-Link* (Ng & Cardie, 2002) inference, where each item i connects to the item j on its left having the maximum weight link, and the final clustering is the result of taking the transitive closure

of such links (removing the links to the dummy item.) Alternatively, this implies that the clustering algorithm considers only the maximum weight left-linking tree in Eq. (5) rather than marginalizing over all left-linking trees.

Overall, L^3M is an expressive model that, by tuning γ, can express inference based on not only the maximum weight link, but, with the same time complexity (i.e. quadratic), inference based on multiple links between an item and a cluster. Also, note that previous works using Max-Left-Link inference (Ng & Cardie, 2002; Bengtson & Roth, 2008; Shen et al., 2006) often treat learning in an ad hoc fashion, without relating it to inference. L^3M presents a principled structured prediction view of learning and inference. In particular, for $\gamma = 0$, the learning algorithm for L^3M, presented next, is novel and experimentally superior.

3.3. Latent Variable Learning

The task of learning is to estimate $\mathbf{w}$, given a set of annotated or training item sets D, where for each item set $d \in D$, $\mathcal{C}_d$ refers to the true clustering.

Objective Function for Learning: Assuming the item sets, $d \in D$, are generated I.I.D., we learn $\mathbf{w}$ by minimizing regularized negative log-likelihood of the data. Using the latent tree representation (from Eq. (5)), this results in the following objective function $LL(\mathbf{w})$:

$$\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{\gamma}{|D|}\sum_{d\in D}\frac{1}{m_d}\Big(\overbrace{\sum_{\mathbf{z}\in\mathcal{Z}_d} e^{\frac{1}{\gamma}\left(\sum_{(i,j)\in\mathbf{z}}\mathbf{w}\cdot\phi(i,j)+\Delta(\mathbf{z},\mathcal{C}_d)\right)}}^{\text{loss-augmented partition function}} - \underbrace{\sum_{\mathbf{z}\in\mathcal{Z}_d^c} e^{\frac{1}{\gamma}\left(\sum_{(i,j)\in\mathbf{z}}\mathbf{w}\cdot\phi(i,j)\right)}}_{\text{unnormalized log probability of clustering}}\Big),$$

where λ is regularization penalty and $\Delta(\mathbf{z}, \mathcal{C}_d)$ measures the loss of a latent tree $\mathbf{z}$ against the true clustering $\mathcal{C}_d$. The technique of augmenting the partition function with the loss-based margin Δ is inspired by max-margin learning (Yu & Joachims, 2009). Pletscher et al. (2010) (also see Schwing et al. (2012)) show that by tuning γ, this formulation can generalize existing latent variable learning techniques. For $\gamma = 1$, $LL(\mathbf{w})$ is the objective function for hidden variable conditional random fields (HCRF) (Quattoni et al., 2007). As γ approaches zero, $LL(\mathbf{w})$ approaches latent structural SVMs (LSSVM) (Yu & Joachims, 2009). Thus by tuning γ, we consider a learning technique more general than LSSVM and HCRF.

For tractability, we use a decomposable loss function $\Delta = \sum_{(i,j)\in\mathbf{z}} 1-\mathcal{C}(i,j)$ that counts the edges in $\mathbf{z}$ that violate $\mathcal{C}$. Furthermore, with this loss function, leveraging the equivalence relation established by Theorem 1, we can rewrite the above objective function in the more tractable original L^3M likelihood formulation presented in Eq. (4) as

$$LL(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{\gamma}{|D|}\sum_{d\in D}\frac{1}{m_d}\sum_{i=1}^{m_d} \left(\log(\sum_{0\le j<i} e^{\frac{1}{\gamma}(\mathbf{w}\cdot\phi(i,j)+\delta(\mathcal{C}_d,i,j))}) - \log Z_i(\mathcal{C}_d; d, \mathbf{w}, \gamma)\right), \quad (7)$$

where $\delta(\mathcal{C}_d, i, j) = 1 - \mathcal{C}_d(i,j)$. Overall, the task of learning is to obtain $\mathbf{w}$ by minimizing $LL(\mathbf{w})$.

Stochastic (Sub)gradient based Optimization: The objective function in (7) is non-convex and hence is intractable to minimize exactly. With finite and relatively small-sized training item sets, one can use the Concave-Convex Procedure (CCCP) (Yuille & Rangarajan, 2003) which reaches a local minimum, but requires one to perform marginal inference over the entire set of items to compute the gradient. Such a technique will not work in an online setting or in cases when the number of items is large.

Observing that $LL(\mathbf{w})$ decomposes not only over training item sets, but also over individual items in each item set, we choose to follow a fast stochastic gradient descent (SGD) strategy that performs rapid online updates on a per-item basis. The stochastic gradient (subgradient when $\gamma = 0$) w.r.t. item i in item set d makes use of a weighted sum of features of all left-links from i and is given by

$$\nabla LL(\mathbf{w})^i_d \propto \sum_{0\le j<i} p_j\phi(i,j) - \sum_{0\le j<i} p'_j\phi(i,j) + \lambda\mathbf{w}, \quad (8)$$

where p_j and p'_j, $j = 0, \ldots, i-1$, are non-negative weights that sum to one and are given by

$$p_j = \frac{e^{\frac{1}{\gamma}(\mathbf{w}\cdot\phi(i,j)+\delta(\mathcal{C}_d,i,j))}}{\sum_{0\le k<i} e^{\frac{1}{\gamma}(\mathbf{w}\cdot\phi(i,k)+\delta(\mathcal{C}_d,i,k))}} \quad \text{and}$$

$$p'_j = \frac{\mathcal{C}_d(i,j)Z_i(\mathbf{w},\gamma)}{Z_i(\mathcal{C}_d,\mathbf{w},\gamma)} Pr[i \to j; d, \mathbf{w}] \ .$$

Intuitively, SGD with the gradient in Eq. (8) promotes a weighted sum of correct left-links from i and demotes a weighted sum of all other left-links from i. The reader should note that our algorithm is not SGD in a pure sense as the items are chosen in a fixed order and not randomly.

SGD is quite succesful and popular in practice when applied to many different non-convex learning problems (Guillory et al., 2009; LeCun et al., 1998)[1] despite being difficult to theoretically characterize for non-convex problems. In Sec. 5, we present extensive experiments which show that our SGD-based learning is robust and when compared with CCCP, converges rapidly without sacrificing empirical accuracy.

[1] See `http://leon.bottou.org/research/stochastic` for a fairly long list.

4. Related Work

Online or streaming data clustering using k-center approaches (Guha et al., 2003) over points in a fixed metric space has enjoyed much popularity in the data mining literature. However, our focus is on pairwise feature-based clustering which is more general than clustering points in a metric space (Xing et al., 2002) as pairwise similarity features (e.g. Jaccard similarity) are not restricted to be metrics. Also, we do not have to specify the number of clusters in advance. Rao et al. (2010) perform coreference clustering on a very large-scale, but use a hard-coded similarity function. Our work can be viewed as a supervised discriminative counterpart to the Distance Dependent Chinese Restaurant Process (Blei & Frazier, 2011) which performs unsupervised clustering of items arriving in an order.

L^3M is most closely related to other discriminative approaches that treat clustering as a structured prediction problem. We divide the discussion on these techniques into two groups: batch techniques that require looking at all the items together and online techniques that can be applied on one item at a time. We experimentally compare with these techniques in Sec. 5.

Batch Structured Prediction Techniques: The following two techniques require looking at all the items together and cannot be used for clustering in an online sense.

- **Correlational Clustering:** Mccallum & Wellner (2003) and Finley & Joachims (2005) perform inference using correlational clustering (Bansal et al., 2002) on a complete graph over all the items with the pairwise similarities as the edge weights. Since correlational clustering is NP Hard (Bansal et al., 2002), using exact inference in this approach is very slow for a large number of items.
- **Latent Spanning Forest (Yu & Joachims, 2009):** This approach posits that a given clustering is produced by taking the transitive closure of a latent spanning forest over the items. Inference in this case is equivalent to finding a maximum weight spanning forest connecting the items. Notably, L^3M also uses a tree structure spanning the items (the latent left-linking tree) as the underlying latent structure (Sec. 3.1.3.) However, the left-linking trees are a more restricted class of spanning trees — the left-linking restriction allows clustering to work in an online fashion and facilitates efficient summation over all left-linking trees. On the other hand, inference for Yu & Joachims (2009) is not online and they consider only the maximum weight forests rather than marginalizing over all latent forests. Furthermore, in our experimental applications, left-linking trees capture the directionality of the items and outperform the spanning forest model that do not have any directionality.

Online Techniques for Clustering We now discuss two techniques that cluster items in a greedy online order. Notably, search-based structured prediction techniques (Daumé III et al., 2009) cannot be used in the online setting as they require access to the entire item set to compute the loss associated with a greedy atomic action used to train a base classifier.

- **Sum-Link Decoding (Haider et al., 2007):** Sum-Link expresses the score of connecting an item i to a cluster c as the sum of the scores of pairwise links from i to all items in c: $\sum_{j \in c, j<i} \mathbf{w} \cdot \phi(i,j)$. Similar to L^3M, item i is greedily connected to the cluster with the highest score provided the score is greater than 0. However, there is a fundamental difference between L^3M and Sum-Link: Sum-Link combines all the links linearly whereas L^3M is a probabilistic log-sum-exponential model and puts significantly more importance on high scoring links than low scoring links. For several applications like coreference resolution, it is believed that only a few strong links and not all links, especially not the weak links, are likely to be informative (Ng & Cardie, 2002). For such cases, L^3M is much more suitable than Sum-Link. We show that L^3M significantly outperforms Sum-Link in our experiments.
- **Max-Left-Link with Binary Classifier:** As described in Sec. 3.2, in the Max-Left-Link strategy, each item connects according to only the maximum weight left-link (corresponds to $\gamma = 0$ in L^3M.) This strategy has been successfully used in applications like coreference clustering (Ng & Cardie, 2002; Bengtson & Roth, 2008) and thread detection (Shen et al., 2006). However, previous works perform learning in an ad hoc fashion by training a "link"/"do not link" binary — and not a structured prediction-based — classifier, without relating it to inference. L^3M not only generalizes the Max-Left-Link inference (by tuning γ), but also provides a more principled structured prediction framework, and experimentally outperforms such ad hoc techniques.

5. Experiments and Results

In this section, we present experiments on four datasets pertaining to two supervised clustering tasks: coreference resolution and document clustering. First, we discuss the competing algorithms and some experimental details.

Competing Algorithms: We compare with the following baselines. **Corr-Clustering:** This is a correlational clustering-based approach; following Finley & Joachims (2005), we use structural SVMs (Tsochantaridis et al., 2004) for learning. **Spanning Forest:** This is the latent spanning forest approach by Yu & Joachims (2009); we use the code provided by the authors. **Sum-Link:** This is an online clustering technique by Haider et al. (2007); we use stochastic gradient descent for learning. **Bin.-Left-Link:** Max-Left-Link inference with relatively ad hoc training

used by Bengtson & Roth (2008); in particular, we train $\mathbf{w}$ with an online SGD-based SVM on binary training data generated by taking for each item, the link to the closest antecedent co-clustered item as a positive example, and links to all other items in between as negative examples.

L^3M : we try two versions of our proposed L^3M approach. **L^3M (tuned γ):** In this version, we tune the value of γ using a validation set picking the best γ from $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. We use the same γ for training and testing. **L^3M ($\gamma = 0$):** In order to test whether considering multiple left-links help, we consider L^3M with γ set to 0 (which only uses the maximum weight left-link.)

For all the online clustering techniques (Sum-Link, Bin.-Left-Link, L^3M), we present results with a single pass over the data as well as with multiple number of passes tuned on a validation set. For all the algorithms, we tune the regularization parameters (and also γ for L^3M) to optimize the targeted evaluation metric on the development set. We use the same set of features for all the techniques.

5.1. English Coreference clustering

Coreference resolution is a challenging NLP task requiring a system to identify denotative noun phrases called *mentions* and clustering those mentions together that refer to the same underlying entity. In the following example, mentions with same subscript numbers are *coreferent*:

[American President]$_1$ [Bill Clinton]$_1$ has been invited by the [Russian President]$_2$, [Vladimir Putin]$_2$, to visit [Russia]$_3$. [President Clinton]$_1$ said [he]$_1$ looks forward to [his]$_1$ visit.

We argue that coreference clustering can be treated as an online data clustering problem as the mentions in documents follow a natural left-to-right order (right-to-left for a few languages.) This is motivated by the linguistic intuition that humans are likely to resolve coreference for a given mention based on antecedent mentions.

We show experimental results on two benchmark English coreference datasets — ACE 2004 (NIST, 2004) and Ontonotes-5.0 (Pradhan et al., 2012). ACE 2004 data contains 442 documents, split into 268 training, 68 development, and 106 testing documents — the same split is used across NLP literature as a benchmark (Bengtson & Roth, 2008) to compare various systems. OntoNotes-5.0 (Pradhan et al., 2012) is the largest annotated corpus on coreference with a total of 3,145 training documents and 348 testing documents. We use 343 documents from the training set for validation. Ontonotes contains documents drawn from different sources — newswire, bible, broadcast transcripts, magazine articles, and web blogs. We train and validate separate models for different parts of the corpus (like newswire or bible).

We use gold mention boundaries (i.e. mentions provided by the dataset) in our experiments in order to compare the algorithms purely on clustering, unmitigated by errors in mention detection. For all the techniques, we use a rich set of features provided by Chang et al. (2012). NLP literature evaluates coreference on primarily three different metrics — MUC (Vilain et al., 1995), B^3 (Bagga & Baldwin, 1998), and CEAF (Luo, 2005). We report F1 scores for these metrics and also their average, which we use as the main metric of comparison[2]. For inference in Corr-Clustering, we use an ILP solver.

Tab. 1 reports the results on coreference. Clearly, our L^3M approach outperforms all the competing baselines. We achieve state-of-the-art B^3 results on the ACE 2004 data[3]. For Ontonotes, we achieve performance close to the best result (with gold mentions) reported for this task (Pradhan et al., 2012) in terms of the average without the use of any additional domain knowledge. For all the settings with the exception of ACE with one pass, L^3M with tuned γ is better than L^3M with $\gamma = 0$ by 0.6-0.7 points in terms of the average showing that considering multiple links is beneficial. For L^3M (tuned γ), the best value of γ for ACE 2004 for one pass was 0; with multiple passes, the best γ was 0.2. For OntoNotes, we obtained different γ values for different parts of the corpus with no clearly better γ value. In a related paper (Chang et al., 2013), we apply an L^3M-related technique to predicted mentions achieving state-of-the-art results. Also, in the multiple pass setting, it took five passes to acheive top performance on the development set for both the datasets and for all the online algorithms.

5.2. Clustering of Online Forum Postings

We present experiments on document clustering using a large number of postings downloaded from discussions on an online forum[4]. We consider two different clustering perspectives for these posts as described below.

- **Author-based Clustering:** In this case, the task is to cluster the postings based on their authorship such that each cluster represents the items written by the same author. This task is essentially equivalent to Author Identification (Stamatatos, 2009), where a system is required to cluster a collection of textual items (e.g. emails, forum postings, articles) based on their authors. This task has potential applications, e.g., in email spam detection, intelligence, and criminal law.
- **Topic-based Clustering:** In this case, the task is to cluster the postings based on their discussion thread — all the postings belonging to the same discussion thread (e.g. '*what is a disabled veteran?*') correspond to one clus-

[2]Following the CoNLL shared task competition (Pradhan et al., 2012) on Coreference Resolution.

[3]Stoyanov & Eisner (2012) report best previously known B^3.

[4]Downloaded from `http://forums.military.com` following Lu et al. (2012)

	MUC	BCUB	CEAF$_e$	AVG	MUC	BCUB	CEAF$_e$	AVG
Technique ↓	ACE 2004				OntoNotes-5.0			
Corr-Clustering	77.45	81.1	77.57	78.71	84.26	75.03	63.07	74.12
Spanning Forest	73.31	79.25	74.66	75.74	84.75	73.93	60.47	73.05
Sum-Link (1 pass)	69.61	77.51	73.86	73.66	80.32	71.83	62.64	71.6
Sum-Link	72.7	78.75	76.42	75.96	82.26	74.59	64.8	73.88
Bin-Left-Link (1 pass)	74.19	79.3	77.77	77.09	80.74	72.15	64.36	72.42
Bin-Left-Link	76.02	81.04	77.6	78.22	81.57	73.18	65.54	73.43
L^{3}M ($\gamma = 0$) (1 pass)	76.7	80.89	78.02	78.54	84.45	76.18	66.41	75.68
L^{3}M ($\gamma = 0$)	77.57	81.77	78.15	79.16	85.14	77.01	67.6	76.58
L^{3}M (tuned γ) (1 pass)	76.7	80.89	78.02	78.54	85.07	76.97	67.17	76.40
L^{3}M (tuned γ)	**78.18**	**82.09**	**79.21**	**79.83**	**85.73**	**77.67**	**68.13**	**77.18**

Table 1: Performance on ACE 2004 and OntoNotes-5.0. Corr-Clustering is proposed by Finley & Joachims (2005); Spanning Forest is the latent spanning forest-based approach by Yu & Joachims (2009); Sum-Link is an online clustering technique by Haider et al. (2007); Bin-Left-Link uses a Best-Left-Link inference and the training strategy by Bengtson & Roth (2008). Our proposed approach is L^{3}M— L^{3}M with tuned γ is when we tune the value of γ using a development set; L^{3}M ($\gamma = \mathbf{0}$) is with γ fixed to 0. Corr-Clustering and Spanning Forest are batch clustering techniques. Sum-Link, Bin-Left-Link, L^{3}M (tuned γ), and L^{3}M ($\gamma = \mathbf{0}$) are online clustering techniques. "(1 pass)" means when trained with just one pass over the data.

total no. of authors	18,617
no. of item sets (one per day)	1,984
total no. of posts	690,498
avg. no. of posts per author	37.09
avg. no. of posts per item set	348
avg. no. of tokens per post	53.64
max. posts by author in item set	72

(a)

Technique ↓	Author	Topic	w/ one pass
Corr-Clustering	143.67	275.7	-
Spanning Forest	134.44	274.70	-
Sum-Link	133.12	245.44	249.75
Bin-Left-Link	133.09	240.69	246.76
L^{3}M ($\gamma = 0$)	133.39	240.73	244.13
L^{3}M (tuned γ)	**132.12**	**235.59**	**240.55**

(b)

Table 2: Tab. (a) presents summary statistics for the forum data. Tab. (b) presents results on the forum data for author-based and discussion topic-based clustering. Note that small VI is desirable. For author-based clustering, one pass over the data was sufficient for online algorithms. For topic-based clustering, we report results with one pass as well as five passes (last column) during training for online algorithms (note that one pass vs five passes distinction only holds for online clustering algorithms; for batch techniques we make ten passes.) All the results are scaled by 100. In all cases, L^{3}M (tuned γ) is statistically significantly better than all other approaches.

ter. In effect this means that we are clustering postings based on *topics*. The application of this includes detecting batches of spam emails that may share the same topic.

For performing 10-fold cross validation, we divided the data into separate item sets — each item set contains postings originating on the same day, ordered by the time of posting. Tab. 2a presents some statistics of the data.

Features and evaluation: We use the following pairwise features $\phi(i, j)$: TFIDF-based cosine similarity of the content, time difference between the posts, difference between their positions ($|j - i|$), and the common words between the posts (weighted by IDF.) For both the tasks, we report results in terms of the Variation of Information (VI) (Meilă, 2007) which is a popular metric used in the machine learning literature to measure distance between clusterings. We use a greedy algorithm for Corr-Clustering proposed by Finley & Joachims (2005) as the number of items in this task are too large for ILP inference.

The results are reported in Tab. 2b. For author-based clustering, a single pass was sufficient to achieve the top performance for online clustering techniques and so we do not report results with multiple passes separately. We observe that L^{3}M with tuned gamma outperforms all the other algorithms (p-value < 0.006 with Wilcoxon Signed Rank test using Holm-Bonferroni correction). In particular, again, tuning the γ value improves the performance significantly over $\gamma = 0$. For L^{3}M with tuned γ, the median best value of γ over the 10 folds was 0.4.

5.3. Impact of Non-Convexity on SGD Learning

While it is difficult to theoretically analyze Stochastic Gradient Descent (SGD) for non-convex functions, we perform some experiments to empirically estimate the impact of non-convexity on our SGD-based learning.

1. **Random Initialization:** In these experiments, we observe the variance in the quality of parameters learned by SGD when randomly initialized to estimate the robustness of SGD learning. We randomly initialize L^3M with $\gamma = 1.0$, perform SGD with one pass over the data, and measure the variance of the training data performance over 30 rounds. In each round, we randomly draw each element in $\mathbf{w}$ from $\mathcal{N}(0, 1)$ (standard Normal.) On coreference clustering over ACE 2004 data, we obtain a mean performance (w.r.t. the average of MUC, B^3, and CEAF) of 72.11 with a standard deviation of 0.17 (the low accuracy compared to the performance reported in Tab. 1 is due to the introduction of noisy and non-sparse feature weights.) On document clustering with randomly selected samples of size 500 (thus different testing data than Tab 2b), the VI results we obtain are: 143.2 +/- 1.8×10^{-2} for author-based clustering and 151.1 +/- 1.9×10^{-3} for topic-based clustering. The low variance in these results indicates that our SGD learning is very robust.
2. **Comparison with CCCP:** Recall that CCCP converges to a local minimum whereas SGD has no such theoretical guarantees for non-convex functions. To see if this indeed affects the performance, we compare their training data performance on author-based clustering for the forum data using L^3M with $\gamma = 1.0$. We find that in order to achieve performance close to just 1 pass of SGD, CCCP needs to perform 100 iterations, with the convex program within each iteration taking 100 further iterations. Early stopping CCCP by relaxing the stopping conditions is not a good option as it gives significantly worse results. As CCCP is very slow, we make comparisons only on randomly drawn (without replacement) small subsets of 100 training item sets. Averaged over 10 iterations, the CCCP performs better (i.e. has lower VI) than SGD by less than 0.2%. Thus SGD provides very slightly worse training data performance than CCCP with around 10,000x speed-up.

5.4. Controlling for the Effects of Item Order

In our experiments, we observe that L^3M not only outperforms other online clustering techniques but also the batch techniques (i.e. Corr-Clustering and Spanning). This result is mildly surprising as batch techniques have access to all the items at the same time and hence potentially more information. In fact, in some cases, other online clustering techniques (viz Sum-Link and Bin-Left-Link) also outperform the batch techniques.

Focusing on L^3M, its superior performance could be because of two reasons. 1) The probabilistic model assumed in L^3M is more suitable for the considered clustering tasks. 2) Considering the items in an online order captures an inherent ordering of items that aligns with how the true clusterings are realized based on the unknown underlying model (naturally, the obtained performance can be because of a combination of both.) In order to tease apart the contribution of these two effects, we conduct a control experiment where we randomize the order of items. With this randomization, we perform learning and inference as before for L^3M, Sum-Link, and Bin-Left-Link. The resulting drop in the performance then approximates the advantage of considering the items in their *natural* ordering for each of the algorithms. We use the same set-up as described before and conduct experiments on ACE 2004 Coreference data, Author Clustering, and Topic Clustering. Note that we keep the pairwise features between the items intact i.e. we make sure that the features that explicitly depend on the distance between items in the item set (such as the difference in the time samps of two posts) remain unaffected.

Results: We observe that after randomization, the performance declines significantly for coreference clustering (≈ 3 points) and for topic-based document clustering (VI goes up by ≈ 10 points), but not so significantly for author-based document clustering (< 1 point.) This implies that the order of the items is indeed key to the improved performance in coreference and topic-based clustering, but not so much for author-based clustering (where the improvement by L^3M over baselines is anyway small.) In retrospect this makes sense, as resolving coreference in a document with jumbled mentions is naturally going to be difficult, and topics in online media are likely to follow a temporal ordering. The exact detailed results are presented in the supplement.

6. Conclusions

We presented a pairwise, feature-based, and discriminative latent variable model for online clustering of data items. Our clustering model takes into account probabilities of multiple links when greedily connecting an item and uses a temperature parameter to tune the entropy of the resulting probability distribution. We proposed a learning framework that generalizes and interpolates between hidden variable CRF and latent structural SVM. We use an online stochastic gradient descent algorithm for learning that enjoys rapid empirical convergence. Applying our model to coreference resolution and document clustering, we showed that our approach outperforms existing online as well as batch structured prediction approaches to supervised clustering. Future work includes speeding up our inference so that it scales linearly with the number of items, and introducing item-to-cluster features in our model.

Acknowledgments This work is supported by an ONR Award on Guiding Learning and Decision Making in the Presence of Multiple Forms of Information, by DARPA under agreement number FA8750-13-2-0008, and by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the view of the agencies.

References

Bagga, A. and Baldwin, B. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, 1998.

Bansal, N., Blum, A., and Chawla, S. Correlation clustering. In *FOCS*, 2002.

Bengtson, E. and Roth, D. Understanding the value of features for coreference resolution. In *EMNLP*, 2008.

Blei, D. M. and Frazier, P. I. Distance dependent chinese restaurant processes. *JMLR*, 2011.

Chang, K.-W., Samdani, R., Rozovskaya, A., Sammons, M., and Roth, D. Illinois-coref: The UI system in the CoNLL-2012 Shared Task. In *CoNLL Shared Task*, 2012.

Chang, K.-W., Samdani, R., and Roth, D. A constrained latent variable model for coreference resolution. In *EMNLP*, 2013.

Daumé III, H., Langford, J., and Marcu, D. Search-based structured prediction. *Machine Learning Journal*, 2009.

Finley, T. and Joachims, T. Supervised clustering with support vector machines. In *ICML*, 2005.

Guha, S., Meyerson, A., Mishra, N., Motwani, R., and O'Callaghan, L. Clustering data streams: Theory and practice. *IEEE Trans. on Knowl. and Data Eng.*, 2003.

Guillory, A., Chastain, E., and Bilmes, J. Active learning as non-convex optimization. *JMLR*, 2009.

Haider, P., Brefeld, U., and Scheffer, T. Supervised clustering of streaming data for email batch detection. In Ghahramani, Zoubin (ed.), *ICML*, 2007.

LeCun, Y., Bottou, L., Orr, G., and Muller, K. Efficient backprop. In Orr, G. and K., Muller (eds.), *Neural Networks: Tricks of the trade*. Springer, 1998.

Lu, Y., Wang, H., Zhai, C., and Roth, D. Unsupervised discovery of opposing opinion networks from forum discussions. In *CIKM*, 2012.

Luo, X. On coreference resolution performance metrics. In *EMNLP*, 2005.

Mccallum, A. and Wellner, B. Toward conditional models of identity uncertainty with application to proper noun coreference. In *NIPS*, 2003.

Meilă, M. Comparing clusterings—an information based distance. *J. Multivar. Anal.*, 2007.

Ng, V. Supervised noun phrase coreference research: the first fifteen years. In *ACL*, 2010.

Ng, Vincent and Cardie, Claire. Improving machine learning approaches to coreference resolution. In *ACL*, 2002.

NIST. The ACE evaluation plan., 2004. URL `http://www.itl.nist.gov/iad/mig//tests/ace/ace04/index.html`.

Pletscher, P., Ong, C. S., and Buhmann, J. M. Entropy and margin maximization for structured output learning. In *ECML PKDD*, 2010.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *CoNLL*, 2012.

Quattoni, Ariadna, Wang, Sybor, Morency, Louis-Philippe, Collins, Michael, and Darrell, Trevor. Hidden conditional random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007. ISSN 0162-8828.

Rao, D., McNamee, P., and Dredze, M. Streaming cross document entity coreference resolution. In *COLING: Poster Volume*, 2010.

Samdani, R., Chang, M., and Roth, D. Unified expectation maximization. In *NAACL*, 2012.

Schwing, A. G., Hazan, T., Pollefeys, M., and Urtasun, R. Efficient structured prediction with latent variables for general graphical models. In *ICML*, 2012.

Shen, D., Yang, Q., Sun, J.-T., and Chen, Z. Thread detection in dynamic text message streams. In *SIGIR*, 2006.

Stamatatos, E. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 2009.

Stoyanov, V. and Eisner, J. Easy-first coreference resolution. In *COLING*, 2012.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, 1995.

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2002.

Yu, C. and Joachims, T. Learning structural svms with latent variables. In *ICML*, 2009.

Yuille, A. L. and Rangarajan, A. The concave-convex procedure. *Neural Computation*, 2003.