Deep Models and Artificial Intelligence for Military Applications:
AAAI Technical Report FS-17-03

Integration of Graphs and Representation Learning

Arjuna Flenner
Naval Air Systems Command

Abstract

Integrating information from many different data sources to provide better situational awareness is an essential Navy issue. Many data fusion models use statistical methods to reduce statistical errors. Machine learning and big data provide, on the other hand, provides a unique framework for information fusion through our ability to learn what added benefits a different modality can provide. In this work, we provide a novel data fusion method that integrates relational data, provided to us in the form of a graph, and image data. We build an energy model that learns a representation of the data where different data sources are assumed to be similar using a graphical model. The energy model is a non-convex function which we optimize using stochastic gradient descent with momentum. The effectiveness of the model is demonstrated in an automated target recognition example.

Introduction

Machine learning and big data applications heavily depends on the data representation. For this reason, many machine learning algorithms carefully preprocess and transform the data. Machine learning analysis of large collections of data sets build representations that facilitate downstream processing such as indexing, display of information, regression, and classification. These goals require the extraction of relevant features that encode interesting aspects of the observed data. Topic modeling (Blei 2012) and representation learning are two closely related techniques to learn the content of large data collections and these methods are the leading techniques in many machine learning application areas such as speech recognition, noise removal, and image classification. One convenient strategy to construct a meaningful structure is to suppose that the data can be represented in a mixed membership model.

Consider N data samples $\mathbf{x}_n \in \mathbb{R}^L$ such as documents or images. A simple data model assumes each sample can be written as a linear combination of vectors, i.e.

$$\mathbf{x}_n = \sum_{k=1}^{K} \boldsymbol{\psi}_k b_{nk} + \boldsymbol{\epsilon}_n, \tag{1}$$

where, the observed data $\mathbf{x}_n$ is represented up to a small reconstruction error $\boldsymbol{\epsilon}_n$ in terms of a set of *vectors* $\boldsymbol{\psi}_k$ (factors) and the coefficients b_{nk} describe the *weights* of the combination (factor loading). In other words, the basis vectors capture patterns present in the data set, i.e. a good set of features, and the learning representation task is to compute them, together with the associated weights. In general, the number of components in the combination, denoted by K, characterizes the complexity of the model, so it is often set beforehand, or its growth is limited by a regularization criteria.

In this work, we explore the utilization of a graph structure as a regularizer, with the hope of driving the learning procedure such that samples that are connected through the graph structure have similar representations. The advantages of using a graph are twofold. First, the graph enables the integration of information from different sources into our learning algorithms to influence the model priors. Second, the graph can encode relationships between data samples that do not come from a metric or distance function. Thus, information such as interactions or common group memberships, explicitly encoded by networks of connections, or social networks, can also be incorporated as part of the learning procedure.

We apply this graph-directed strategy dictionary learning (Mairal et al. 2009). Note that, in contrast with (Li et al. 2011), we are not trying to integrate topic modeling and dictionary learning. Instead, we are trying to integrate graph information into representation learning models. We show that using the graph structure to encode a-priori relations between observations allow for more distinctive basis vectors and, at the same time, lower average reconstruction errors.

To make the representation learning tractable, we build variational energy models and embed them in a learn the model parameters. The computations are carried out using a stochastic gradient descent with momentum method, whose energy-based formulation facilitates the information integration, in particular the graph encoded priors.

Previous Work

The problem of determining the vectors and vector weights in 1 is not well-defined without extra regularization conditions. The learning procedure overcomes this problem by building more structure into the problem by exploiting the inherent range of the data at hand or assuming some conditions over the basis vectors. These different assumptions are

the essential characteristics of the different methods. In the factor analysis field, different assumptions are used to decompose the data into a few factors using sparsity priors or a restriction on the number of basis vectors. The basic procedure is to establish a stochastic generative model to describe the data set and provide a framework for learning the parameters of the model.

Many early factor analysis models of integer valued data, such as non-negative matrix factorization (Lee and Seung 1999), neglect the discrete nature of count data (Wedel, Böckenholt, and Kamakura 2003). A case in point is the description of a corpus of text documents. In general, the corpus is given in terms of the times a word from the (corpus) vocabulary appears per document, and the learning task is expressed as the construction of topic models (Blei, Ng, and Jordan 2003; Griffiths and Steyvers 2004; Lafferty and Blei 2005; Blei and Lafferty 2006; Wallach, Mimno, and McCallum 2009). In the language of topic modeling literature, the set of basis vectors correspond to *topics*, and each topic is assimilated to a probability distribution of the words in the corpus vocabulary. Thus, more probable distributions are the ones that are compatible with the observed count of words. In (Blei, Ng, and Jordan 2003), each document is a mixture of topics, and the topics a distribution of words. The priors in both cases are symmetric Dirichlet distributions, leading to the well known Latent Dirichlet Allocation (LDA) model. Several subsequent works have studied variants of the LDA model. These include, correlated topic models (Lafferty and Blei 2005), where the basis elements are assumed correlated, while the words per topic are still assumed independent and dynamic topic models (Blei and Lafferty 2006) where the topics or topic weights are smoothed using techniques related to Kalmen filters . The work of Wallace et al (Wallach, Mimno, and McCallum 2009) studies the influence of stop words, number of topics selected and Dirichlet priors have in the resulting LDA topic model, and shows how the performance is improved when an asymmetric Dirichlet prior is used for the document-topic distributions. The LDA model has been extended to non-parametric Bayesian models in (McAuliffe, Blei, and Jordan 2006).

Related methods use spatial modeling with stick breaking, where a kind of spatial dependence of the basis vectors component is assumed by using a similarity kernel and a stick-breaking construction (Teh, Görür, and Ghahramani 2007; Paisley and Carin 2009; Paisley, Blei, and Jordan 2012).

Similar in spirit are the nonnegative matrix and tensor factorizations (Cichocki et al. 2009; **?**).

Gaussian Markov Random Fields (Rue/Held) Use a graph to define a Gaussian Process. Another work that tries to use graph priors is the work of Mimno et al (Mimno, Wallach, and McCallum 2008).

Representation Learning and Graph-Based Models

As introduced before, we are learning representations of the data by constructing linear combinations of appropriate set of basis vectors. We define the task as dictionary learning and use a sparse data representation as the mathematical framework to learn the dictionary representation. In both cases, we try to exploit additional information through a graph structure. The following subsections describe each of these model components.

Dictionary Learning

For the case of real valued data, a dictionary representation is constructed. Hence, each of the data points $\mathbf{x}_n$ is represented by

$$\mathbf{x}_n = \sum_{k=1}^{K} \boldsymbol{\psi}_k w_{nk}\, z_{nk} + \boldsymbol{\epsilon}_n, \tag{2}$$

where the basis vectors $\boldsymbol{\psi}_k$ correspond to the different dictionary atoms, the coefficients b_{nk} specify the weight associated to dictionary atom k in signal n, z_{nk} is an indicator variable that is equal to 1 if the dictionary atom k is used to represent $\mathbf{x}_n$ or 0 otherwise, and $\boldsymbol{\epsilon}$ is a residual or measurement noise (generally Gaussian), uncorrelated with the basis. In a stochastic framework.

Consequently, the dictionary learning task involves determining the number of atoms, the atoms themselves, as well as the atoms that intervene in a specific signal representation and the coefficients of the combination. These model parameters can be computed by minimizing an energy functional that incorporates the fidelity and sparsity goals of the regular dictionary learning problem (2), as well as regularizations given by the stochastic framework and the selected prior distributions. Further restrictions can be imposed by using graph-based priors as described next.

Graphical Models

Graphical models are often used to describe joint probability distributions of multiple variables (Cevher et al. 2010). A generic graph, denoted by $G(V, E)$, can be regarded as a node (vertex) set V and a collection of edges E that connect the nodes. The nodes in the graph are in one-to-one correspondence to random variables in the model, while edges in the graph encode dependency relationships between the nodes (random variables) they connect. The graph can be undirected, in which case the edges denote dependence between the corresponding nodes, or directed, in which case the conditional dependence is restricted to incoming edges.

However, an alternative take on the variable dependence representation with graphs can be constructed. Instead of correspondence between nodes and random variables, a correspondence between nodes and observations can be established. Specifically, each element in the node set $V = \{v_n\}_{n=1}^{N}$ is associated with a data sample $\mathbf{x}_n$ and an edge E_{ij} between the i-th and j-th nodes exists if sample i is related to sample j and does not exist otherwise. Note that this allows to encode known interactions between data samples. The interactions could be simple connections, or more specific quantitative dependencies such as metric information given in terms of a similarity measurement. The latter case is represented by a weighted undirected graph, $G(V, E, W)$, with W, the set of edge weights. The weight of edge E_{ij} can be given, for example, in terms of a Gaussian similarity

function

$$W_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right), \quad (3)$$

with σ^2 a positive constant value. Note that in this formulation, the weight W_{ij} measures the strength of the relationship between nodes i and j (equivalently, how similar are data points $\mathbf{x}_i$ and $\mathbf{x}_j$) .

Graph Laplacian and Graph Energy Let's define the degree of node i as

$$d_i = \sum_j W_{ij}. \quad (4)$$

Thus, by definition of W_{ij}, d_i measures how strong is the relation between sample $\mathbf{x}_i$ and the rest of the samples in the data set.

If $\mathbf{W}$ is the matrix of edge weights W_{ij}, and $\mathbf{D}$, a $N \times N$ diagonal matrix with diagonal elements $D_{ii} = d_i$, the graph Laplacian can be written as the matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{W}. \quad (5)$$

A state vector $\boldsymbol{\phi}_j = (\phi_{j1}, \ldots, \phi_{jK})^T$ can be associated to each of the $j \in \{1, \ldots, N\}$ nodes in the graph. The graph Laplacian allows to define the energy of the graph using the quadratic form

$$\langle \boldsymbol{\Phi}, \mathbf{L}\boldsymbol{\Phi} \rangle = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2 = \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}(\phi_{ik} - \phi_{jk})^2, \quad (6)$$

with matrix $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N)$, where each column corresponds to the state of a node in the graph. Note that this form of energy penalizes the differences in state for nodes that are closely related (edge with a large weight W_{ij}). Then, a state of minimal energy is characterized by a homogeneous state of strongly connected nodes. This does not exclude the trivial case where all the nodes have the same state. Other energy functions, based on p-Laplacian can be used (Bühler and Hein 2009). They are similar to the quadratic form but use an exponent p, with $1 \leq p < 2$.

As will be shown in the next section, previous information about the relationships of data points, encoded in terms of a weighted or unweighted graph, can be included in the computations of the model parameters by incorporating a graph energy term, expressed as a function of the graph Laplacian.

Defining the Energy Function

We use the state $\mathbf{b}_n = (b_{n1}, \ldots, b_{nK})^T$ for the state of node n in the dictionary learning problem. Thus, the matrix $\mathbf{B}$ corresponds to $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, and the energy potential can be written

$$\begin{aligned} U(\psi_{kw}, w_{dk}) &= \gamma_\epsilon \sum_n^N \left\| \mathbf{x}_n - \sum_k^K \boldsymbol{\psi}_k b_{nk}\, z_{nk} \right\|_2^2 \\ &+ \gamma_\psi \sum_k^K \|\boldsymbol{\psi}_k\|_2^2 \\ &- \sum_n \sum_k z_{nk} \log(\pi_k) \\ &- (1 - z_{nk}) \log(1 - \pi_k) \\ &- (c-1)\log(\pi_k) - (d-1)\log(1-\pi_k) \\ &+ \langle \boldsymbol{\Phi}, \mathbf{L}\,\boldsymbol{\Phi} \rangle \end{aligned}$$

Model Computations: Gradient Descent with Momentum

We exploit gradient descent to find a local minimal solution, but the problem is highly convex so we use stochastic gradient descent with momentum to find a deeper minima.

In order to define a momentum term we define the energy as

$$H(\psi_{kw}, w_{dk}, p) = U(\psi_{kw}, w_{dk}) + \frac{p^2}{2m}$$

The parameter p is the momentum term, which we draw from a Normal distribution with zero mean and variance m independently for each iteration. This term allows us to get unstuck from many local minima, but with high probability we remain in deep local minima.

The system evolution is simulated by means of the dynamics,

$$\begin{aligned} \frac{dq_i}{dt} &= +\frac{\partial H}{\partial p_i} = p_i, \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i} = -\frac{\partial U}{\partial q_i}. \end{aligned}$$

This dynamics is approximated by a *leapfrog* discretization using finite time steps. A leapfrog step can be expressed as

$$p_i\left(t + \frac{\epsilon}{2}\right) = p_i(t) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(t)) \quad (7)$$

$$q_i(t+\epsilon) = q_i(t) + \epsilon\, p_i\left(t + \frac{\epsilon}{2}\right) \quad (8)$$

$$p_i(t+\epsilon) = p_i\left(t + \frac{\epsilon}{2}\right) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(t+\epsilon)), \quad (9)$$

with ϵ representing the stepsize. This leapfrog update is applied for a specified number of steps L to simulate the evolution of the system for a time $\Delta t = \epsilon L$.

Note that in order to compute the dynamical updates it is necessary to compute the partial derivatives of U with respect to q_i.

Results

We apply the procedure to an automated target recognition (ATR) example. This example consists of recognizing nine different targets with several different viewing angles. Using

	1	2	3	4	5	6	7	8	9
1	454	44	0	0	0	0	2	0	0
2	43	457	0	0	0	0	0	0	0
3	0	0	493	3	2	1	1	0	0
4	2	0	5	479	4	1	0	0	9
5	0	0	6	5	482	6	0	1	0
6	0	0	1	0	3	490	5	1	0
7	3	0	3	2	2	8	482	0	0
8	0	0	0	0	0	1	0	496	3
9	0	0	0	7	0	0	0	2	491

(a) Without Graph - 96.09% Correct Classification

	1	2	3	4	5	6	7	8	9
1	487	10	0	1	0	1	1	0	0
2	11	489	0	0	0	0	0	0	0
3	1	0	480	15	0	4	0	0	0
4	0	0	7	481	5	1	0	0	6
5	0	0	1	6	481	11	0	1	0
6	1	0	4	0	10	470	14	1	0
7	0	0	0	0	0	9	491	0	0
8	0	0	0	0	0	0	0	500	0
9	0	0	0	2	0	0	0	0	498

(b) With Graph - 97.27% Correct Classification

Figure 1: ATR performance is increased 1.7% by including the graphical model.

our model, we increase our vehicle detection performance by 1.2%, but most of the performance gain was in the truck versus SUV class in which we gained a 7% improvement in performance, as shown in Figure 1.

Conclusion

In the modern Navy environment, sensor and and data storage technologies have advanced faster than our ability to analyze the data. Furthermore, there are a multitude of different sources of information. In this work we have demonstrated that machine learning is a practical method to integrate relational information and representations to improve target recognition. Our approach improved target recognition by 1.7% for a simple target recognition problem.

Central to our approach are representation learning algorithms. Representation learning does not require any prior knowledge of how two data sets should be related, but we learn how the representations of the data sets co-vary through an integration distribution or the associated integration potential.

References

Blei, D. M., and Lafferty, J. D. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, 113–120. ACM.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3:993–1022.

Blei, D. M. 2012. Probabilistic topic models. *Communications of the ACM* 55(4):77–84.

Bühler, T., and Hein, M. 2009. Spectral clustering based on the graph p-Laplacian. In Bottou, L., and Littman, M., eds., *Proceedings of the 26th International Conference on Machine Learning*. Montreal, Canada: Omnipress. 81–88.

Cevher, V.; Indyk, P.; Carin, L.; and Baraniuk, R. G. 2010. Sparse signal recovery and acquisition with graphical models. *IEEE Signal Processing Magazine* 27(6):92–103.

Cichocki, A.; Zdunek, R.; Phan, A. H.; and Amari, S.-i. 2009. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley.com.

Griffiths, T. L., and Steyvers, M. 2004. Finding scientific topics. *PNAS* 101:5228–5235.

Lafferty, J. D., and Blei, D. M. 2005. Correlated topic models. In *Advances in neural information processing systems*, 147–154.

Lee, D. D., and Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788.

Li, L.; Zhou, M.; Sapiro, G.; and Carin, L. 2011. On the integration of topic modeling and dictionary learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 625–632.

Mairal, J.; Bach, F.; Ponce, J.; and Sapiro, G. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, 689–696. ACM.

McAuliffe, J. D.; Blei, D. M.; and Jordan, M. I. 2006. Nonparametric empirical bayes for the dirichlet process mixture model. *Statistics and Computing* 16(1):5–14.

Mimno, D.; Wallach, H. M.; and McCallum, A. 2008. Gibbs sampling for logistic normal topic models with graph-based priors. In *NIPS Workshop on Analyzing Graphs*.

Paisley, J., and Carin, L. 2009. Hidden markov models with stick-breaking priors. *Signal Processing, IEEE Transactions on* 57(10):3905–3917.

Paisley, J. W.; Blei, D. M.; and Jordan, M. I. 2012. Stick-breaking beta processes and the poisson process. In *International Conference on Artificial Intelligence and Statistics*, 850–858.

Teh, Y. W.; Görür, D.; and Ghahramani, Z. 2007. Stick-breaking construction for the indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Wallach, H. M.; Mimno, D. M.; and McCallum, A. 2009. Rethinking LDA: Why priors matter. In Bengio, Y.; Schuurmans, D.; Lafferty, J.; Williams, C.; and Culotta, A., eds., *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc. 1973–1981.

Wedel, M.; Böckenholt, U.; and Kamakura, W. A. 2003. Factor models for multivariate count data. *Journal of Multivariate Analysis* 87:356–369.