Diversity Regularization of Latent Variable Models: Theory, Algorithm and Applications

Pengtao Xie, Machine Learning Department, Carnegie Mellon University

1. Background

Latent Variable Models (LVMs) are a family of machine learning (ML) models that have been widely used in text mining, computer vision, computational biology, recommender system, to name a few. One central task in machine learning is to extract the latent knowledge and structure from observed data and LVMs elegantly fit into this task. LVMs consist of observed variables used for modeling observed data and latent variables aimed to characterize the hidden knowledge and structure. The interaction between the observed and latent variables encodes modelers’ prior belief regarding how the observed data is generated from or correlated with the latent knowledge and structure. Under LVMs, extracting knowledge from data corresponds to inferring the value of the latent variables given the observed ones.

The hidden knowledge and structure behind data is normally assumed to contain multiple latent factors. The latent factors could be topics underlying document collection, clusters behind image collection, motifs in biopolymer sequences, etc. LVMs are designed to uncover these latent factors. To achieve this goal, LVMs are accordingly parametrized with multiple components. Under different models, the term component is called differently. For example, it is called topic in topic models, cluster in clustering models, atom in dictionary learning, component in Principal Component Analysis, neuron in Neural Network, to name a few. The modelers expect the components in the models to be aligned with the latent factors behind data as best as possible such that the latent factors could be uncovered by learning the components in LVMs.

2. Diversity Regularization of Latent Variable Models

In many real world applications, the popularity (or frequency) of latent factors behind data is distributed in a power-law fashion, where a few dominant factors occur very frequently while most factors in the long-tail region of the power-law distribution (called long-tail factors) have low popularity. For instance, in news corpora, a few topics like politics, economics, sports appear in nearly every newspaper whereas most topics such as art, lose-weight, trip are of very low frequency. As another example, in the Flickr image collection, a small amount of image clusters (or categories) such as animal, building, landscape have a lot of image instances while most clusters like painting, furniture, jewelry have a few images. Due to this high skewness of factor popularity, the standard LVMs tend to learn multiple components to best cover the dominant latent factors while choosing to ignore those in the long tail region. Failing to capture long-tail factors incurs significant information loss. Though the probability of each long-tail factor is low, the total probability mass of the long-tail region is large because the number of factors in this region is large. These long-tail factors with such a large total probability mass are of equal importance with, if not more importance than, the few dominant factors. In addition, the long-tail latent factors could be more interesting than the dominant ones. For example, in recommender systems, discovering a long-tail topic such as lose-weight is more valuable than a dominant topic like color in improving ads click-through rates and making profits.
Another issue often encountered in latent variable models is how to properly choose the number of components, which incurs a tradeoff between model effectiveness and computational efficiency: a small amount of components are not powerful and expressive enough to capture the latent factors while a large number of components cause high computational overhead. I am interested in investigating whether it is possible to achieve the best of both worlds: given a small set of components which facilitate computational efficiency, can the effectiveness of the LVM be comparable to that with a large number of components?

To solve the above stated two problems, I study a new type of regularization approach: diversity regularization, which encourages the components in LVMs to be different from each other. The motivation is as follows: for the first problem, the diversity regularizer can make the components in LVMs to diversely spread out, increasing the chance that long-tail factors can be covered; for the second problem, since the components are encouraged to be different from each other, each one captures information that is complementary to others, making it possible to capture sufficient information with a small amount of components. Specifically, I am interested in: 1) how to design a regularizer that encourages the components in LVMs to be diverse; 2) how to design algorithms to optimize this regularizer; 3) how to theoretically justify the effectiveness of this regularizer; 4) apply this regularizer to improve the performance of LVMs on real world application.

2.1 Define a Diversity Regularizer

In most LVMs, each component is characterized and parametrized with a vector, which we call characteristic vector. For example, in topic models, each topic is represented with a multinomial topical vector; in neural network, each neuron is associated with a vector of weights connecting to neurons in the lower layer. To diversify the components is precisely to diversify their characteristic vectors. Given the characteristic vectors $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_K]$ of $K$ components, where each column in matrix $\mathbf{A}$ corresponds to one component, their diversity can be informally described as how different each vector is from others. We start with measuring the dissimilarity of a pair of vectors, then proceed to measure the diversity of a vector set based on the pairwise dissimilarity measure. While there are many ways to measure the dissimilarity between vectors $\mathbf{a}_i$ and $\mathbf{a}_j$, we prefer to use the angle between them since the angle is invariant to translation, rotation and scaling (with positive factors) of the two vectors. In addition, we do not care about the orientation of vectors, thus preferring the angle to be acute or right. If the angle $\theta$ is obtuse, we replace it with $\pi - \theta$. To sum up, we define the angle $\theta(\mathbf{a}_i, \mathbf{a}_j)$ between vector $\mathbf{a}_i$ and $\mathbf{a}_j$ to be $\arccos\left(\frac{|\mathbf{a}_i \cdot \mathbf{a}_j|}{\|\mathbf{a}_i\|\|\mathbf{a}_j\|}\right)$. Based on this pairwise dissimilarity measure of vectors, we define the diversity of a set of vectors $\mathbf{A}$ as $\Omega(\mathbf{A}) = \Psi(\mathbf{A}) - \Pi(\mathbf{A})$, where $\Psi(\mathbf{A})$ is the mean of the non-obtuse angles between all pairs of vectors and $\Pi(\mathbf{A})$ is the variance of these angles. This diversity metric is defined with the rationale that a set of vectors possess a larger diversity if the pairwise angles between them have a larger mean $\Psi(\mathbf{A})$ and a smaller variance $\Pi(\mathbf{A})$. A larger mean implies that these vectors share larger angles on the whole, hence are more different from each other. A smaller variance indicates that these vectors uniformly spread out to different directions and each vector is evenly different from all other vectors. Encouraging the variance to be small can prevent the phenomenon that the vectors fall into several groups where vectors in the same group have small angles and vectors between groups have large angles. Such a phenomenon renders the vectors to be redundant and less diverse, and hence should be prohibited.
2.2. Diversity Regularized LVMs
We employ the above defined diversity metric to regularize the components in LVMs to encourage them to be diverse, and define the diversity regularized LVM problem as

$$\text{(P}_1\text{) } \max_{\mathbf{A}} L(\mathbf{A}) + \lambda \Omega(\mathbf{A})$$

where $L(\mathbf{A})$ is the objective function (which is assumed to be maximized) of the original LVM, $\Omega(\mathbf{A})$ is the diversity regularizer. $\lambda$ is a tradeoff parameter. $\lambda$ plays an important role in balancing the fitness of the characteristic vectors to the objective function and their diversity. Under a small $\lambda$, the characteristic vectors are learned to best maximize the objective function and their diversity is ignored. As discussed earlier, such components have high redundancy and may not be able to cover long-tail factors effectively and are of high complexity. Under a large $\lambda$, $\mathbf{A}$ is learned with high diversity, but may not be well fitted to the objective function and hence lose the capability to properly model data. To sum up, a proper $\lambda$ needs to be chosen to achieve the optimal balance.

3. Algorithm
The diversity regularizer $\Omega(\mathbf{A})$ is non-smooth and non-convex, presenting great challenge for solving problem $\text{(P}_1\text{)}$. In this section, we develop optimization techniques to tackle this issue. For the ease of optimization, we first reformulate this problem. Let $\mathbf{A} = \text{diag}(\mathbf{g})\hat{\mathbf{A}}$, where $\mathbf{g}$ is a vector and $g_i$ denotes the L2 norm of the $i$th column of $\mathbf{A}$, then the L2 norm of each column vector $\mathbf{a}_i$ in $\hat{\mathbf{A}}$ is 1. Based on the definition of the diversity metric, we have $\Omega(\mathbf{A}) = \Omega(\hat{\mathbf{A}})$. Accordingly, $\text{(P}_1\text{)}$ can be reformulated as

$$\text{(P}_2\text{) } \max_{\mathbf{g}, \hat{\mathbf{A}}} L(\text{diag}(\mathbf{g})\hat{\mathbf{A}}) + \lambda \Omega(\hat{\mathbf{A}})$$

s.t. $\forall i = 1, \ldots, K$, $\|\mathbf{a}_i\| = 1$, $g_i > 0$

($\text{P}_2\text{)}$ can be solved by alternating between $\mathbf{g}$ and $\hat{\mathbf{A}}$: optimizing $\mathbf{g}$ with $\hat{\mathbf{A}}$ fixed and optimizing $\hat{\mathbf{A}}$ with $\mathbf{g}$ fixed. With $\hat{\mathbf{A}}$ fixed, the problem defined over $\mathbf{g}$ is

$$\max_{\mathbf{g}} L(\text{diag}(\mathbf{g})\hat{\mathbf{A}})$$

s.t. $\forall i = 1, \ldots, K$, $g_i > 0$

which can be efficiently solved with many optimization methods. Fixing $\mathbf{g}$, the problem defined over $\hat{\mathbf{A}}$ is

$$\max_{\hat{\mathbf{A}}} L(\text{diag}(\mathbf{g})\hat{\mathbf{A}}) + \lambda \Omega(\hat{\mathbf{A}})$$

s.t. $\forall i = 1, \ldots, K$, $\|\mathbf{a}_i\| = 1$

where $\Omega(\hat{\mathbf{A}})$ is non-smooth and non-convex, thus is hard to solve. Now we focus on how to tackle it. We consider to use first-order method to solve this problem for its simplicity, wide applicability and efficiency. Since $\Omega(\hat{\mathbf{A}})$ is first-order and non-convex, (sub)gradient methods are not applicable. To address this issue, we derive a smooth lower bound $\Gamma(\hat{\mathbf{A}})$ of $\Omega(\hat{\mathbf{A}})$ and require $\Gamma(\hat{\mathbf{A}})$ to have the following two traits: 1) the gradient of $\Gamma(\hat{\mathbf{A}})$ is easy to compute; 2) optimizing $\Gamma(\hat{\mathbf{A}})$ with projected gradient ascent (PGA) can increase $\Omega(\hat{\mathbf{A}})$. The lower bound is given in Lemma 1.

**Lemma 1** Let $\det(\hat{\mathbf{A}}^T\hat{\mathbf{A}})$ denote the determinant of the Gram matrix of $\hat{\mathbf{A}}$, then $0 < \det(\hat{\mathbf{A}}^T\hat{\mathbf{A}}) \leq 1$. Let $\Gamma(\hat{\mathbf{A}}) = \arcsin \left( \sqrt{\det(\hat{\mathbf{A}}^T\hat{\mathbf{A}})} \right) - \left( \frac{\pi}{2} - \arcsin \left( \sqrt{\det(\hat{\mathbf{A}}^T\hat{\mathbf{A}})} \right) \right)^2$, then $\Gamma(\hat{\mathbf{A}})$ is a lower bound of $\Omega(\hat{\mathbf{A}})$. $\Gamma(\hat{\mathbf{A}})$ and $\Omega(\hat{\mathbf{A}})$ have the same global optimal.
We prove that maximizing $\Gamma(\mathbf{A})$ using PGA can increase the diversity metric $\Omega(\mathbf{A})$ in each iteration. The statement is formally described in Theorem 1.

**Theorem 1** Let $\mathbf{G}^{(t)}$ be the gradient of $\Gamma(\mathbf{A})$ w.r.t $\mathbf{A}^{(t)}$ at iteration $t$. For $\exists \tau > 0$, such that $\forall \eta \in (0, \tau)$, $\Omega(\mathbf{A}^{(t+1)}) \geq \Omega(\mathbf{A}^{(t)})$, where $\mathbf{A}^{(t+1)} = \mathbf{P}(\mathbf{A}^{(t)} + \eta \mathbf{G}^{(t)})$ and $\mathbf{P}(\cdot)$ denotes the projection to the unit sphere. Note that $\Omega(\mathbf{A}^{(t+1)})$ consists of two parts $\Psi(\mathbf{A})$ and $\Pi(\mathbf{A})$. To prove Theorem 1, we prove that maximizing $\Gamma(\mathbf{A})$ using PGA can increase the mean $\Psi(\mathbf{A})$ and reduce the variance $\Pi(\mathbf{A})$ simultaneously.

### 4. Theory

In this section, we present a theoretical analysis of the diversity regularizer, showing that with diversity regularization, the estimation error of supervised latent variable models can be reduced. Specifically, we use neural network to perform the theoretical study, while noting that the results can be easily extended to other supervised LVMs such as supervised Restricted Boltzmann Machine and distance metric learning. For simplicity, we only analyze neural network with one hidden layer. Extensions to multiple layers are straightforward.

#### 4.1. Setup

- **Task:** univariate regression
- **Network structure:** input layer, one hidden layer, output layer
- **Activation function:** rectified linear $h(x) = \max(0, x)$
- **Loss function:** $\ell(f(x), y)$
- **Prediction function:** $f(x) = \sum_{i=1}^{m} \alpha_i h(w_i^T x)$
- **Prediction function set:** $\mathcal{F} = \{ f \mid \forall i, j = 1, \ldots, m, \|w_i\|_2 \leq C_3, \theta_{ij} \geq \theta; \|\alpha\|_2 \leq C_4; \text{activation function } h(x) = \max(0, x) \}$
- **Loss function:** $l(x, y, f) = (f(x) - y)^2$
- **Loss function set:** $\mathcal{A} = \{ l \mid l(x, y, f) = (f(x) - y)^2 \}$

#### 4.2. Main Result

**Theorem 2** Let $L(f) = E_{(x, y) \sim P^*}[\ell(x, y, f)]$ denote the estimation error of prediction function $f$, where $P^*$ is the distribution of $(x, y)$, and $f^* \in \text{argmin}_{f \in \mathcal{F}} L(f)$ is the expected risk minimizer. Let $\hat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(x^{(i)}, y^{(i)}, f)$ be the training error and $\hat{f} \in \text{argmin}_{f \in \mathcal{F}} \hat{L}(f)$ be the empirical risk minimizer, then with probability at least $1 - \delta$

$$L(\hat{f}) - L(f^*) \leq 4C_1^2 C_3^2 C_4^2 m^2 \cos^2 \left( \frac{\theta}{2} \right) + 4C_1 C_2 C_3 C_4 \frac{\sqrt{m}}{\sqrt{n}} + 4C_5 + \frac{2 \log(2/\delta)}{n}$$

where $g(\cdot)$ is a non-decreasing function and $C_5 = \mathbb{E}[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i (y^{(i)})^2]$ is a constant that is
irrelevant to the prediction function set $\mathcal{F}$.

Note that in the first term of the right hand side, $\cos^2\left(\frac{\theta}{2}\right)$, is a decreasing function. A larger $\theta$ would induce a lower estimation error. $\Theta$ is the lower bound of the pairwise angles between the weight vectors connecting the hidden units. Hence, if the weight vectors have larger angles (hence are more different from each other and diverse), the neural network bears lower estimation error.

5. Applications

We impose diversity regularization over two popular latent variable models and empirical results demonstrate the effectiveness of the regularizer.

5.1. Restricted Boltzmann Machine with Diversity Regularization

Restricted Boltzmann Machine (RBM) is an undirected graphical model that has been successfully used for representation learning. An RBM consists of two layers of units where one layer of hidden units are used to capture latent features of data and one layer of visible units are used to represent the observed data. The correlation between hidden units and visible units are modeled with undirected weighted edges. Each hidden unit is characterized by a weight vector connecting to the visible units. We instantiate the general diversity regularized LVM problem $\mathcal{P}_1$ defined above to RBM and encourage the hidden units in RBM to diversely cover not only the dominant features, but also the long-tail features, to obtain more effective representations. We apply the diversity regularized RBM (DRBM) to learn representations for documents, and perform document retrieval and clustering over the learned representations. Three benchmark text datasets are used in the experiments, including TDT, 20-News, Reuters.

Figure 1 shows the retrieval precision on three datasets. The x-axis corresponds to the number of hidden units, the y-axis corresponds to precision@100 (the higher, the better). The blue bars denote results of standard RBM without diversity regularization, the red bars denote results of diversity regularized RBM (DRBM). As can be seen from the figures, with diversity regularization, the retrieval performance can be significantly improved, indicating that the representations learned by DRBM are much more effective than RBM, which corroborates the effectiveness of the diversity regularizer.

![Figure 1. Retrieval Precision](image)

Figure 2 shows the clustering accuracy (the higher, the better) on three datasets. Similarly, with diversity regularization the clustering accuracy can be greatly improved.
To examine whether diversifying the hidden units of RBM can capture long-tail factors, we break the retrieval precision down to the 9 categories in Reuters dataset (Table 1). The 2nd row shows the number of documents in each category. The distribution of document frequency is in a power-law fashion, where dominant categories (such as 1 and 2) have a lot of documents while most categories (called long-tail categories) have a small amount of documents. The 3rd and 4th row show the precision@100 achieved by RBM and DRBM on each category. The 5th row shows the relative improvement of DRBM over RBM. The relative improvement is defined as $\frac{P_{\text{drbm}} - P_{\text{rbm}}}{P_{\text{rbm}}}$, where $P_{\text{drbm}}$ and $P_{\text{rbm}}$ denote the precision@100 achieved by DRBM and RBM respectively. While DRBM improves RBM over all the categories, the improvements on long-tail categories are much more significant than dominant categories. For example, the relative improvements on category 8 and 9 are 366% and 397% while the improvements are 31% and 81% on category 1 and 2. This indicates that DRBM can effectively capture the long-tail factors.

<table>
<thead>
<tr>
<th>Category ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Documents</td>
<td>3713</td>
<td>2055</td>
<td>321</td>
<td>298</td>
<td>245</td>
<td>197</td>
<td>142</td>
<td>114</td>
<td>110</td>
</tr>
<tr>
<td>Precision@100 of RBM</td>
<td>0.69</td>
<td>0.44</td>
<td>0.09</td>
<td>0.10</td>
<td>0.06</td>
<td>0.04</td>
<td>0.04</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Precision@100 of DRBM</td>
<td>0.90</td>
<td>0.80</td>
<td>0.31</td>
<td>0.40</td>
<td>0.27</td>
<td>0.23</td>
<td>0.09</td>
<td>0.14</td>
<td>0.13</td>
</tr>
<tr>
<td>Relative Improvement</td>
<td>31%</td>
<td>81%</td>
<td>245%</td>
<td>289%</td>
<td>324%</td>
<td>421%</td>
<td>148%</td>
<td>366%</td>
<td>397%</td>
</tr>
</tbody>
</table>

Table 1. Breakdown of Retrieval Precision on Reuters Dataset

5.2 Distance Metric Learning with Diversity Regularization
Distance metric learning (DML) aims to learn a set of latent factors (distance metric) based on which the distances between data points can be effectively measured. The number of latent factors incurs a tradeoff: a small amount of factors are not powerful and expressive enough to measure distances while a large number of factors cause high computational overhead. We aim to achieve two seemingly conflicting goals: keeping the number of latent factors to be small for the sake of computational efficiency, meanwhile making them as effective as a large set of factors. To approach this goal, we impose a diversity regularizer over the latent factors to encourage them to be uncorrelated, such that each factor can capture some unique information that is hard to be captured by other factors. In this way, a small amount of latent factors can be sufficient to capture a large proportion of information, which retains computational efficiency while preserving the effectiveness in measuring distances. We apply the diversity regularized DML (DDML) to text, image and sensor datasets including 20-News, 15-Scenes, 6-Activities, and use the learned distance metric to perform
data retrieval.

Figure 3 shows the average precision under different number k of latent factors on 20-News, 15-Scenes and 6-Activities dataset respectively. The red and blue bars are the results of DDML and DML respectively. As shown in these figures, DDML with a small k can achieve retrieval precision that is comparable to DML with a large k. For example, on the 20-News dataset (left figure), with 10 latent factors, DDML is able to achieve a precision of 76.7%, which cannot be achieved by DML with even 900 latent factors. As another example, on the 15-Scenes dataset (middle figure), the precision obtained by DDML with k = 10 is 82.4%, which is largely better than the 80.8% precision achieved by DML with k = 200. Similar behavior is observed on the 6-Activities dataset (right figure). This demonstrates that, with diversification, DDML is able to learn a distance metric that is as effective as (if not more effective than) DML, but is much more compact than DML. Such a compact distance metric greatly facilitates retrieval efficiency. Performing retrieval on 10-dimensional latent representations is much easier than on representations with hundreds of dimensions. It is worth noting that the retrieval efficiency gain comes without sacrificing the precision, which allows one to perform fast and accurate retrieval.

Figure 3. Retrieval Precision