DRLnet: Deep Difference Representation Learning Network and An Unsupervised Optimization Framework *

Puzhao Zhang, Maoguo Gong†, Hui Zhang, Jia Liu
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi’an 710071, China
{omegazhangpzh, omegazhanghui, omegaliuj}@gmail.com

Abstract
Change detection and analysis (CDA) is an important research topic in the joint interpretation of spatial-temporal remote sensing images. The core of CDA is to effectively represent the difference and measure the difference degree between bi-temporal images. In this paper, we propose a novel difference representation learning network (DRLnet) and an effective optimization framework without any supervision. Difference measurement, difference representation learning and unsupervised clustering are combined as a single model, i.e., DRLnet, which is driven to learn clustering-friendly and discriminative difference representations (DRs) for different types of changes. Further, DRLnet is extended into a recurrent learning framework to update and reuse limited training samples and prevent the semantic gaps caused by the salutation in the number of change types from over-clustering stage to the desired one. Experimental results identify the effectiveness of the proposed framework.

1 Introduction
As remote sensing technology develops, there are more and more on-orbit satellites, which bring a large size of remote sensing data with high time-, spatial- and spectral resolutions [Chi et al., 2016; Marchetti et al., 2016]. It is demanding to process these increasing remote sensing data, and CDA is one of the most important applications in joint interpretation of remote sensing data, which aims not only to detect changes but also distinguish different types of changes. To better achieve this, it is necessary to learn more powerful and discriminative DRs from spatial-temporal images.

Recently, deep learning has achieved tremendous success in many vision and speech tasks such as image classification [Hinton and Salakhutdinov, 2006] and recognition [He et al., 2016; Krizhevsky et al., 2012], video understanding [Shao et al., 2016], image captioning [You et al., 2016] and natural language processing [Conneau et al., 2016] etc. The success of deep learning lies in the core that deep neural networks (DNN) has powerful ability in learning good representation from data in their raw form [Jiang et al., 2016]. For remote sensing images, it is also necessary to learn abstract representation for promoting joint interpretation of spatial-temporal images. However, unlike natural images, it is so lack of labeled data in the field of remote sensing that it is very hard to train a reliable model with certain scalability [Zhang et al., 2016a; 2016b]. Therefore, for practical purposes, it is demanding to develop unsupervised framework for joint interpretation of remote sensing data.

Clustering is one of the most popular unsupervised techniques often used to explore the hidden patterns and group similar structures [Shi and Malik, 2000; Hong et al., 2016; Dash et al., 2016]. However, it is very limited to perform clustering on the raw data. Deep learning outperforms others in learning good representation, but it is depended too much on supervised information during its training, while clustering may provide some reliable supervision information for training DNN in an unsupervised way [Wang et al., 2016]. Therefore, it is natural and promising to combine DNN and clustering for representation learning and special tasks, especially in the field of remote sensing where it is extremely lack of labeled data. Recently, clustering has been successfully integrated into the framework of deep learning, such as Deep Clustering Network (DCN) [Yang et al., 2016a], Deep Embedding Clustering (DEC) [Xie et al., 2016], and Variational Deep Embedding (VaDE) [Jiang et al., 2016] etc.

To effectively represent difference and measure difference degree, in this paper, a novel DRLnet and a recurrent learning framework are proposed. DRLnet firstly maps bi-temporal images into a suitable feature space to extract key information and suppress noise. Then, after automatic feature selection by the merging layer, the subsequent layers can learn more abstract DRs. By applying network forward passing and clustering, we can obtain the corresponding classification errors and clustering errors, which is used to tune network parameters in back propagation. As stated above, difference measurement, difference representation learning and unsupervised clustering are combined as a single model, i.e., DRLnet, which is driven to learn clustering-friendly and discriminative DRs for different types of changes. To strengthen its adaptability and

*This work was supported by the National Natural Science Foundation of China (Grant no. 61422209, 61672409), and the National Program for Support of Top-notch Young Professionals of China.

†Corresponding author: gong@ieee.org
stability, DRLnet can be extended into a recurrent framework by gradually decreasing the number of change types. In the recurrent framework, training samples would be updated and reused to learn more powerful model for CDA.

2 Related Work

2.1 Unsupervised Clustering

Clustering aims to group the data with similar structures into the same categories. Though many clustering methods have been proposed, such as K-means, spectral clustering [Nie et al., 2011], Gaussian mixture model (GMM) clustering [Bieracki et al., 2000] and agglomerative clustering [Gowda and Krishna, 1978] etc., unsupervised clustering still remains a fundamental challenge in the field of machine learning. In most of clustering methods, the similarity measures are limited in discovering the local patterns in data space, and thus it is hard to capture hidden and hierarchical dependencies in latent spaces [Dilokthanakul et al., 2016], while deep models can be used to encode rich latent structures hidden in raw data for improving the performance of clustering.

2.2 Deep Learning for Clustering

Recent works have demonstrated that optimizing representation learning and clustering jointly can greatly improve the performance of both. DCN [Yang et al., 2016a] combines learning good representation and clustering as a single model and optimize them jointly to find a ‘K-means-friendly’ space. DEC [Xie et al., 2016] was proposed to simultaneously learn representation and cluster assignments using DNN, and it works by iteratively optimizing a KL divergence based on clustering objective with a self-training target distribution. VaDE [Jiang et al., 2016] embeds the probabilistic clustering problems into a Variational Auto-Encoder (VAE) framework, and models the data generative procedure by combining a Gaussian Mixture Model (GMM) and DNN. Additionally, Yang et al. proposed a recurrent framework for joint unsupervised learning of deep representation and image clusters, where successive operations in clustering are unfolded as steps in a recurrent process, stacked on the top of representations output by a Convolutional Neural Network (CNN) [Yang et al., 2016b].

3 DRLnet Formulation

As depicted in Fig. 1, we present a novel DRLnet and a recurrent learning framework, where DRLnet is established to learn clustering-friendly and discriminative DRs for distinguishing different types of changes. In DRLnet, unsupervised clustering and DNN are combined as a single model for learning clustering-friendly DRs without any supervision. And it starts to run with over-clustering pseudo labels, and achieves the desired number of change types by gradually decreasing the number of clusters with a recurrent way.

3.1 Loss Function of DRLnet

K-means is one of the most famous unsupervised technologies that explore the latent pattern hidden in data and group the data with similar pattern into the same categories. However, in most cases, the data distribution is not friendly to K-means clustering. Recently, DNN has show its powerful ability in representation learning, but it often needs supervised fine-tuning to greatly improve the performance on specific tasks. Inspired by this, we try to combine K-means and DNN as a single model called DRLnet for difference representation learning in CDA task. With over-clustering pseudo labels, DRLnet is driven to learn more abstract DRs friendly to desired number of clusters. The over-clustering pseudo
labels may be not very reliable, but they are enough credible to avoid acquiring a trivial solution for DRLnet.

By combining DNN-based classification and K-means clustering as a single model, DRLnet can be formulated as the following loss function:

\[
\mathcal{L}(X; \theta) = \frac{1}{n} \sum_{i=1}^{n} \| f(x_1^i, x_2^i; \theta) - s_i^0 \|^2_2 + \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} \lambda_i \| d(x_1^i, x_2^i; \theta) - \mathbf{M} s_i \|^2_2 \tag{1}
\]

\[s.t. \quad \lambda_i, s_{j,i} \in \{0, 1\}, 1^T s_i = 1, \quad \forall i, j.
\]

where \( n \) denotes the total number of training samples set \( X \in \mathbb{R}^{d \times n} \), and \( \alpha \) controls the balance between classification accuracy and clustering performance. \( x_1^i \in \mathbb{R}^{d \times 1} \) and \( x_2^i \in \mathbb{R}^{d \times 1} \) is one pair of bi-temporal patches extracted from image-pair \( \mathbf{I}_1 \) and \( \mathbf{I}_2 \), respectively, and let \( x_t = [x_1^i, x_2^i] \). \( f(\cdot) \) is a deep network classifier, \( d(\cdot) \) is a difference representation extractor, and both of them share a part of parameters, and \( \theta \) collects all network parameters such as weights and biases. \( \mathbf{M} \in \mathbb{R}^{d \times K} \) is the collection of centroid with desired clusters, \( s_i^0 \in \mathbb{R}^{K_0 \times 1} \) is the initial pseudo labels acquired by unsupervised over-clustering, while \( s_i \in \mathbb{R}^{K \times 1} \) is the desired assignment with respect to \( \mathbf{M} \), where \( K_0 \) is the number of clusters in over-clustering, and \( K \) is the desired number of change types.

In Eq. (1), \( \lambda_i \) determines whether to execute K-means clustering on the difference representation of sample \( x_t \) or not, and it can be computed by the following Eq. (2). This equation means that if the difference between the feature pair \( (h_1^i, h_2^i) \) is larger than a threshold estimated by GGKI [Bazi et al., 2005], the corresponding sample \( x_t \) is taken as changed sample, otherwise, it is treated as unchanged sample.

\[
\begin{align*}
\lambda_i &= \frac{1}{2} (1 + \text{sgn} (\| h_1^i - h_2^i \|^2_2 - \tau)) \\
\tau &= GGKI (H^1, H^2)
\end{align*}
\tag{2}
\]

where \( h_1^i \) and \( h_2^i \) are the corresponding features of patches \( x_1^i \) and \( x_2^i \) learned by the feature extracting network, while \( H^1 = [h_1^1, \ldots, h_1^n] \) and \( H^2 = [h_2^1, \ldots, h_2^n] \), \( \tau \) is the automatic threshold estimated by GGKI [Bazi et al., 2005].

DRLnet is driven to learn K-means-friendly DRs from bi-temporal patches for better distinguishing different types of changes. By doing DNN-based classification and K-means clustering on the learned DRs, we can obtain classification and clustering errors, both of which are used to tune the network parameters of DRLnet by using back propagation. After training, all testing samples are fed into DRLnet and the corresponding DRs can be obtained, and then the desired CDA map can be generated by carrying out K-means on DRs.

### 3.2 An Recurrent Learning Framework

From the over-clustering pseudo labels to the desired number of change types, there may exists huge semantic gaps between them, which may be hard to bridge by a single salutation in number of clusters from over-clustering stage to the desired one. On the other hand, enough training samples is necessary for establishing a reliable deep network with certain capacity. Therefore, in this section, an recurrent framework is proposed to solve these two problems mentioned above. Instead of only one single salutation, the recurrent framework can achieve the desired number of change types by gradually decreasing the number of clusters from over-clustering stage to the desired one. In this procedure, the training samples are also updated and reused as the number of clusters decreases.

Suppose that the recurrent framework achieves the desired number of change types \( K_T \) from over-clustering classes \( K_0 \) through \( T \) timesteps, then the total loss function of the recurrent framework over all timesteps \( t \) from 1 to \( T \) can be formulated as:

\[
\mathcal{L}_{sum}(X; \theta) = \sum_{t=1}^{T} \mathcal{L}^t(X; \theta) \tag{3}
\]

For convenience, the \( f(x_1^i, x_2^i; \theta) \) and \( d(x_1^i, x_2^i; \theta) \) in Eq. (1) are written as \( f(x_t) \) and \( d(x_t) \), respectively. And the loss function at timestep \( t \) can be formulated as:

\[
\mathcal{L}^t(X; \theta) = \frac{1}{n} \sum_{i=1}^{n} \| f(x_t) - s_i^{t-1} \|^2_2 + \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} \lambda_i \| d(x_t) - \mathbf{M}^t s_i^t \|^2_2 \tag{4}
\]

\[s.t. \quad \lambda_i, s_{j,i} \in \{0, 1\}, 1^T s_i = 1, \quad \forall i, j.
\]

where all variables share the similar meanings as in Eq. (1), and \( s_i^t \in \mathbb{R}^{K_t \times 1} \) represents the label of sample \( x_t \) at timestep \( t \), \( \mathbf{M}^t \in \mathbb{R}^{d \times K_t} \) is the collection of centroids at timestep \( t \), here \( K_t \) is the desired number of change types at timestep \( t \). It’s worth noting that the superscript \( t \) on each variable means the current timestep, where \( t \) in \( \{1, 2, \ldots, T\} \).

---

**Algorithm 1** Alternating Optimization for DRLnet

**Input:** \( n \) pairs of bi-temporal patch pairs \( (x_1^i, x_2^i) \), where \( i \in \{1, n\} \), known \( \{s_i^{t-1}\} \) and \( \{\lambda_i^t\} \), initialized \( \mathbf{M}^t \) and \( \{s_i^t\} \).

**Output:** Parameters \( \theta \) of DRLnet at timestep \( t \).

1: \textbf{for} iter = 1 : Iteration \textbf{do}
2: \quad Update parameters \( \theta \) by Eq. (7);
3: \quad Update assignments \( \{s_i^t\} \) by Eq. (8);
4: \quad Update centroids \( \{m_i^t\} \) by Eq. (9);
5: \textbf{end for}
6: \textbf{return} DRLnet parameters \( \theta \) at timestep \( t \).

---

### 3.3 Optimization

The loss of the recurrent framework is formulated as Eq. (3), which needs to be minimized to achieve the desired solution. As mentioned in Section 3.2, the training samples would be updated at each timestep \( t \). And the minimization of the loss at each timestep \( t \) is dependent on the final stage acquired at timestep \( t - 1 \). Therefore, the minimization of the total loss in Eq. (3) can be achieved by minimizing the loss function in Eq. (4) at each timestep \( t \) from 1 to \( T \).
Figure 2: Datasets used in our experiments and CDA results of DRLnet on these four datasets. From top to bottom, they are Xi’an-1, Xi’an-2, Xi’an-3 and Xi’an-4 datasets, respectively. (a) Images acquired at time $t_1$. (b) Images acquired at time $t_2$. (c) 25 classes of changes. (d) 15 classes of changes. (e) 8 classes of changes. (f) 5 classes of changes (3 classes for Xi’an-2 dataset). (g) Final change intensity (CI) maps. (h) Ground truth maps.

**Algorithm 2 An Recurrent Learning Framework for CDA**

**Input:** $n$ pairs of bi-temporal patches $(x_i^1, x_i^2)$, where $i \in [1, n]$, and DRLnet. $C = \{K_0, K_1, \ldots, K_t, \ldots, K_T\}, 1 \leq t \leq T$, and $K_t > K_{t-1}$, where $K_t$ denotes the number of change types at timestep $t$, and $K_T$ is the targeted number of change types.

**Output:** Parameters $\theta$ of DRLnet and CDA map.

1: Obtain the initial pseudo labels $\{s_1^0\}$ of each sample by over-clustering the stacked raw data into $K_0$ classes; at the very beginning, all samples are taken as changed ones.
2: Pre-train DRLnet with over-clustering labels $\{s_1^0\}$.
3: for $t = 1 : T$ do
4: Compute $\{\lambda_i^t\}$ through Eq. (2) by applying GGKI on the learned DRs.
5: Initialize centers $M^t$ and assignments $\{s_i^t\}$ at timestep $t$: Assign unchanged samples with the same labels and compute the centroid of their corresponding DRs, and group the DRs of changed samples into $K_t$ classes to initialize $(1 + K_t)$ centers and $n$ assignments $\{s_i^t\}_{i=1}^n$.
6: Fine-tune DRLnet to learn K-means-friendly DRs at timestep $t$ via Algorithm 1 with known $\{s_i^{t-1}\}$ and $\{\lambda_i^t\}$, initialized $M^t$ and $\{s_i^t\}$.
7: Update $\{s_i^t\}$: Assign unchanged samples with the same labels, and group the DRs of changed samples into $K_t$ classes with K-means clustering.
8: $t \leftarrow t + 1$;
9: end for
10: return DRLnet and CDA map.

For each timestep $t$, we desire to minimize the loss function shown in Eq. (4), where three groups of parameters need to be solved, i.e., $(\theta, \{s_i^t\}, M^t)$, where $\{s_i^t\}$ is short for $\{s_i^t\}_{i=1}^n$.

For convenience, we let

$$L_i^t = \| f(x_i) - s_i^{t-1} \|^2 + \alpha \lambda_i^t \| d(x_i) - M^t_s_i^t \|^2$$  (5)

Having the DRs learned at timestep $(t - 1)$, we assign unchanged samples with the same label and group the DRs of changed samples into $K_t$ classes using K-means, initializing the centroids $M^t$ and the corresponding assignments $\{s_i^t\}$, where $K_t$ is the desired number of change types at timestep $t$.

For fixed $(M^t, \{s_i^t\})$, the network parameters $\theta$ can be updated by:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} L_i^t$$  (6)

$$\nabla_{\theta} L_i^t = \frac{\partial f(x_i)}{\partial \theta} (f(x_i) - s_i^{t-1}) + \alpha \lambda_i^t \frac{\partial d(x_i)}{\partial \theta} (d(x_i) - M^t_s_i^t)$$  (7)

where $\eta$ is the learning rate set in advance, and the partial derivatives $\frac{\partial f(x_i)}{\partial \theta}$ and $\frac{\partial d(x_i)}{\partial \theta}$ can be computed by using back propagation algorithm.

For fixed $(\theta, M^t)$, the assignments $\{s_i^t\}$ can be updated by Eq. (8) shown as follows:

$$s_{j,i}^t = \begin{cases} 1, j = \arg \min_{k=1, \ldots, K_t} \| d(x_i) - m_k^t \|_2 \\ 0, \text{otherwise.} \end{cases}$$  (8)
where \( s^t_{j,i} \) is the \( j \)-th element of the assignment \( s^t_i \).

When fixing \( (\theta, \{ s^t_{j,i} \}) \), the update of the centroid matrix \( M^t \) can be easily done as follows:

\[
M^t_k \leftarrow M^t_k - (1/c^t_k)(M^t_k - d(x_i))s^t_{k,i}
\]

\( s.t. \ k = 1, 2, \ldots, K \).

where \( K \) is the number of desired change types at timestep \( t \), \( c^t_k \) is the count of number of times this algorithm assign a sample to the cluster \( k \) before handling the next incoming sample \( x_i \), and the gradient step size \( 1/c^t_k \) controls the learning rate of the current centroid \( M^t_k \). The alternating optimization procedure of DRLnet at each timesep \( t \) is summarized in Algorithm 2.

4 Experiments

4.1 Datasets and Measurements

Four bi-temporal remote sensing datasets are used in our experiments to assess the effectiveness of the proposed framework. All of them are cut from two large-format remote sensing images, which are acquired by GF-1 satellite at August 19th, 2013 and August 29th, 2015, respectively. Each of them is composed of four bands, i.e., red, green, blue and near-infrared bands, and of the same spatial resolution 2m. Additionally, each pair of them has been radiometrically corrected and co-registered to make them as more comparable as possible. The overall detection accuracy (ACC) and Kappa coefficient are selected to quantitatively evaluate the proposed framework.

4.2 Experimental Setup

The proposed DRLnet is implemented based on deeplearn-toolbox [Palm, 2012]. We apply learning rate as 0.005 with a momentum 0.5 to all layers of DRLnet, and the sparsity of activation on each layer is set as 0.05. The proposed DRLnet is compared with a variety of baseline methods listed as follows:

- GGKI followed by K-means (Raw+KM): The classic change intensity map analysis method followed by applying KM to distinguish different types of changes.
- Stacked Autoencoder followed by GGKI and K-means (SAE+KM): This is a three-stage approach, SAE

be learned by DRLnet, which is beneficial to clustering for identifying different types of changes. The intermediate CDA maps also demonstrate that, as the number of change type decreases, the real changes are detected correctly and different types of changes can be distinguished better.

Fig. 3 visualizes the distribution of the DRs learned by DRLnet on the Xi’an-1 dataset. At the very beginning, it is not easy to group DRs learned by DRLnet with over-clustering, see Fig. 3(a). However, as the training goes with less number of change types, better DRs can be learned, and unchanged points and different types of changes can be easily grouped, as shown in Fig. 3(d). Clearly, these experimental results demonstrate the effectiveness of the proposed framework.

4.4 Comparison and Analysis

The proposed DRLnet is compared with a variety of baseline methods listed as follows:

- GGKI followed by K-means (Raw+KM): The classic change intensity map analysis method followed by applying KM to distinguish different types of changes.
- Stacked Autoencoder followed by GGKI and K-means (SAE+KM): This is a three-stage approach, SAE

Table 1: Quantitative Comparison with Baseline Methods

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Xi’an-1</th>
<th>Xi’an-2</th>
<th>Xi’an-3</th>
<th>Xi’an-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation Criteria</td>
<td>ACC(%)</td>
<td>Kappa</td>
<td>ACC(%)</td>
<td>Kappa</td>
</tr>
<tr>
<td>Raw+KM</td>
<td>88.12%</td>
<td>0.6785</td>
<td>90.89%</td>
<td>0.7912</td>
</tr>
<tr>
<td>SAE+KM</td>
<td>85.82%</td>
<td>0.6092</td>
<td>93.58%</td>
<td>0.8549</td>
</tr>
<tr>
<td>DRLnet(^1)</td>
<td>87.83%</td>
<td>0.6483</td>
<td>94.90%</td>
<td>0.8855</td>
</tr>
<tr>
<td>DRLnet(^2)</td>
<td>98.81%</td>
<td>0.9693</td>
<td>97.10%</td>
<td>0.9323</td>
</tr>
</tbody>
</table>

is used for feature learning, and then these features are compared to highlight changes. Finally, the changes are grouped into different clusters using KM.

- **DRLnet with fine-tuning only the top layers after the merging layer** (DRLnet\(^1\)).
- **DRLnet with fine-tuning across all layers of it** (DRLnet\(^2\), the proposed approach).

![Figure 6](image-url)

Figure 6: Clustering performance on the DRs learned by different layers of DRLnet over the Xi’an-1 dataset. (a) Top 3-th layer. (b) Top 2-th layer. (c) Top 1-th layer.

4.5 Clustering Performance on the Learned DRs

Fig. 6 shows the clustering performance on the DRs learned by different layers of DRLnet over the Xi’an-1 and Xi’an-4 datasets. From this figure, it is easy to find that deeper DRs has much better clustering performance. The reason lies in the fact that deeper layer has the ability to capture more abstract difference information from bi-temporal images. Compared with the ground truth maps, the clustering maps also show that deeper layer captures more accurate difference objects, while unchanged points may be grouped into different clusters because they are different ground objects in fact.

5 Conclusion

In this paper, we have presented a novel DRLnet and an recurrent learning framework for CDA in spatial-temporal remote sensing data. In DRLnet, difference measurement, difference representation learning and unsupervised clustering are combine as a single model, which can be driven to learn clustering-friendly and discriminative DRs for different types of changes without any supervision. And a recurrent learning framework is proposed to update limited training data and reuse them by gradually decreasing the number of change types from over-clustering stage to the desired one. Experimental studies demonstrate the effectiveness of the proposed DRLnet and the corresponding recurrent learning framework.
References


[Hong et al., 2016] Seunghoon Hong, Jonghyun Choi, Jan Feyereisl, Bohyung Han, and Larry Davis. Joint image clustering and labeling by matrix factorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(7):1411–1424, 2016.


