Heterogeneous Features Integration via Semi-supervised Multi-modal Deep Networks

Lei Zhao, Qinghua Hu(✉), and Yucan Zhou

School of Computer Science and Technology, Tianjin University, Tianjin, China
huqinghua@tju.edu.cn

Abstract. Multi-modal features are widely used to represent objects or events in pattern recognition and vision understanding. How to effectively integrate these heterogeneous features into a unified low-dimensional feature space has become a crucial issue in machine learning. In this work, we propose a novel approach which integrates heterogeneous features via an elaborate Semi-supervised Multi-Modal Deep Network (SMMDN). The proposed model first transforms the original data to high-level abstract homogeneous features. Then these homogeneous features are integrated into a new feature vector. By this means, our model can obtain abstract fused representations with lower-dimensionality and stronger discriminative ability. A Series of experiments are conducted on two object recognition datasets. Results show that our approach can integrate heterogeneous features effectively and achieve better performance compared to other methods.

Keywords: Deep neural network · Feature fusion · Pattern recognition

1 Introduction

With the development of sensor technology, multiple information from different channels can be obtained for one object. For example, information is described by images and text annotations in many social media websites, such as Flikcr, Google Picasa, etc. Meanwhile, more and more feature descriptors for one channel information (image or text) are proposed to extract high-level semantic information from the original low-level data. Different feature descriptors depict different aspects of the pattern's intrinsic structure. For an instance, in image processing, the Color Histogram depicts the color property of the image, HOG conveys the shape information and the feature of LBP extracts texture information from the original image.

Intuitively, heterogeneous features containing rich information can help to achieve better performance than single type of feature descriptors in many recognition tasks. However, these features usually have different structures, making it difficult or inefficient to use them in a classifier. What's more, they are always of high dimension which brings a drawback to computation efficiency. Therefore, how to integrate these heterogeneous features into new low-dimensional feature representations has become a crucial and attractive issue.

S. Arik et al. (Eds.): ICONIP 2015, Part IV, LNCS 9492, pp. 11–19, 2015.
DOI: 10.1007/978-3-319-26561-2_2

A simple way to integrate these high-dimensional heterogeneous features is concatenating all the features into a long feature vector and then utilizing conventional dimensionality reduction methods, e.g., PCA, ICA. However, it ignores the distinction among different types of features. Simply concatenation may even deteriorate the intrinsic structure of original features. To overcome this obstacle, some algorithms have been proposed.

Several methods are based on graph models. In paper [2,3], a shared common cluster indicator with non-negative constraint is constructed by non-negative matrix factorization (NMF), the different features are merged with the unsupervised spectral clustering. Cai [1] proposed another semi-supervised graph based approach for image classification. However, an intractable barrier for these graph based methods is the high computational complexity. Computation of these algorithms is infeasible when the dataset is very large.

Multiple Kernel Learning (MKL) based methods [6,7,12] are another commonly investigated methods for multi-modal learning. Considering each type of features as one modality, the MKL methods allocate one independent kernel for every feature modality. Then an ensemble kernel is learned to project all the features into an ensemble Reproducing Kernel Hilbert Space. A problem of these MKL based methods is that the base kernels for each modality should be specified manually. Since the base kernels can impact the final performance, how to select a proper kernel for each modality is a difficult and confusing problem.

Recently, deep learning has shown its power of learning latent feature representations. Some multi-modal deep learning models based on Deep Belief Net are also proposed [8,9]. However, these methods learn deep networks with two modalities, eg., image-text or video-audio pairs. In this paper, we propose a more flexible multi-modal learning model which integrates heterogeneous features. The proposed model is a Semi-supervised Multi-Modal Deep neural Networks (SMMDN) including multiple sub-networks and several top hidden layers. Treating each type of features as an independent modality, we allocate it a relatively independent sub-networks. The corresponding sub-networks of SMMDN will first transform the heterogeneous inputs into high-level abstract homogeneous representations. Then the top layers of SMMDN will integrate these homogeneous modality-free representations into a fused representation in a lower dimensional space, which is trained by another network.

In the following section, we will describe the architecture of the proposed model in detail.

2 Semi-Supervised Multi-Modal Deep Networks

2.1 Model Architecture

Figure 1 illustrates the complete structure of SMMDN. The whole model is decoupled into two subsections: Root Networks and Top Networks. As shown in Fig. 1, the root network comprises m sub-networks. We should note that all these sub-networks are different in terms of their inner layer structures. Different types of networks have distinct capability to extract features from raw data.

We should select appropriate deep neural networks dependent on the applications. After that, we introduce an auxiliary bridge layer to connect all sub-networks when jointly training them. These sub-networks are responsible for extracting high-level and modality-free representations from the input data. Once all the sub-networks are trained, batches of refined homogeneous feature representations can be extracted from the top hidden layers of these sub-networks. The Top Networks merge the refined homogeneous feature representations and project them into a shared low-dimensional feature space. In the following subsections, we introduce the two parts of SMMDN in detail and explain how to execute the training procedure.

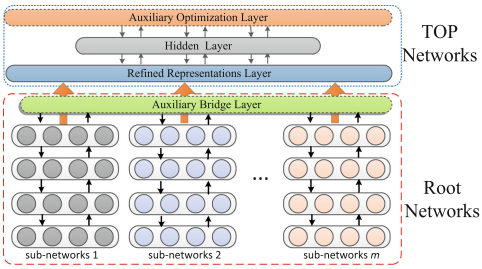


Fig. 1: Architecture of the SMMDN. The model consists of two parts: the Top Networks (*surrounded by the dotted box*) and the Root Networks (*boxed by dashed lines*).

2.2 Extracting Homogeneous Representations by Root Networks

A major problem of multiple features integration is the heterogeneity existing in the discriminative features. Fortunately, deep learning has demonstrated its potential in discovering latent hierarchical representations. In our model, we exploit this advantage of deep neural networks to eliminate the heterogeneity.

The intrinsic distribution structures of different modalities differ from each other. As to some simple structures, relatively shallow neural networks are able to extract their high-level latent representations. To the contrary, some modalities with complicated intrinsic structures may need deeper networks to extract their high-level representations. starting from this consideration, we design m heterogeneous sub-networks for the m modalities. They differ from each other in the number of hidden layers and hidden nodes. We denote the number of hidden layers of m-th modality as n_m. Figure 2 illustrates the detailed structure of the Root Networks.

There are two stages in training the Root Networks: unsupervised pre-training and supervised jointly fine-tuning, as shown in Fig. 2. Different from the methods of [8,9], in the pre-training stage, we train the sub-networks as Stacked Denoising Autoencoders (SDA) [13]. By this means, we can assign proper initialization to the connection weights between hidden layers. This unsupervised pre-training could avoid the networks from converging at local minimums. To find

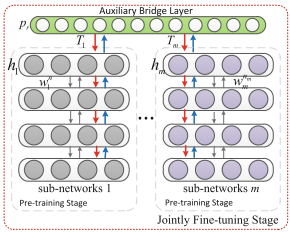


Fig. 2: Illustration of root networks. The training of root networks includes two stages: unsupervised pre-training and supervised fine-tuning. In stage of pre-training,we train every sub-networks (*bounded in grey dashed boxes*) independently. We introduce an auxiliary bridge layer in the fine-tuning stage to connect all the sub-networks and finetune all the sub-networks' parameters jointly.

the latent correlations across modalities, in the following fine-tuning stage we introduce an auxiliary bridge layer to connect all the sub-networks by utilizing the label information. As shown in Fig. 2, the auxiliary bridge layer is connected to the sub-networks top layer with the corresponding weights T_m.

In consideration of the correlations across modalities and to get homogeneous modality-free features, we let all the weights matrix T_m share the same weights matrix T. Let h_m denotes the top hidden layer of m-th modality and p_r denote the indicators of auxiliary bridge layer. We set the loss function as Eq. 1 where N denotes the number of training samples, $y^{(i)}$ denotes the label of sample $x^{(i)}$ and $h_j^{(i)}$ denotes the j-th modality's top hidden layer's output of sample $x^{(i)}$. For k-way classification problem, the probability that an input vector h_m belong to class i is denoted as Eq. 2, where b_{root} is a bias vector, T^ℓ denotes the ℓ-th row of matrix T. Actually, the loss is a logistical loss.

$$\mathcal{L} = -\sum_{j}^{m}\sum_{i}^{N} \log(P(Y = y^{(i)}|h_m^{(i)}, T, b_{root})) \tag{1}$$

$$P(Y = y^{(i)}|h_m, T, b_{root}) = \frac{exp(T^i h_m + b_{root_i})}{\sum_\ell exp(T^\ell h_m + b_{root_\ell})} \tag{2}$$

By minimizing the loss, we finely tune all the parameters of the Root Networks. Finally, we extract the abstract homogeneous features h_m from every sub-networks. Since we enforce the shared weighs T on all the connection weights T_m shown in Fig. 2. In fine-tuning stage, we fine-tune one sub-network with the remaining fixed and repeat this procedure until all the sub-networks are trained in every iteration. The auxiliary layer is just used for jointly finetuning all the Root Network, so it will be discarded when the fine tuning is done.

2.3 Feature Fusion with Top Networks

Since we have got a batch of homogeneous high-level abstract representations, the Top Networks will merge these representations with a non-linear transformation. Figure 3 demonstrates the detailed structure of the Top Networks.

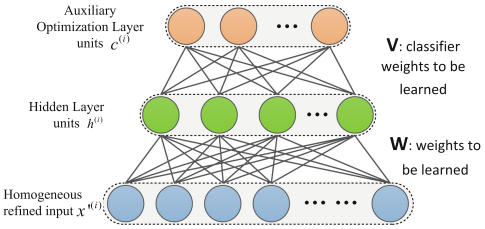


Fig. 3: Structure of the top networks. The homogeneous refined representations x' are projected into a low-dimensional feature space by the function $h = s(Wx' + b)$. The top auxiliary optimization layer is used for combining the supervised and unsupervised information to find the optimal weights W with back-propagation.

The homogeneous refined features will be concatenated as the input x' of the bottom layer. With the transformation matrix W, the input features will be projected into a new low-dimensional fused feature space. The final integrated feature h is extracted from the middle hidden layer. To take advantage of label information, we add an auxiliary optimization layer on the top of the Top Networks. This auxiliary could be a logistical classifier for our final classification task as well. This triple layer networks is used for the final feature integration. To train this networks, we set the final cost function $\mathcal{L}$ as following:

$$\mathcal{L} = \mathcal{L}_{dis} + \beta\mathcal{L}_{gen} + \lambda_1\|W\|_{\ell_1} + \lambda_2\|W\|_F^2 \tag{3}$$

$$\mathcal{L}_{dis} = -\sum_{i}^{N} \log(P(Y = y^{(i)}|h^{(i)}, V, b_{top})) \tag{4}$$

$$\mathcal{L}_{gen} = -\sum_{i}^{N} [x'^{(i)} \log \hat{x}'^{(i)} + (1 - x'^{(i)}) \log(1 - \hat{x}'^{(i)})] \tag{5}$$

There are three terms in the loss function: discriminative loss $\mathcal{L}_{dis}$, generative loss $\mathcal{L}_{gen}$ and the regularization. The discriminative loss let the integrated feature have strong discriminative ability. Similar to Eq. 2, we use the logistical loss here and formulated as Eq. 4. Beside the strong discriminative capability, we expect the fused features have better generative ability at the same time. The generative loss measures an average reconstruction error between x' and it's reconstruction $\hat{x}'$ with the transformation matrix of W. The small error mean that the fusion feature h has preserved most of the information from x'. We define this generative loss as Eq. 5. In Eq. 5, $\hat{x}' = s(W^T s(Wx+b)+b')'$ is the reconstruction of input x' where $s(\cdot)$ is the sigmoid function, b and b' are bias items. Now the weight matrix W is not only learned from the label information, but also from reconstructing the input x' unsupervisedly. Parameter β is introduced to balance loss $\mathcal{L}_{gen}$ and loss $\mathcal{L}_{dis}$. Moreover, we add two additional regularization items, which encourage sparsity and margin on weights W. This makes the model more robust. All these items form the final objective function. We use the gradient descent algorithm

to minimize the objective function and obtain the optimal parameters of the Top Networks. The regularization item $\|W\|_{\ell_1}$ in Eq. 3 is not differentiable at 0. It brings a problem for the gradient descent method. So we set the following approximation to smooth the loss function:

$$\|W\|_{\ell_1} = \sum_{ij} \sqrt{W_{ij} + \sigma} \tag{6}$$

where σ is a pre-specified scalar with a very small positive value. We set it as 10^{-31} in our all experiments.

3 Experiments

3.1 Datasets and Experiment Setup

We evaluate our model on two public image classification tasks: NUS-Object[1] and AWA[2]. The NUS-Object dataset includes 30000 imgaes collected from Flickr. Text descriptions are attached to every image. These images are categorized into 31 classes. We utilize the published five types of image features: 64-dimension Color Histogram (CH) features, 144 dimension color auto-CORRelogram (CORR) features, 73 dimension Edge Direction Histogram (EDH) features, 225 dimension Wavelet Texture (WT) features, Block-wise Color Moments (CM) features. Besides, we get two types of text features from the attached text with the LDA method: Doc-Topic distribution of LDA Topic model with 31 topics (31 dimension LDA31) and 81 topics (81 dimension LDA81). Generally, a deep neural networks could converge well just with relatively large-scale training data for its numerous parameters. So we take 20000 samples randomly as train set, the rest 10000 images as test data.

The AWA dataset consists of 30475 images of 50 animal classes. We take all the published features: 2000 dimension Local Self-Similarity (LSS) features, 2000 dimension colorSIFT (RGSIFT) features, 2000 dimension SIFT features, 2000 dimension SURF features, 2688 dimension Color Histogram (CQ) features and 252 dimension Pyramid HOG (PHOG) features. We choose 9607 images of 10 classes and take 8000 of them as train set and the rest as test set.

We compare our model with some MKL based methods and simple SVM methods described in Sect. 1: (1) SVM with the concatenation of all the original features, (2) PCA+SVM, we first reduce the dimension of the concatenation of original features to 300 dimension, then SVM is applied to the low-dimensional projected features, (3) lMKL [5], (4) gMKL [11], (5) sMKL [10], (6) cabMKL [4]. Graph based methods couldn't process large-scale data with acceptable time cost, so we don't compare our model with these graph based approaches. The SVM in all our experiments is implemented with the LIBSVM[3] software package. For MKL methods, we test different kernel including gaussian, polynomial and

[1] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm.
[2] http://attributes.kyb.tuebingen.mpg.de/.
[3] http://www.csie.ntu.edu.tw/%7ecjlin/libsvm/.

linear kernel and select the best kernels according to the final classification accuracy for every modality. Our multi-modal deep neural networks is implemented with the deep learning library of Theano[4]. There are three hyper-parameters $(\beta, \lambda_1, \lambda_2)$ in our model as shown in Eq. 4. We tried a series of values in the range of $[10^{-5}, 1]$ for every hyper-parameter and select the better value according to the final classification accuracy on a small validation set.

3.2 Experimental Results

First we evaluate the discriminative ability of the integrated feature. In Table 1, we give the final classification accuracy with integrating multiple heterogeneous features on the two datasets by different methods. To judge the promotion of discriminative ability of the fused features, we also give the accuracy with single modality by SVM in Table 2.

Table 1: Classification accuracies produced with different methods

Methods	SVM	PCA+SVM	lMKL	gMKL	sMKL	cabMKL	Our Methods
NUS-Obejct	0.4563	0.4290	0.5733	0.6300	0.6145	0.5579	**0.6373**
AWA	0.6623	0.5893	0.3385	0.6459	0.6727	0.4306	**0.7119**

Table 2: Classification accuracies of SVM with different modalities

NUS-Object	Features	CH	CORR	EDH	WT	CM	LDA31	LDA81
	Acc	0.2426	0.3066	0.2894	0.3063	0.2857	0.5062	0.4564
AWA	Features	LSS	RGSIFT	SIFT	SURF	CQ	PHOG	
	Acc	0.4941	0.5022	0.4088	0.5271	0.3920	0.3765	

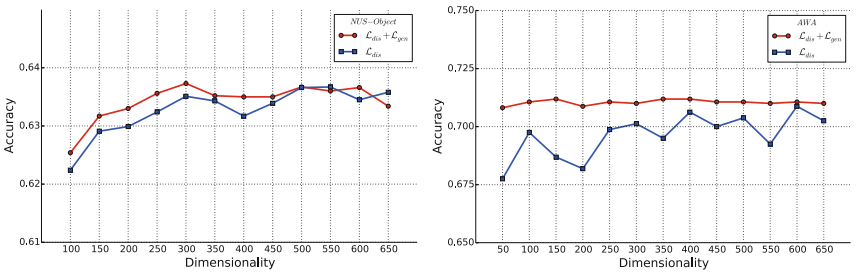
For dataset of NUS-Object, its seven types of features are extracted from two different channels: image and text. We noticed that the features from text (LDA31, LDA81) have much more discriminative ability than those features from image. However, if we simply concatenate all the features into a long feature vector and employ SVM, the classification accuracy is even lower than only using the LDA31 features. Because of the information loss of PCA, the method of PCA+SVM get even lower accuracy. These results verify the necessity of heterogeneous integrating especially when they come from quite different information channels. Though some MKL based methods performance well, our method outperform them and achieve better performance.

For dataset AWA, its six types of features are all from image. For SVM, PCA+SVM and most of the MKL methods, using the concatenation of all features can yields higher accuracy than using any single feature representations. However, our method get much higher classification accuracy compared with

[4] http://deeplearning.net/software/theano/index.html.

them. Results on two datasets confirm that our model can integrate heterogeneous features effectively and keep strong discriminative ability.

Though the discriminative ability is a very important criterion to evaluate the effectiveness of integration, outstanding fused features should preserve more information with less dimensionality from the original multiple features. So we verify our model's ability of dimensionality reduction. Figure 4 shows the variation of classification accuracy with changes in the fused feature's dimensionality. Many proposed deep architectures just exploit the discriminative loss, however we utilize the discriminative and generative information simultaneously. We also compared the experiment results with different loss function items in Eq. 3. The method with only discriminative loss is denoted by $\mathcal{L}_{dis}$ and shown with blue lines in Fig. 4. Red lines indicates the results of method with all the loss items in Eq. 3 and we denote it as $\mathcal{L}_{dis} + \mathcal{L}_{gen}$. It's clearly that our model can integrate all the original features into one fusion feature with very-low dimensionality while the fused features keep very strong discriminative ability. For the dataset of AWA, we can even integrate all its image features into a 10-dimension fused features without any classification accuracy loss nearly. Meanwhile, the experiment results show that with the introduced generative loss item $\mathcal{L}_{gen}$, our model can achieve better performance with same dimensionality. This fact confirms the integrating effectiveness of our model again.



(a) *NUS-Object* (b) *AWA*

Fig. 4: The classification accuracy changes with different dimensionality on the two Dataset (Color figure online).

4 Conclusion

We propose a novel deep neural networks based approach for heterogeneous features integration. Our model can integrate heterogeneous features from different sources into new fused features effectively. The fused feature have strong discriminative ability while low-dimensionality. All the experiment results confirm this effectiveness.

Acknowledgments. This work was supported in part by National Natural Foundation of China (No. 61222210) and 973 Program (2013CB329304). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

1. Cai, X., Nie, F., Cai, W., Huang, H.: Heterogeneous image features integration via multi-modal semi-supervised learning model. In: ICCV 2013, pp. 1737–1744. IEEE (2013)
2. Cai, X., Nie, F., Huang, H., Kamangar, F.: Heterogeneous image feature integration via multi-modal spectral clustering. In: CVPR 2011, pp. 1977–1984. IEEE (2011)
3. Chen, H., Cai, X., Zhu, D., Nie, F., Liu, T., Huang, H.: Group-wise consistent parcellation of gyri via adaptive multi-view spectral clustering of fiber shapes. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part II. LNCS, vol. 7511, pp. 271–279. Springer, Heidelberg (2012)
4. Cortes, C., Mohri, M., Rostamizadeh, A.: Algorithms for learning kernels based on centered alignment. J. Mach. Learn. Res. **13**(1), 795–828 (2012)
5. Gönen, M., Alpaydin, E.: Localized multiple kernel learning. In: ICML 2008, pp. 352–359. ACM (2008)
6. Guillaumin, M., Verbeek, J., Schmid, C.: Multimodal semi-supervised learning for image classification. In: CVPR 2010, pp. 902–909. IEEE (2010)
7. Lin, Y.Y., Liu, T.L., Fuh, C.S.: Local ensemble kernel learning for object category recognition. In: CVPR 2007, pp. 1–8. IEEE (2007)
8. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML 2011, pp. 689–696 (2011)
9. Srivastava, N., Salakhutdinov, R.R.: Multimodal learning with deep boltzmann machines. In: NIPS 2012, pp. 2222–2230 (2012)
10. Subrahmanya, N., Shin, Y.C.: Sparse multiple kernel learning for signal processing applications. IEEE Trans. Pattern Anal. Mach. Intell. **32**(5), 788–798 (2010)
11. Varma, M., Babu, B.R.: More generality in efficient multiple kernel learning. In: ICML 2009, pp. 1065–1072. ACM (2009)
12. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV 2009, pp. 606–613. IEEE (2009)
13. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: ICML 2008, pp. 1096–1103. ACM (2008)

http://www.springer.com/978-3-319-26560-5

Neural Information Processing
22nd International Conference, ICONIP 2015, November 9-12, 2015, Proceedings, Part IV
Arik, S.; Huang, T.; Lai, W.K.; Liu, Q. (Eds.)
2015, XVII, 702 p. 257 illus., Softcover
ISBN: 978-3-319-26560-5