Why Does Synthesized Data Improve Multi-sequence Classification?

Gijs van Tulder[1] and Marleen de Bruijne[1,2]

[1] Biomedical Imaging Group Rotterdam
Erasmus MC University Medical Center, The Netherlands
[2] Department of Computer Science
University of Copenhagen, Denmark

Abstract. The classification and registration of incomplete multi-modal medical images, such as multi-sequence MRI with missing sequences, can sometimes be improved by replacing the missing modalities with synthetic data. This may seem counter-intuitive: synthetic data is derived from data that is already available, so it does not add new information. Why can it still improve performance? In this paper we discuss possible explanations. If the synthesis model is more flexible than the classifier, the synthesis model can provide features that the classifier could not have extracted from the original data. In addition, using synthetic information to complete incomplete samples increases the size of the training set.

We present experiments with two classifiers, linear support vector machines (SVMs) and random forests, together with two synthesis methods that can replace missing data in an image classification problem: neural networks and restricted Boltzmann machines (RBMs). We used data from the BRATS 2013 brain tumor segmentation challenge, which includes multi-modal MRI scans with T1, T1 post-contrast, T2 and FLAIR sequences. The linear SVMs appear to benefit from the complex transformations offered by the synthesis models, whereas the random forests mostly benefit from having more training data. Training on the hidden representation from the RBM brought the accuracy of the linear SVMs close to that of random forests.

1 Introduction

Multi-sequence data can be very informative in medical imaging, but using it may cause some practical problems. Training a classifier on multi-modal data, for instance, generally requires that all modalities are available for all samples. If some modalities are missing, there is a range of methods for handling or imputing the missing values in standard statistical analysis [1]. Specifically for image analysis, there are synthesis methods that predict missing modalities. Some methods model the physical properties of the imaging process, e.g., to derive intrinsic tissue parameters from MRI scans [2] or to derive pseudo-CT from MRI in radiotherapy applications [3,4]. But an explicit model of the imaging process is not even required, as image processing techniques can be sufficient: for example, pseudo-CT images have also been made with tissue segmentation [5,6], with Gaussian mixture models [7] or by registering and combining CT images [8,9].

A. Frangi et al. (Eds.): MICCAI 2015, Part I, LNCS 9349, pp. 531–538, 2015.
The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-24553-9_65

Interestingly, data synthesis can not only generate images but also helps as an intermediate step. For example, Iglesias et al. [10] found that synthetic data improved the registration of multi-sequence brain MRI. Roy et al. [11] showed that synthetic sequences can improve segmentation consistency in datasets with multiple MRI contrasts. Li et al. [12] predicted PET patches from MRI data with convolutional neural networks, and found that including this synthetic PET data could improve classification of Alzheimer's disease.

There is something paradoxical about these results: if the synthetic data is derived from the available data and does not add new information, how can it still improve the performance? If the data synthesis is more flexible than the existing model, the synthetic data could add a useful transformation that makes the data easier to analyze. Data synthesis may also help to use the training data more efficiently, by allowing samples with different missing modalities to be combined into a single, large training set. Finally, synthesis methods that use unlabeled data, such as those discussed here, are an elegant way to add unsupervised learning to supervised models. However, most studies with synthetic data do not feature mixed training data or extra unlabeled examples, which suggests that the extra modeling power of the synthesis method could be important.

We present experiments that compare simple and complex classifiers trained with synthetic data on multi-sequence MRI data from the BRATS brain tumor segmentation challenge [13]. We use neural networks and restricted Boltzmann machines (RBMs) to provide synthetic replacements for missing image sequences. These representation learning [14] methods aim to learn new, abstract representations from the data. We use these representations to train linear support vector machines (SVMs) and random forests. We compare the results of using data synthesis with those of simply replacing missing data with a constant value. The data synthesis models are non-linear, so we expect that they can improve the results of the linear SVM but have a smaller effect for the random forests.

2 Methods

Image Synthesis with Neural Networks. We use a neural network with three layers: an input layer with nodes v_i to represent the voxels from the 3D input patches, a hidden layer with nodes h_j, and a layer with nodes y_k representing the 3D patch to be predicted. In this feed-forward network the visible nodes v_i are connected with weights W_{ij} to the hidden nodes h_j, which are connected to the output nodes $\hat{y}_k$ with weights U_{jk}. The parameters b_j and c_k are biases. The activation of the nodes given input $\mathbf{v}$ is given by

$$h_j = \text{sigm}(\sum_i W_{ij} v_i + b_j) \quad \text{and} \quad \hat{y}_k = \sum_j U_{jk} h_j + c_k, \tag{1}$$

with $\text{sigm}\,(x) = \frac{1}{1+\exp(-x)}$. We use backpropagation to learn the weights that optimize the reconstruction error between the predicted $\hat{\mathbf{y}}$ and true values $\mathbf{y}$:

$$\text{err}\,(\mathbf{y}, \hat{\mathbf{y}}) = \sum_k |y_k - \hat{y_k}| \,. \tag{2}$$

Restricted Boltzmann Machines. A restricted Boltzmann machine (RBM) models the joint probability over a set of visible nodes $\mathbf{v}$ and hidden nodes $\mathbf{h}$, with an undirected connection with weight W_{ij} between each visible node v_i and hidden node h_j. Each visible node has a bias b_i, each hidden node a bias c_j. We use noisy rectified linear units in the hidden layer and real-valued nodes with a Gaussian distribution for the visible nodes [15]. The weights and biases define the energy function

$$E\left(\mathbf{v},\mathbf{h}\right) = \sum_j \frac{\left(v_i - b_i\right)^2}{2\sigma_i^2} - \sum_{i,j} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_j c_j h_j \ , \tag{3}$$

where σ_i is the standard deviation of the Gaussian noise of visible node i. The joint distribution of the input $\mathbf{v}$ and hidden representation $\mathbf{h}$ is defined as

$$P\left(\mathbf{v},\mathbf{h}\right) = \frac{\exp\left(-E\left(\mathbf{v},\mathbf{h}\right)\right)}{Z}, \tag{4}$$

where Z is a normalization constant. The conditional probabilities for the hidden nodes given the visible nodes and vice versa are

$$P\left(h_j \,|\mathbf{v}\,\right) = \max(0, \sum_i W_{ij} v_i + c_j + \mathcal{N}(0, \mathrm{sigm}(\sum_i W_{ij} v_i + c_j))) \text{ and} \tag{5}$$

$$P\left(v_i \,|\mathbf{h}\,\right) = \mathcal{N}(\sum_j W_{ij} h_j + b_i,\ \ \sigma_i), \quad \text{with } \mathrm{sigm}\left(x\right) = \frac{1}{1 + \exp\left(-x\right)}. \tag{6}$$

We use stochastic gradient descent with persistent contrastive divergence [15,16] to find weights $\mathbf{W}$ and biases $\mathbf{b}$ and $\mathbf{c}$ that give a high probability to samples from the training distribution.

Although the energy $E\left(\mathbf{v},\mathbf{h}\right)$ can be calculated with Eq. 3, the normalization constant Z prohibits computing the probability $P\left(\mathbf{v},\mathbf{h}\right)$ for non-trivial models. However, we can still sample from the distribution using Gibbs sampling and the conditional probabilities $P\left(h_j \,|\mathbf{v}\,\right)$ and $P\left(v_i \,|\mathbf{h}\,\right)$ (Eqs. 5 and 6).

The standard RBM has one set of visible nodes. To model the patches for multiple sequences we use a separate set of visible nodes $\mathbf{v}^s$ for each sequence s, connected to a shared set of hidden nodes $\mathbf{h}$. There are no direct connections between visible nodes, so the interactions between sequences are modeled through the hidden nodes. We train this RBM on training samples with the same patch in every sequence to learn the joint probability distribution of the four sequences.

Image Synthesis with RBMs. In theory we could calculate the probability of one sequence given the others, $P\left(\mathbf{v}^s \,|\mathbf{v}\backslash\mathbf{v}^s\,\right)$, to predict a missing sequence, but the normalization constant Z makes this impossible. We resort to Gibbs sampling to synthesize the missing sequence. We initialize the model with the available sequences and keep these values fixed. We set the visible nodes for the missing sequence to 0, the mean value for our normalized patches. During Gibbs sampling we alternate sampling from the visible and hidden layers. We use the final values of the visible nodes for the missing sequence as the synthesized patch.

3 Data and Implementation

We used data of 30 patients from the BRATS 2013 brain tumor segmentation challenge [13] with four MRI sequences per patient: T1, T1 post-contrast (T1c), T2 and FLAIR. The scans of each patient are rigidly registered to the T1c image, which has the highest resolution, and resampled to 1 mm isotropic resolution. The dataset includes brain masks and class labels for four tumor structures.

For each patient we extracted patches of $9 \times 9 \times 9$ voxels from the same location in each sequence. For feature learning we used 10 000 patches per scan, centered at random voxels in the brain mask. For classification we used the label data to create a balanced training set with approximately $\frac{1}{5}$th of the samples for each class (four tissue classes and the non-tumor background).

We normalized the data twice. First, each scan was normalized to zero mean and unit variance to remove large differences between scans. After extracting patches we calculated the mean intensity, standard deviation and the intensity of the center voxel for each patch, since these features may help to discriminate tissue classes. Finally, we normalized each patch before training the neural networks and RBMs, since this helps to learn the local image structures.

We trained the neural network and RBM on unlabeled patches, implemented with the Theano library [17] for Python. The neural networks had one hidden layer of 600 binary nodes; the RBMs had 600 noisy rectified linear units in the hidden layer. Using more nodes or layers did not improve the performance. We used stochastic gradient descent with a decreasing learning rate for both models, with persistent contrastive divergence to estimate the updates of the RBM.

After training the models, we synthesized missing sequences from three known sequences, using Eq. 1 for the neural network and Gibbs sampling (20 iterations) for the RBM. As a baseline method, we replaced missing sequences with all zeros, the mean value of the normalized patches.

We trained random forest and linear SVM classifiers from Scikit-learn [18] to classify the five tissue types. The feature vectors were composed of either the normalized intensity values of observed and synthesized patches, or the values of the hidden layer of the RBM. We also included the intensity of the center voxel and the mean intensity and standard deviation of the patch intensities.

We repeated our experiments for five train/validation/test splits, each with 20 training scans, 5 scans to validate the model parameters and 5 test scans. For each split, we used the validation set to optimize the number of trees (up to 200) in the random forest, the L2 regularization of the SVM, and the hyperparameters of the neural networks and RBMs. We report the mean accuracy on the test sets.

4 Experiments

We present two classification scenarios. In the first, all samples are missing the same sequence. As a baseline we use the classification accuracy without data synthesis, measured on the full dataset and on datasets where we removed one sequence from the training and test data. Next, we look at data synthesis to complete the missing sequences. We trained classifiers on complete samples and tested on samples with one synthetic sequence. We also give the accuracy of classifiers trained on samples with a synthetic sequence, because the synthetic data might have a different distribution than the real data. Training and testing a classifier on data with different distributions might reduce its performance. Finally, we trained classifiers on the hidden representation from the RBM directly.

The second scenario uses a mixed training set, in which every sample is still missing one sequence, but where every quarter of the training set is missing a different sequence to simulate a combination of heterogeneous datasets. Without data synthesis, a separate classifier is needed for each subset of samples with the same three sequences. We use this as a baseline for the synthesis experiments. The RBM can be trained on the mixed training set. The neural networks have a practical problem: with no training samples with four sequences, we cannot train a network that predicts one sequence from the other three. Instead, we trained networks with one (MLP 1–1) or two (MLP 2–1) input sequences to predict one output sequence. Each option yields three networks to predict one sequence for a sample with three available sequences; we used the average prediction. We used the synthesis methods to complete the training set and compare with replacing the missing values with zeros, the mean value of the normalized patches.

5 Results

Table 1 shows the results of removing one of the MRI sequences from the test set. When training without synthesis, removing T1c or FLAIR reduced the accuracy more than removing T1 or T2, suggesting that T1c and FLAIR provide information that is not in T1 or T2. (The T1c scans also had a higher resolution.)

Training and testing with one synthetic sequence gave an accuracy similar to that of training on the dataset without the sequence. Replacing the synthetic data with zeros also gave similar results. This fits with our hypothesis that the synthetic data might not add new information. Adding synthetic data did not make the results much worse, which is useful if the synthetic data is used to combine data from multiple datasets. Using RBM synthesis was slightly better than using a neural network or replacing the sequence with zeros. Training on synthetic data instead of on real data slightly improved the accuracy, most likely because classifiers were confused by the different distributions of the real and synthetic data. Training on the hidden representation from the RBM increased the accuracy of the linear SVM and brought it closer to that of the random forest. This suggests that although the RBM does not add new information, it can still transform the data in a way that helps the linear SVM. The RBM representation did not improve the accuracy of the more complex random forests.

Table 2 shows the results of training with a mixed training set with partially incomplete data. Training on subsets of complete samples (sharing the same three sequences, $\frac{1}{4}$th of the samples) gave a lower accuracy than training on the full set. Using the synthesis methods to complete the samples, we trained a classifier on all samples, which gave a higher accuracy than training on subsets. There

Table 1. Classification accuracy (linear SVM | random forest) for different synthesis methods, with test sets in which all samples are missing the same sequence. Results in bold are significantly different from the baseline results in the top row ($p < 0.05$).

		Missing sequence			
	Full set	T1	T1c	T2	FLAIR
Train and evaluate on voxel values, without synthesis					
	68.83|73.22	67.90|72.97	58.67|61.62	68.26|72.87	59.13|69.60
Train on complete samples, evaluate with synthesized data					
with zeros		67.32|72.61	54.26|59.77	**67.17**|**72.08**	58.10|**65.03**
by MLP		68.32|72.95	**56.21**|**60.00**	67.48|72.52	58.33|68.53
by RBM		68.42|73.06	**55.34**|**60.33**	67.35|**72.38**	59.66|**67.57**
Train and evaluate with synthesized data					
with zeros		**68.47**|**73.36**	57.88|61.75	67.90|72.73	**59.94**|**69.38**
by MLP		67.37|73.01	58.34|61.22	**66.59**|**72.89**	**60.19**|**69.90**
by RBM		69.25|73.24	**60.53**|**61.47**	68.17|72.55	**62.30**|**69.88**
Train and evaluate on values from the RBM hidden layer					
RBM	**72.89**|**74.16**	**72.18**|**73.47**	**61.68**|**61.51**	**70.78**|**72.93**	**66.33**|**69.52**

Table 2. Classification accuracy (linear SVM | random forest) with partially incomplete training data, in which every scan is missing a random sequence. Boldface indicates a significant difference with the baseline ($p < 0.05$). The results for the full test set are compared with the best performing baseline (missing T2).

		Missing sequence in evaluation			
	Full test set	T1	T1c	T2	FLAIR
Train on subsets with complete samples (three sequences, $\frac{1}{4}$th of the full set)					
		62.30|67.99	54.92|59.48	62.71|69.03	51.06|65.51
Train on the mixed training set, with missing sequences filled-in					
with zeros	**66.85**|**70.86**	63.64|**69.90**	58.21|**63.70**	62.90|70.55	54.03|67.63
by MLP 1–1	**66.99**|**71.44**	64.28|**69.99**	59.27|**63.95**	63.50|**71.17**	**55.82**|**68.73**
by MLP 2–1	**65.42**|**71.22**	65.03|**69.76**	59.15|**64.01**	63.88|**71.21**	55.84|68.38
by RBM	**57.81**|**70.26**	**54.56**|**69.25**	**51.94**|**63.23**	**56.12**|**70.65**	50.60|68.10
Train and evaluate on values from the RBM hidden layer (all samples)					
RBM	**70.17**|**70.79**	**69.80**|**69.60**	**59.72**|**59.78**	**68.57**|**70.30**	**62.90**|**65.90**

was little difference between the two neural network approaches and replacing the missing values by zeros. The RBM synthesis gave a lower accuracy, possibly because synthesizing the missing training sequences made it harder to optimize the model. Training directly on the hidden representation from the RBM gave the highest accuracy for the linear SVM, as in the first experiment. The results with random forests were comparable to those of training on synthesized data.

6 Discussion and Conclusion

Data synthesis methods can improve the classification accuracy of multi-modal image analysis by providing synthetic data for incomplete examples. We first explored the explanation that the synthesis models may offer data transformations that are useful to the classifier. In our experiments in which the same modality was missing for all samples, we found few significant improvements from using synthetic T1, T1c or T2. We suspect that these modalities are too similar to produce useful transformations. Synthesized FLAIR did give a small improvement. Moreover, training on the RBM hidden layer significantly improved the accuracy for both classifiers and brought the SVMs close to the random forests. This suggests that the RBM extracts features that are new to the linear SVMs, but that could already be extracted by the random forests.

We found stronger improvements from using synthetic data in our second experiment. The synthesis methods made it possible to combine samples with different missing sequences in one training set. Using this larger training set increased the accuracy of both linear SVMs and random forests. We found similar results by replacing the missing values with zeros, the mean intensity after normalization. This suggests that at least part of the in accuracy improvement might be the result of having more training data.

In these applications the RBMs have a practical advantage over neural networks, because RBMs learn a joint probability distribution that can be used to predict any missing sequence. In contrast, neural networks are explicitly trained to predict one sequence given the others, so they need a separate network for each sequence. In our experiments the neural networks had a slightly lower reconstruction error, because the RBMs optimize a different learning objective.

Both neural networks and RBMs are trained with unlabeled data, a useful property that makes it easier to train them on large datasets. This can be an elegant way to use unlabeled data to improve a supervised classifier.

In conclusion: synthetic data might help classification because it allows better use of available training data, and because it offers new transformations of the data. This second contribution depends on the difference in complexity of the synthesis model and the classifier. A simpler classifier is more likely to benefit from the additional features that the synthesis model can extract from the data, even though the synthetic data does not contain extra information. In contrast, more complex classifiers can extract more information from the original data and are less likely to benefit from synthetic data. Whether it is better to include the extra complexity in the classifier or in a synthesis model is up for discussion.

Acknowledgements. This research is financed by the Netherlands Organization for Scientific Research (NWO). Brain tumor image data were obtained from the NCI-MICCAI 2013 Challenge on Multimodal Brain Tumor Segmentation (http://martinos.org/qtim/miccai2013/).

References

1. Little, R.J.A., Rubin, D.B.: Statistical analysis with missing data, 2nd edn. Wiley, New York (2002)
2. Fischl, B., Salat, D.H., van der Kouwe, A.J.W., Makris, N., Ségonne, F., Quinn, B.T., Dale, A.M.: Sequence-independent segmentation of magnetic resonance images. NeuroImage 23, S69–S84 (2004)
3. Johansson, A., Karlsson, M., Nyholm, T.: CT substitute derived from MRI sequences with ultrashort echo time. Medical Physics 38(5) (2011)
4. Johansson, A., Garpebring, A., Asklund, T., Nyholm, T.: CT substituties derived from MR images reconstructed with parallel imaging. Medical Physics 41 (2014)
5. Eilertsen, K., Vestad, L.N.T.A., Geier, O., Skretting, A.: A simulation of MRI based dose calculations on the basis of radiotherapy planning CT images. Acta Oncologica 47(7), 1294–1302 (2008)
6. Kapanen, M., Tenhunen, M.: T1/T2*-weighted MRI provides clinically relevant pseudo-CT density data for the pelvic bones in MRI-only based radiotherapy treatment planning. Acta Oncologica (Stockholm, Sweden) 52(3), 612–618 (2013)
7. Larsson, A., Johansson, A., Axelsson, J., Nyholm, T., Asklund, T., Riklund, K., Karlsson, M.: Evaluation of an attenuation correction method for PET/MR imaging of the head based on substitute CT images. Magnetic Resonance Materials in Physics, Biology and Medicine 26(1), 127–136 (2013)
8. Hofmann, M., Steinke, F., Scheel, V., Charpiat, G., Farquhar, J., Aschoff, P., Brady, M., Schölkopf, B., Pichler, B.J.: MRI-based attenuation correction for PET/MRI: a novel approach combining pattern recognition and atlas registration. Journal of Nuclear Medicine 49(11), 1875–1883 (2008)
9. Hofmann, M., Pichler, B., Schölkopf, B., Beyer, T.: Towards quantitative PET/MRI: a review of MR-based attenuation correction techniques. European Journal of Nuclear Medicine and Molecular Imaging 36(suppl. 1), March 2009
10. Iglesias, J.E., Konukoglu, E., Zikic, D., Glocker, B., Van Leemput, K., Fischl, B.: Is synthesizing MRI contrast useful for inter-modality analysis? In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013, Part I. LNCS, vol. 8149, pp. 631–638. Springer, Heidelberg (2013)
11. Roy, S., Carass, A., Prince, J.: A compressed sensing approach for MR tissue contrast synthesis. In: Székely, G., Hahn, H.K. (eds.) IPMI 2011. LNCS, vol. 6801, pp. 371–383. Springer, Heidelberg (2011)
12. Li, R., Zhang, W., Suk, H.-I., Wang, L., Li, J., Shen, D., Ji, S.: Deep learning based imaging data completion for improved brain disease diagnosis. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) MICCAI 2014, Part III. LNCS, vol. 8675, pp. 305–312. Springer, Heidelberg (2014)
13. Menze, B.H., Jakab, A., Bauer, S., et al.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Transactions on Medical Imaging (2014)
14. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. Technical report, Université de Montréal (2012)
15. Hinton, G.E.: A Practical Guide to Training Restricted Boltzmann Machines. Technical report, University of Toronto (2010)
16. Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. In: ICML (2008)
17. Bergstra, J., et al.: Theano: A CPU and GPU Math Compiler in Python. In: Proceedings of the Python for Scientific Computing Conference, SciPy (2010)
18. Pedregosa, F., et al.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)