Name:	Jens Beyermann
Course:	Explainable Machine Learning Seminar
Student number:	2905081
Date:	July 3, 2018

Deep Unsupervised Similarity Learning Using Partially Ordered Sets

published by Miguel A. Bautista, Artsiom Sanakoyeu and Björn Ommer

Jens Beyermann

1 Introduction

One of the central goals in explainable machine learning is to make the decisions of a complex machine learning system understandable for humans in an intuitive way. Similarity learning is especially interesting here, because the information, what entities get rated similar or dissimilar by the machine allows us to understand how the machine learned to see the data. Furthermore similarities allow us to visualize groupings in our data with the help of projections to a imaginable space while preserving most of the information given by the similarities. This leads us to another important perspective of explainability. To improve our understanding, how machine learning methods work, and why certain methods work better on different kinds of data, we need to understand how the feature spaces, into which our methods project data, work. An advantage of similarities is that they almost always come, explicitly or implicitly, with some kind of embedding into a feature space with a stable metric. Those spaces are relatively open to human interpretation and analysis.

Unsupervised methods are interesting since they are able to make use of unlabelled data. In the paper "Deep Unsupervised Similarity Learning" [1] the authors define a method, that trains a modern machine learning approach based on the results of a traditional method and therefore is able to work without external annotations of the data.

2 Learning with Partially Ordered Sets

To apply unsupervised learning to similarity learning the authors construct artificial "classes", so called surrogate classes, from the results of a traditional method. The traditional method here is the "histogram of oriented gradients" (HOG), which produces reliable results for images, that are very similar or very unsimilar. For more fine grained similarities the results of the HOG are unreliable. The underlying concept of the described paper is, to use the reliable results of the HOG to learn a similarity measure for the respective dataset. For a dataset $X \in \mathbb{R}^{n \times p}$ with n samples (images) and p pixels per sample, we get HOG similarities $s_{ij} = \exp(-||\phi(x_i) - \phi(x_j)||_2)$, where $\phi(x_i)$ is the representation of sample x_i in the HOG feature space. Each surrogate class will be represented by a label $\{-1, 0, \ldots, C-1\}$, where -1 labels all samples, not assigned to any class. The set of samples assigned to the surrogate class with label c is denoted with C_c.

Calculating the Initial Surrogate Classes

Starting with HOG similarities the authors assign a neighbourhood $\mathcal{N}(x_i)$ to every sample x_i defined as follows:

$$\mathcal{N}(x_i) := \{x_j \mid s_{i,j} \text{ within the top 5\%}, i \neq j\}.$$

Since this neighbourhoods are derived for every sample, they will be overlapping in many cases. To reduce the redundancy of this assignments, the authors use agglomerative clustering, that terminates if the merged classes inter-class similarity will be less than half of the inter-class similarity of it's constituents. The resulting classes represent the initial surrogate classes.

Defining Partially Ordered Sets

Most samples are not similar enough to any other sample to get assigned to any surrogate class. As a result the vast majority of the information in the dataset remains unused in the first step, because the relative similarities of the samples, not assigned to any class, would be ignored if the learning process would be based only on the surrogate class labels. To model the more fine grained similarities the authors introduce the concept of partially ordered sets or "posets".

Definition 1 (Partially Ordered Sets). *A Poset P_c with respect to a surrogate class c is the set $\{x_j, \ldots, x_k\}$ of all unclassified Points x_j, x_k that satisfy the following condition for all $x_i \in C_c$:*

$$e^{-||\phi^\theta(x_i)-\phi^\theta(x_j)||} > e^{-||\phi^\theta(x_i)-\phi^\theta(x_k)||} \Leftrightarrow j < k \;\forall\; j,k.$$

Where C_c denotes the points assigned to a surrogate class c. ϕ^θ denotes the feature representation given by the CNN with parameters θ.

Since elements of C_c are close to each other, compared to other elements, it is enough to represent each class by its medoid $\bar{x}_c = \sum_{x_j \in C_c} ||\phi^\theta(x_i) - \phi^\theta(x_j)||_2$.

Soft “Preassignment” Matrix

For the training process the described method preserves a tensor **R** containing a soft assignment of the Z nearest surrogate classes to every sample x_i. Therefore the matrix of the z nearest surrogate class representatives for every sample is defined as follows.

$$\mathbf{R}^z := \begin{pmatrix} \mathbf{r}_1^z \\ \vdots \\ \mathbf{r}_n^z \end{pmatrix} = \begin{pmatrix} r_{11}^1, \dots, r_{1d}^z \\ \vdots \\ r_{n1}^1, \dots, r_{nd}^z \end{pmatrix}$$

Loss Function and Optimization

The loss function for the CNN has two goals in this setting. On the one hand it should ensure the correct classification of samples, already labelled with a surrogate class. On the other hand it should “pull” unclassified samples closer to their Z nearest class representatives and “push” them away from other classes. To achieve this, the authors design a combined loss Function:

$$\mathscr{L}(X, y, \mathbf{R}; \theta) = \frac{1}{N} \sum_{i=1}^{N} \mathscr{L}_1(x_i, y_i; \theta) + \lambda \mathscr{L}_2(x_i, \mathbf{R}; \theta).$$

The classification loss $\mathscr{L}_1$ penalizes misclassification of samples x_i with label $y_i \neq -1$. Therefore it is basically a cross entropy loss with respect to the surrogate classes.

$$\mathscr{L}_1(x_i, y_i; \theta) = -\log \frac{\exp(t_{i,y_i}^\theta)}{\sum_{j=0}^{C-1} \exp(t_{i,j}^\theta)} \mathbb{1}_{y_i \neq -1}$$

This function is close to zero, if and only if the logits $t_{i,j}^\theta$ for the correct class $j = y_i$ are high compared to the logits for the wrong classes:
The poset loss $\mathscr{L}_2$ penalizes high distances between each sample x_i and it’s Z nearest class representatives $\{r_i^z\}_{z \in \{1,\dots,Z\}}$:

$$\mathscr{L}_2(x_i, R; \theta) = -\log \frac{\sum_{z=1}^{Z} \exp(\frac{-1}{2\sigma^2}(||\phi^\theta(x_i) - \phi^\theta(r_i^z)||_2^2 - \gamma))}{\sum_{j=1}^{C'} \exp(\frac{-1}{2\sigma^2}(||\phi^\theta(x_i) - \phi^\theta(r_j)||_2^2))}$$

This function can be seen as an generalized cross entropy loss on the basis of feature space similarities. In analogy to the cross entropy loss the $\mathscr{L}$ loss is close to zero, if (and only if) the distances between the respective sample x_i and its Z nearest class representatives are low compared to the distances between x_i and all class representatives. Therefore $\mathscr{L}$ pulls each sample in the feature space towards an $Z-1$-simplex spanned by the Z nearest surrogate representatives of the respective sample x_i. Since our data has to be processed batchwise, we can only take those classes into account, that are part of the current batch. C' denotes those respective classes. σ is the standart deviation of the current assignment of samples top the surrogate classes and γ is an hyper parameter for the margin between surrogate classes. On the basis of $\mathscr{L}$ we can construct a CNN to augment our initial feature space given by the HOGs, that is trained with the HOG based surrogate classes, as well as the fine grained similarities between the samples and their neighbouring classes. The resulting projection is denoted with ϕ^{θ} for CNN parameters θ.

Comparison to the “tuple”- or “triplet”-based similarity learning

Similarity learning is traditionally framed as a so called tuple or triplet based approach. This follows the idea to learn a representation of samples (and therefore a similarity in the represented feature space) from positive (i.e. similar) and negative (i.e. dissimilar) samples with respect to a given anchor sample those triplets (anchor, positive, negative) can be used to learn a representation on the basis of a “triplet loss” that prefers similarities between the anchor and the positive example and penalizes similarities between the anchor and a negative example. The authors point out that the poset learning approach is a generalization from this method since the method does not only take a positive or a negative sample into account but a whole set of samples along with respective similarities. The soft assignment of every sample to its Z nearest surrogate classes and the posets encoded in this assignment explicitly model fine grained similarities, that would have to be learned implicitly by a triplet loss network. The loss function of the poset learning method forces the CNN to order samples according to the similarity of their next Z classes.

Joint optimization

During the optimization process of θ the position of samples in the feature space gets augmented with every update of θ. Therefore it is possible that the initial assignment of surrogate classes is not valid anymore, after a certain amount of optimization steps for

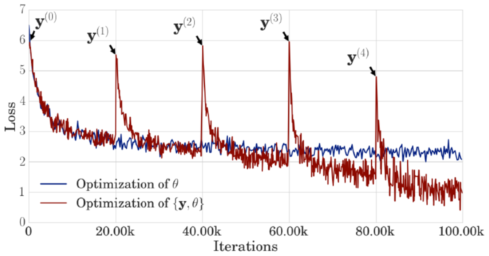

Figure 1: The development of the loss over several iterations of joint optimization

θ. Because of this, one has to see the optimization of θ and the assignment of surrogate classes $\mathbf{y}$ as interdependent optimization processes. The authors define an alternating approach between the optimization of θ and the assignment of surrogate classes on the basis of clustering. This process is denoted joint optimization. They describe the joint optimization as RNN setting. In time step m of the “RNN” y and θ get updated one after another. The optimization of y is described as follows:

$$y^{(m)} = \arg\max_{y} \mathcal{G}(X; \phi^{\theta^{(m-1)}}, y^{(m-1)})$$
$$\textbf{s.t.} \sum_{i:y_i=c} 1 > t, \ \forall c \in \{0, \ldots, C-1\}$$

where t denotes a lower bound on the number of each samples per cluster and $\mathbb{G}$ defines a “cost” function $\mathcal{G}$ that measures the quality of the current clustering:

$$\mathcal{G}(X; \phi^{\theta^{m-1}}, y^{(m-1)}) = \sum_{c=0}^{C-1} \frac{\sum\limits_{i:y_i=c} \sum\limits_{j:y_j=c} e^{(-||\phi^{\theta}(x_i) - \phi^{\theta}(x_j)||_2)}}{\left(\sum\limits_{j:y_j=c} 1\right)^2}$$

After the update of the assignments of the surrogate classes, the method can further optimize θ with the former described loss function $\mathcal{L}$:

$$\theta^{(m)} = \arg\min_{\theta} \mathcal{L}(\mathbf{X}, \mathbf{y}^{(m)}, \mathbf{R}^{(m)}; \theta^{(m-1)})$$

In Figure 1 we can see, that after each new assignment the loss starts at a worse point than before, since the network has to learn the new assignments given by the clustering, but overall we can observe that the joint optimization does in fact improve the total quality of the feature space representation.

For the actual training the CNN gets optimized by stochastic gradient decent for a number of iterations for each fixed assignment $\mathbf{y}$.

3 Experiments

In the experiment section we want to discuss whether the proposed method is able to compete with other unsupervised or even with traditional supervised approaches. Another question is whether one can use the weights of a (possibly more general) pretrained network for initialization, alternatively to the HOG similarities. It appears possible, that additional training based on the poset similarities might enhance the performance of a traditionally trained network. Another interesting task is the opposite idea to the former task. Instead of using supervised results as starting point for poset learning one could also use the results of poset learning as initialization for a supervised model. The comparison methods are a triplet based approach denoted with shuffle&learn[8], a tuple approach from Doersch et al.[3], an unsupervised feature learning network called exemplar-CNN[4], alexnet[6], a support vector machine denoted with exemplar-SVM[7], the authors own method from a previous paper [2] and the initial HOG-LDA method as baseline[5].

Human Pose Estimation

The central application in this papers experiments is the human pose estimation task. For this purpose deep unsupervised similarity learning is applied to three datasets. The first is the Olympic sports (OS) dataset, the second is the Leeds sport pose (LSP) dataset and the third one is the MPII dataset.

Olympic Sports

This dataset consists of video sequences showing 16 different kinds of sport competitions. The description of the experiments leave some room for interpretation, but we have to assume, that the authors use the similarities (i.e. the deep feature representations) to evaluate a traditional classifier based on similarity measures. They state, to follow the evaluation protocol from their previous paper [2]. Here they use a nearest neighbour based evaluation. They compute the nearest neighbour frame to a given query frame by similarities obtained from the poset learning and use the training label to evaluate the unsupervised and therefore unlabelled data in comparison to supervised methods.

HOG-LDA	Ex-SVM	Ex-CNN
0.62	0.72	0.64
Alexnet	Doersch et. al	Suffle & Learn
0.65	0.62	0.63
CliqueCNN	Ours scratch	Ours Imagenet
0.83	0.78	0.85

Table 1: Evaluation of the Olympic Sports dataset.

At this point it would have been interesting to see how the overlap between the final surrogate classes and the "real" classes would look like. For evaluation they use an AUC score which can be interpreted as the probability that a randomly chosen positive sample (for a certain clasificaion task) gets ranked higher than a randomly chosen negative sample. The classification task is, according to their evaluation script at the projects github page, the retrieval of the correct category out of the 16 competitions. The results of the authors evaluation imply that poset learning works pretty well. Poset learning, trained from scratch works significantly better than nearly all other methods the authors compare to (see table 1). Only clique-CNN and poset learning based on imagenet pretraining work better. The latter achieves the highest score in the experiment with an AUC of 0.85. This answers the question, whether poset learning can profit from transfer learning and shows that the usage of a pretrained network for initialization can indeed be superior to the initialization by HOG similarities. It still might be important that the initialization network is trained to perform a "more general" task than the poset learning model, since the method would risk to learn a distribution not related to the actual problem. In direct comparison to other similarity learning tasks, based on classical tuple or triplet formulations, poset learning achieves a 16% higher performance. The authors explain this discrepancy with the more detailed similarities encoded in posets. The big discrepancy still seems astonishing and I tried to find papers with comparable experiments on this dataset, but the only one i could find was a paper from the same group (but different authors) [9]. This paper got comparable results but their approach was seemingly not build on the methods of poset learning and the closely related CliqueCNN [2] paper. Otherwise it has to be assumed, that there has been some information transfer inside a group so the later paper would have profited at least indirectly from the knowledge obtained by the first two papers. On the other hands most of the methods do not perform much better than the HOG method from 2012 which might mean, that

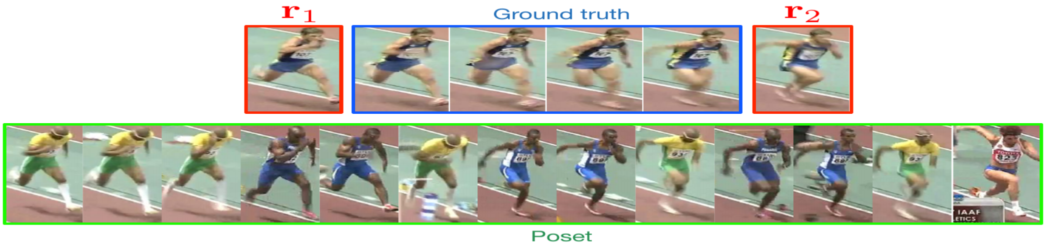

Figure 2: In the top row we see the frames from a video sequence connecting two surrogate class representatives. A partially ordered set obtained from the poset learning method, successfully represents the states of movement in the sequence even though no temporal component was explicitly encoded.

they are not very well suited for this task anyway. Without further experiments it would be hard to give a reliable statement here.

In the context of explainability the qualitative evaluation is more interesting. In fig. 2 we can see, that the similarity order between two surrogate classes represents the process of the posture transformation between the two class representatives. In other words: There are two runner poses and other poses ordered by their relative similarity to those two runners do not only reproduce our intuition, they even model the process of postures one would have to go through to move from the first posture to the second one. This tells us that the learned similarities are closely related to “real” similarities.

Leeds Sport Pose

On this dataset poset learning is applied in form of a unsupervised (zero shot) and semi supervised setup. For the unsupervised Experiment the authors transfer the representation learned on the OS dataset without any additional training on the LSP data. To evaluate during testing, they estimate the joint coordinates of the most similar frame from the training set to a given query frame and assign its joint coordinates to the tested frame. Furthermore they estimate an upper bound of the total performance of any unsupervised method by obtaining the most similar pose for a given query by the frame which is closest to average distance of ground truth pose annotations. This method achieves 69.2 PCP (*percentage of correctly estimated body parts*). This border is a theoretical construct, since all compared unsupervised methods perform not nearly as good as this upper limit. It still shows that unsupervised learning is a pretty difficult task in this setting since even with a significant advantage one still can’t get near to 100%. For their evaluation they structure the compared methods into 3 groups table 2a. The

Method	T	UL	LL	UA	LA	H	Total
Ours - Imagenet	83.5	54.0	46.8	34.1	16.8	54.3	48.3
CliqueCNN	80.1	50.1	45.7	27.2	12.6	45.5	43.5
Alexnet	76.9	47.8	41.8	26.7	11.2	42.4	41.1
Ours - Scratch	67.0	38.6	34.9	20.5	9.8	35.1	34.3
Shuffle&Learn	60.4	33.2	28.9	16.8	7.1	33.8	30.0
Ground Truth	93.7	78.8	74.9	58.7	36.4	72.4	69.2
P. Machines	93.1	83.6	76.8	68.1	42.2	85.4	72.0

(a)

Initialization	T	UL	LL	UA	LA	H	Total
Ours	89.7	62.1	48.2	36.0	16.0	54.2	51.0
Shuffle&Learn	90.4	62.7	45.7	33.3	11.8	52.0	49.3
Random init.	87.3	52.3	35.4	25.4	7.6	44.0	42.0
Alexnet	92.8	68.1	53.0	39.8	17.5	62.8	55.7

(b)

Table 2: The Evaluation of the unsupervised (a) and the semi supervised (b) experiment on the LSP dataset.

first group contains methods pre-trained on imagenet data. The second one contains unsupervised methods without pre-training and the last one is the comparison group with a fully supervised method and the theoretical upper limit for similarity learning. Poset learning gives the best results for the pre-trained and the completely unsupervised methods. Even though none of those methods id competitive to the supervised model.

To perform a semi-supervised experiment, the resulting network of poset learning is used as initialization for the training process of the supervised method "*DeepPose*" [10]. Again poset learning is the best unsupervised method in the experiment with a PCP of 51.0%. Therefore poset learning is indeed useful for pre-training for a supervised training process. (see table 2b). Here the differences between poset learning and shuffle&learn is smaller (< 2%). The unsupervised pre-training improves the performance of DeepPose by 7% (shuffle&learn) to 9% (poset learning) (see table 2b). For comparison with a supervised pre-training they include an approach with alexnet pre-trained on imagenet as initialization. This approach performs better than the unsupervised initializations but the authors point out, that pre-training on imagenet is an expensive task, that needs vast amounts of labelled data, that the unsupervised pre-training does not need. With this in mind a performance difference of 5% can probably count as competitive.

MPII Pose

On this dataset the authors only perform semi supervised learning. Following the experiment for LSP they use poset learning, triplet based learning and an alexnet pre-trained on imagenet as different initializations for the DeepPose method. The results show again, that pre-training on unsupervised data improves the performance of DeepPose significantly from 65.4% (random initialization) to 69.3% (triplet based) or 72.7% (poset learning). As well as in the LSP experiment, the pretraining on an imagenet

	Ours	Shuffle&Learn	Random Init.	AlexNet
Head	83.8	75.8	79.5 7	87.2
Neck	90.9	86.3	87.1	93.2
LR Shoulder	77.5	75.0	71.6	85.2
LR Elbow.	60.8	59.2	52.1	69.6
LR Wrist	44.4	42.2	34.6	52.0
LR Hip	74.6	73.3	64.1	81.3
LR Knee	65.4	63.1	58.3	69.7
LR Ankle	57.4	51.7	51.2	62.0
Thorax	90.5	87.1	85.5	93.4
Pelvis	81.3	79.5	70.1	86.6
Total	72.7	69.3	65.4	78.0

Table 3: Evaluation of the MPII Pose dataset.

trained alexnet gives superior results (78.0%) but again the difference is not unreasonable, according to the difference in needed labelled training data.

4 Conclusion

The authors showed, that their method is competitive with other state of the art unsupervised methods. For explainability this method is interesting in the sense that it gives meaningful fine grained similarities. As mentioned in the introduction those similarities are beneficial in many ways for explainability.

References

1. M. A. Bautista, A. Sanakoyeu, and B. Ommer. “Deep unsupervised similarity learning using partially ordered sets”. In: *Proceedings of IEEE Computer Vision and Pattern Recognition*. 2017.
2. M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer. “Cliquecnn: Deep unsupervised exemplar learning”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3846–3854.
3. C. Doersch, A. Gupta, and A. A. Efros. “Unsupervised visual representation learning by context prediction”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1422–1430.
4. A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. “Discriminative unsupervised feature learning with convolutional neural networks”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 766–774.

5. B. Hariharan, J. Malik, and D. Ramanan. “Discriminative decorrelation for clustering and classification”. In: *European Conference on Computer Vision*. Springer. 2012, pp. 459–472.
6. A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
7. T. Malisiewicz, A. Gupta, and A. A. Efros. “Ensemble of exemplar-svms for object detection and beyond”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 89–96.
8. I. Misra, C. L. Zitnick, and M. Hebert. “Shuffle and learn: unsupervised learning using temporal order verification”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 527–544.
9. Ö. Sümer, T. Dencker, and B. Ommer. “Self-supervised learning of pose embeddings from spatiotemporal relations in videos”. In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE. 2017, pp. 4308–4317.
10. A. Toshev and C. Szegedy. “Deeppose: Human pose estimation via deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1653–1660.