A Robust Distance Metric for Deep Metric Learning

authors: Ziyang Qiu and Thomas Rekers
Matrikelnummer: 2759427

betreut von Prof. Dr. Sebastian Herr

28. März 2018
Universität Bielefeld

Inhaltsverzeichnis

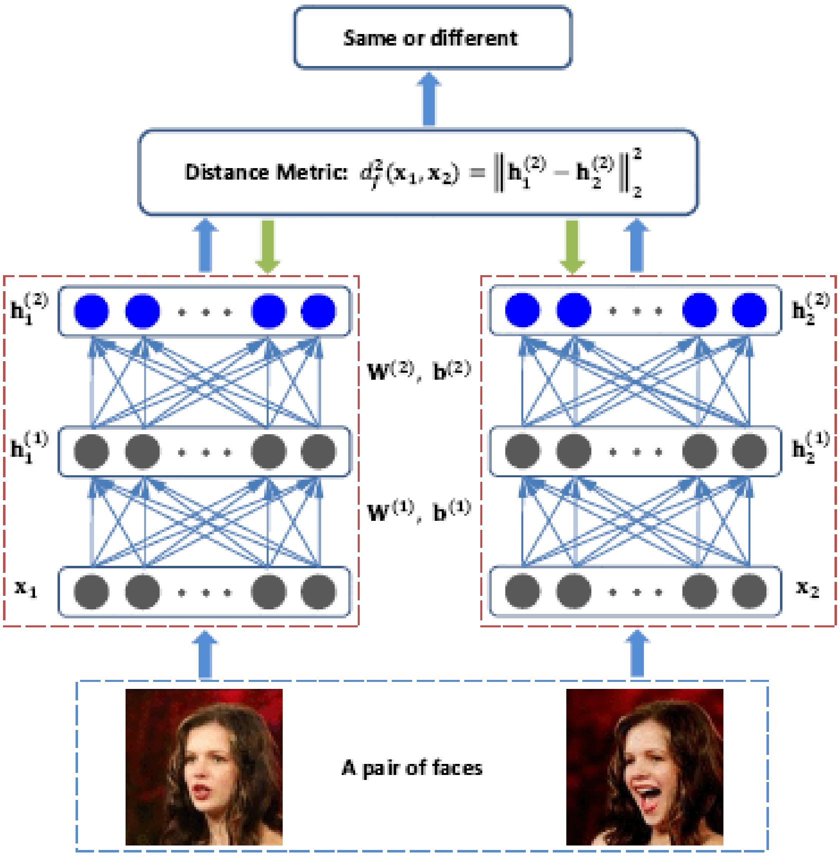
1 Executive Summary

Deep metric learning deals with the learning of discriminative features to process classification tasks and image clustering. It is also called similarity learning and gained increasing attention in recent years. The basic idea of deep metric learning is that the image input data is mapped by a neural network into a multidimensional feature space, where similar samples are mapped close to each other and dissimilar samples are mapped more distant to each other. This report is based on the paper Signal-to-Noise Ratio: A Robust Distance Metric for Deep Metric Learning. At first we will summarize the paper by explaining the two different approaches in metric learning, the structure learning and the distant learning, and introducing the robust SNR distance metric, which is based on Signal-to-Noise Ratio (SNR) for measuring the similarity of images. We give explanations regarding the modifications in the loss functions for training models with the SNR metric and discuss the results from the experiments based on Alexnet in the paper. Afterwards we introduce our own experiments based on Alexnet and Resnet and present the differences between these two network models. As explained in the paper the SNR metric can be considered as a relative Euclidian metric under special circumstances. In this case relative Euclidian metric means that the ratio between the Euclidian distance of two samples and the Euclidian distance of one sample to the origin of the feature space is returned. Therefore we decided to design an analogous relative metric not based on the Euclidian metric but on the Mahalanobis metric and call it relative Mahalanobis metric. We summarize the properties of the mahalanobis as described in the paper Distance Metric Learning for Large Margin Nearest Neighbor Classification and we present the resuts of our experiments including the self-designed relative Mahalanobis metric. Finally we discuss our results by comparing the scores of all metrics used in the experiments.

2 Introduction

The fundamental idea of metric learning is to map the input data, in most cases a set of images, into a multidimensional feature space. Similar samples are mapped close to each other in the feature space and dissimilar samples are mapped farther apart. This approach can be used both for classification and clustering tasks. While in conventional metric learning the images are mapped linearly into the feature, in deep metric learning we use deep neural network models like Alexnet or Resnet to get a nonlinear embedding of the input data.

Because of its great successes in last years deep metric learning is widely applied in face recognition, where the model has to be trained with a very large number of different classes. The contrary approach to deep metric learning is to design a network architecture where as many neurons are in the output layer as different classes are in the dataset. In this case, the model identifies each neuron in the output layer with one class of the dataset and the classification is done by selecting the neuron with the largest output. While this approach becomes problematically in case of a large number of different clas-


Same or different
Distance Metric: $d_f^2(\mathbf{x}_1,\mathbf{x}_2) = \left\|\mathbf{h}_1^{(2)} - \mathbf{h}_2^{(2)}\right\|_2^2$
$\mathbf{h}_1^{(2)}$
$\mathbf{h}_2^{(2)}$
$\mathbf{W}^{(2)}, \mathbf{b}^{(2)}$
$\mathbf{h}_1^{(1)}$
$\mathbf{h}_2^{(1)}$
$\mathbf{W}^{(1)}, \mathbf{b}^{(1)}$
$\mathbf{x}_1$
$\mathbf{x}_2$
A pair of faces

The flowchart shows the frame work of DML. A given pair of face images x_1 and x_2 are mapped into the same feature space as h_1 and h_2 through deep neural networks, where their similarity is calculated.

ses and a small number of samples per class, deep metric learning concepts still show good results.

In case of many different classes the number of classes is usually greater than the number of neurons in the output layer in deep metric learning. Therefore we cannot predict the class by identifying each class with one neuron. Instead the classification follows from the distances in the feature space. For example, the k-Nearest-Neighbor algorithm can be chosen to assign a sample to its class.

In deep metric learning we can distinguish between two different learning approaches. The first and most common approach is structure learning. Structure learning tries to construct more effective structures for the loss functions. For example, many neural network architectures include sampler functions which deliver a customized batch of samples from the feature space to the loss function. Varying the batch size or increasing the number of negative labeled samples in a batch can be strategies in structure learning. The loss function is designed according to the shape of the batch for the backward training. We will introduce such loss function and the shape of their batches in the preliminaries. It is to notice that all loss functions use metrics to get a semantic distance between an anchor sample and a comparison sample in the feature space. But structure learning

approaches just try to optimize the sampling for the loss function or the summation and weights inside of the loss function. The metric is not changed in structure learning but is changed in distance learning, what is the second approach of metric learning.
The most common learning processes use the Euclidian metric to calculate the distances of the samples in the feature space. In contrast, distance learning tries to get longer distances between similar and shorter distances between dissimilar sammples in the feature space by choosing a suitable metric inside of the loss function.

3 Preliminaries

3.1 Loss Functions

As mentioned before optimizing a neural network by tuning the objective function of the training process is part of structure learning. The loss functions introduced in this chapter are based on the Euclidian metric. We will see later that the loss functions for other metrics look similar, but in some cases we have to add a regularization term.

3.1.1 Contrastive Loss

The most common structure learning approach is constrastive embedding with its objective function contrastive loss. The idea is to introduce a margin so that dissimilar samples are pushed apart by this given margin. Similar samples are pulled as close as possible to each other. The objective function for contrastive loss is

$$L = \frac{1}{2N} \sum_{(i,j)\in\Gamma}^{N} \left(y d_{i,j}^2 + (1-y) \max\left(margin - d_{ij} \right)^2 \right) \,, \tag{1}$$

where d_{ij} is the Euclidean distance between the sample pairs in the feature space. We have $y = 1$ if two samples are in the same class and $y = 0$ otherwise. The $margin$ is the threshold to evaluate whether a pair of samples is in the same class or not. Therefore d_{ij} should be greater than the margin in case of a pair of unsimilar samples and less than the margin in case of two samples of the same class. Γ is the set of the samples and N is the size of the set.

3.1.2 Triplet Loss

Another common objective function is triplet loss. In triplet loss the loss function operates on a set of triplets, where each triplets consists of an anchor sample, a positive sample and a negative sample. Positive means that the considered sample has the same class as the anchor. Otherwise the sample is negative. The objective function affects that the distance between the anchor and the positive sample is smaller than the distance between the anchor and a negative sample with a margin. The formula for triplet loss is

$$L = \frac{1}{|\Gamma|} \sum_{(i,j,k)\in\Gamma} \max(d_{ij}^2 + m - d_{ik}^2, 0) \,, \tag{2}$$

where i is the index of the anchor, j the index of the positive sample and k the index of the negative sample. d is the Euclidian metric again and Γ the set of the samples.

3.1.3 N-Pair Loss

N-pair loss is a generalization of triplet loss by exploiting the possibility to interact with negative samples of all classes. This means that we deal with one positive sample from the same class as the anchor and with $N - 1$ negative samples, where every negative sample is from a different class of the $N - 1$ other classes. The formula for N-pair loss is

$$L = \frac{1}{N} \sum_{i=1}^{N} \log \left(1 + \sum_{j \neq i} \exp \left(f_i' f_j^+ - f_i' f_i^+ \right) \right) , \tag{3}$$

where $f_i = f(x_i)$ and $\{(x_i, x_i^+)\}_{i=1}^N$ are N pairs of input samples from N different classes with x_i as anchors and x_i^+ as their positive samples. It applies for their labels y_i that $y_i \neq y_j \, \forall \, i \neq j$ and therefore $\{x_j^+, j \neq i\}$ are the negative input samples to x_i.

3.1.4 Lifted Structured Loss

In lifted structured loss all negative samples are incorporated through training. The positive samples are pulled as close as possible to the anchor and the negative samples are pushed away farther than a given margin. The formula for structured lifted loss is

$$L = \frac{1}{2|P|} \sum_{(i,j) \in P} \max \left(L_{ij}, 0 \right)^2 \tag{4}$$

$$\text{with} \quad L_{ij} = \log \left(\sum_{(i,k) \in N} \exp \left(\alpha - d_{ik} \right) + \sum_{(j,l) \in N} \exp \left(\alpha - d_{jl} \right) \right) + d_{ij} \, , \tag{5}$$

where α is the margin parameter. N denotes the set of negative pairs and P denotes the set of positive pairs.

figure

3.2 Distance Metrics

Measuring similarities between pairs of samples have been a research topc for many years. The most well-known distance metric undoubtedly is Euclidian distance, fwhich has been widely used in learning discriminative embeddings. But Euclidean distance metric only measures the distance between paired samples in n-dimensional space, and fail to preserve ghe correlation and improve the robustness of the pairs. Some other distance metrics have been proposed to avoid the weakness of Euclidean distance. One of those is Mahalanobis Distance, which is based on correlations between varaibles by which different patterns can be identified and analyzed. Recently, many new improvements in metric learning have been proposed. To improve the generalization of the learned features, Chen et al. introduced the energy confusion metric to confuse the network metric. Taking the angular relationship between samples into account, Jian Wang et al. introduced an Angular Loss improving the robustness of objective against feature variance. A new framework is also being explored. For example, Xinshao Wang et al. proposed ranked list loss that could better make use of datasets by constructing lists consisting of weighted negative examples and positive examples mined from the training set. Moreover, instead of using distance metrics, Flood Sung et al. managed to teach the Relation Network to larn the metric by itself.

4 Approach of the Paper

4.1 Notations and Technical Terms

In deep metric learning a neural network can be considered as function f called learning function, that maps input data like image samples into a multidimensional feature space. We write $h = f(\theta, x)$ where x is an image sample, h is the embedded sample in the feature space and *theta* represents the parameters and weights of the neural network. Given two features h_i and h_j in the feature space, we consider h_i as anchor feature and h_j as compared feature. Then we call h_i signal and h_j noisy signal and define the noise n_{ij} in h_i and h_j as $n_{ij} = h_j - h_i$. The signal-to-noise ratio is defined as the ratio of signal vvariance to noise variance. Therefore we can define the SNR between h_i and h_j as

$$SNR_{i,j} = \frac{\mathrm{Var}(h_i)}{\mathrm{Var}(h_j - h_i)} = \frac{\mathrm{Var}(h_i)}{\mathrm{Var}(n_{ij})} \tag{6}$$

$$\text{with} \quad \mathrm{Var}(a) = \frac{\sum_{i=1}^{n}(a_i - \mu)^2}{n}, \tag{7}$$

where μ denotes the mean of sample a.

4.2 SNR-Based Metric

From the perspective of information theory, a greater signal variance represents more useful information. In contrast the variance of noise can be seen as a measure of useless information. As a result, the greater the SNR is, the greater is the ratio of useful information in ratio to useless information. The most important constraint to a distance metric in deep metric learning is that similar samples shall have a short distance, while dissimilar samples shall have a longer distance. Though, the SNR, as basing on the variance on the samples and their differences, assumes an independent Gaussian distribution in every component of the samples, what might be a wrong assumption in some cases. As we found in the SNR considerations, greater SNR indicates similarity and smaller SNR indicates dissimilarity. Therefore we define an SNR based distance measure as the reciprocal of the SNR and call it SNR metric d_S:

$$d_S(h_i, h_j) = \frac{1}{SNR_{i,j}} = \frac{\mathrm{Var}(n_{ij}\mathrm{Var}(h_i)}{}. \tag{8}$$

figure

In figure 3 we see a comparison between anchor signals and compared signals, that have different SNR distances to the anchor signals. The 32-dimensional anchor features are in a Gaussian distribution with mean zero and variance one. The compared signals are synthesized by adding Gaussian noises with mean zero and different variances σ^2 to the anchor features, where $\sigma^2 = \{0.2, 0.5, 1.0\}$. As we see in the figure, the longer the SNR distances are, the greater the differences between the anchor signals and the compared signals. We notice again that the SNR metric may not be suitable if the anchor samples and compared samples are not in Gaussian distribution or the components of the samples are not independent to each other.

4.3 Superiority Analysis

Now we want to compare the SNR distance to the Euclidean distance. Given two samples in an n-dimensional feature space. Then their Euclidean distance is defined as

$$d_E(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}\,. \tag{9}$$

Assuming a zero-mean-distribution in every component of the samples, two samples h_i and h_j in an n-dimensional feature space have the SNR distance

$$d_S(h_i, h_j) = \frac{\mathrm{Var}(n_{ij})}{\mathrm{Var}(h_i)} = \frac{\sum_{m=1}^{n}(h_{jm} - h_{im})^2}{\sum_{m=1}^{M} h_{im}^2} = \frac{d_E(h_i, h_j)^2}{d_E(h_i)^2}\,, \tag{10}$$

where $d_E(h_i)$ denotes the Euclidean distance from h_i to the origin zero of the feature space. We will continue with the assumption of a zero-mean-distribution. This assumption is guaranted by a regularization term that we will introduce later. Figure 4 visualizes the differences between Euclidean and SNR distance. We easily notice that the SNR distance gives us more confidence in the determination whether a pair of samples is similar or not. The Euclidean distance can be seen as an absolute error between the two samples by calculating only the distance from one point to another. On the other hand, the SNR distance can be seen as an relative error between the two samples by taking the distance to the origin of the feature space into account.

figure 4

Because the parameters of the neural network are optimized while training the learning function, we observe two forces that influence the mapping of the samples as shown in figure 5. The first force influences the intra-class distances of the samples. As we see in formula ..., by reducing the numerator $d_E(h_i, h_j)$, we also reduce the value of the SNR distance and the return of the objective function. Therefore the distance between samples of the same class decreases while training. The second force influences the inter-class disntaces. Because of the denominator $d_E(h_i)$, the distance to the mean of all samples increases regardless of the selected loss function. The pulling intra-classes-force and the pushing inter-classes-force effects a distribution in the feature space. Where samples of similar classes form close groups with far distance two the groups of other classes.

figure 5

4.4 SNR Distance and Correlation

Now we derive the relationship between the SNR distance and the correlation coefficient of two samples in the feature space. Assuming a zero-mean-distribution in every component of the samples and an independence of the noise to the signal feature, we get

$$\begin{aligned}
\mathrm{corr}(h_i, h_j) &= \frac{\mathrm{cov}(h_i, h_j)}{\sqrt{\mathrm{var}(h_i)\mathrm{var}(h_j)}} = \frac{\mathrm{E}(h_i, h_j)}{\sqrt{\mathrm{var}(h_i)\mathrm{var}(h_j)}} && (11) \\
&= \frac{\mathrm{E}\left(h_i\left(h_i + n_{ij}\right)\right)}{\sqrt{\mathrm{var}(h_i)\mathrm{var}(h_i + n_{ij})}} = \frac{\mathrm{E}(h_i)^2}{\sqrt{\mathrm{var}(h_i)\mathrm{var}(h_i + n_{ij})}} && (12) \\
&= \frac{\mathrm{var}(h_i)}{\sqrt{\mathrm{var}(h_i)^2 + \mathrm{var}(h_i)\mathrm{var}(n_{ij})}} = \frac{1}{\sqrt{1 + \frac{\mathrm{var}(n_{ij})}{\mathrm{var}(h_i)}}} && (13) \\
&= \frac{1}{\sqrt{1 + \frac{1}{SNR_{ij}}}} = \frac{1}{\sqrt{1 + d_s(h_i, h_j)}}\,. && (14)
\end{aligned}$$

As we see in figure 6, the SNR distance is between two samples in the feature space, the greater the correlation coefficient between these two samples.

figure 6

4.5 Deep SNR-based Metric Learning

bla