A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses

Malik Boudiaf[*1][0000−0003−2047−6447], Jérôme Rony[*1][0000−0002−6359−6142],
Imtiaz Masud Ziko[*1][0000−0002−9254−2426], Eric Granger[1][0000−0001−6116−7945],
Marco Pedersoli[1][0000−0002−7601−8640], Pablo Piantanida[2][0000−0002−8717−2117],
and Ismail Ben Ayed[1][0000−0002−9668−8027]

[1] Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle (LIVIA),
ÉTS Montreal, Canada
{malik.boudiaf.1, jerome.rony.1, imtiaz-masud.ziko.1}@etsmtl.net
[2] Laboratoire des Signaux et Systèmes (L2S),
CentraleSupelec-CNRS-Université Paris-Saclay, France

Abstract. Recently, substantial research efforts in Deep Metric Learning (DML) focused on designing complex pairwise-distance losses, which require convoluted schemes to ease optimization, such as sample mining or pair weighting. The standard cross-entropy loss for classification has been largely overlooked in DML. On the surface, the cross-entropy may seem unrelated and irrelevant to metric learning as it does not explicitly involve pairwise distances. However, we provide a theoretical analysis that links the cross-entropy to several well-known and recent pairwise losses. Our connections are drawn from two different perspectives: one based on an explicit optimization insight; the other on discriminative and generative views of the mutual information between the labels and the learned features. First, we explicitly demonstrate that the cross-entropy is an upper bound on a new pairwise loss, which has a structure similar to various pairwise losses: it minimizes intra-class distances while maximizing inter-class distances. As a result, minimizing the cross-entropy can be seen as an approximate *bound-optimization* (or *Majorize-Minimize*) algorithm for minimizing this pairwise loss. Second, we show that, more generally, minimizing the cross-entropy is actually equivalent to maximizing the mutual information, to which we connect several well-known pairwise losses. Furthermore, we show that various standard pairwise losses can be explicitly related to one another via bound relationships. Our findings indicate that the cross-entropy represents a proxy for maximizing the mutual information – as pairwise losses do – without the need for convoluted sample-mining heuristics. Our experiments[†] over four standard DML benchmarks strongly support our findings. We obtain state-of-the-art results, outperforming recent and complex DML methods.

Keywords: Metric Learning, Deep Learning, Information Theory

---

[*]Equal contributions

[†]Code available at: https://github.com/jeromerony/dml_cross_entropy

1   Introduction

The core task of metric learning consists in learning a metric from high-dimensional data, such that the distance between two points, as measured by this metric, reflects their semantic similarity. Applications of metric learning include image retrieval, zero-shot learning or person re-identification, among others. Initial attempts to tackle this problem tried to learn metrics directly on the input space [15]. Later, the idea of learning suitable embedding was introduced, with the goal of learning Mahalanobis distances [2, 5, 25, 38, 41], which corresponds to learning the best linear projection of the input space onto a lower-dimensional manifold, and using the Euclidean distance as a metric. Building on the embedding-learning ideas, several papers proposed to learn more complex mappings, either by kernelization of already existing linear algorithms [2], or by using a more complex hypothesis such as linear combinations of gradient boosted regressions trees [10].

The recent success of deep neural networks at learning complex, nonlinear mappings of high-dimensional data aligns with the problem of learning a suitable embedding. Following works on Mahalanobis distance learning, most Deep Metric Learning (DML) approaches are based on pairwise distances. Specifically, the current paradigm is to learn a deep encoder that maps points with high semantic similarity close to each other in the embedded space (w.r.t. pairwise Euclidean or cosine distances). This paradigm concretely translates into *pairwise losses* that encourage small distances for pairs of samples from the same class and large distances for pairs of samples from different classes. While such formulations seem intuitive, the practical implementations and optimization schemes for pairwise losses may become cumbersome, and randomly assembling pairs of samples typically results in slow convergence or degenerate solutions [8]. Hence, research in DML focused on finding efficient ways to reformulate, generalize and/or improve sample mining and/or sample weighting strategies over the existing pairwise losses. Popular pairwise losses include triplet loss and its derivatives [4,8,26,27,47], contrastive loss and its derivatives [6,37], Neighborhood Component Analysis and its derivatives [5,16,40], among others. However, such modifications are often heuristic-based, and come at the price of increased complexity and additional hyper-parameters, reducing the potential of these methods in real-world applications. Furthermore, the recent experimental study in [17] showed that the improvement brought by an abundant metric learning literature in the last 15 years is at best marginal when the methods are compared fairly.

Admittedly, the objective of learning a useful embedding of data points intuitively aligns with the idea of directly acting on the distances between pairs of points in the embedded space. Therefore, the standard cross-entropy loss, widely used in classification tasks, has been largely overlooked by the DML community, most likely due to its apparent irrelevance for Metric Learning [39]. As a matter of fact, why would anyone use a point-wise prediction loss to enforce pairwise-distance properties on the embedding space? Even though the cross-entropy was shown to be competitive for face recognition applications [13, 33, 34], to the best of our knowledge, only one paper empirically observed competitive results of a normalized, temperature-weighted version of the cross-entropy in the context of

deep metric learning [46]. However, the authors did not provide any theoretical insights for these results.

On the surface, the standard cross-entropy loss may seem unrelated to the pairwise losses used in DML. Here, we provide theoretical justifications that connect directly the cross-entropy to several well-known and recent pairwise losses. Our connections are drawn from two different perspectives; one based on an explicit optimization insight and the other on mutual-information arguments. We show that four of the most prominent pairwise metric-learning losses, as well as the standard cross-entropy, are maximizing a common underlying objective: the Mutual Information (MI) between the learned embeddings and the corresponding samples' labels. As sketched in Section 2, this connection can be intuitively understood by writing this MI in two different, but equivalent ways. Specifically, we establish tight links between pairwise losses and the *generative* view of this MI. We study the particular case of contrastive loss [6], explicitly showing its relation to this MI. We further generalize this reasoning to other DML losses by uncovering tight relations with contrastive loss. As for the cross-entropy, we demonstrate that the cross-entropy is an upper bound on an underlying pairwise loss – on which the previous reasoning can be applied – which has a structure similar to various existing pairwise losses. As a result, minimizing the cross-entropy can be seen as an approximate *bound-optimization* (or *Majorize-Minimize*) algorithm for minimizing this pairwise loss, implicitly minimizing intra-class distances and maximizing inter-class distances. We also show that, more generally, minimizing the cross-entropy is equivalent to maximizing the *discriminative* view of the mutual information. Our findings indicate that the cross-entropy represents a proxy for maximizing the mutual information, as pairwise losses do, without the need for complex sample-mining and optimization schemes. Our comprehensive experiments over four standard DML benchmarks (CUB200, Cars-196, Stanford Online Product and In-Shop) strongly support our findings. We consistently obtained state-of-the-art results, outperforming many recent and complex DML methods.

Summary of contributions

1. Establishing relations between several pairwise DML losses and a generative view of the mutual information between the learned features and labels;
2. Proving explicitly that optimizing the standard cross-entropy corresponds to an approximate bound-optimizer of an underlying pairwise loss;
3. More generally, showing that minimizing the standard cross-entropy loss is equivalent to maximizing a discriminative view of the mutual information between the features and labels.
4. Demonstrating state-of-the-art results with cross-entropy on several DML benchmark datasets.

Table 1. Definition of the random variables and information measures used in this paper.

General		Model	
Labeled dataset	$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$	Encoder	$\phi_{\mathcal{W}} : \mathcal{X} \to \mathcal{Z}$
Input feature space	$\mathcal{X}$	Soft-classifier $f_{\boldsymbol{\theta}} : \mathcal{Z} \to [0,1]^K$	
Embedded feature space	$\mathcal{Z} \subset \mathbb{R}^d$	**Random variables (RVs)**	
Label/Prediction space	$\mathcal{Y} \subset \mathbb{R}^K$	Data	X, Y
Euclidean distance	$D_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2$	Embedding	$\widehat{Z}\|X \sim \phi_{\mathcal{W}}(X)$
Cosine distance	$D_{ij}^{\cos} = \frac{\boldsymbol{z}_i^{\mathsf{T}} \boldsymbol{z}_j}{\|\boldsymbol{z}_i\|\|\boldsymbol{z}_j\|}$	Prediction	$\widehat{Y}\|\widehat{Z} \sim f_{\boldsymbol{\theta}}(\widehat{Z})$

Information measures	
Entropy of Y	$\mathcal{H}(Y) := \mathbb{E}_{p_Y}\left[-\log p_Y(Y)\right]$
Conditional entropy of Y given Z	$\mathcal{H}(Y\|\widehat{Z}) := \mathbb{E}_{p_{Y\widehat{Z}}}\left[-\log p_{Y\|\widehat{Z}}(Y\|\widehat{Z})\right]$
Cross entropy (CE) between Y and $\widehat{Y}$	$\mathcal{H}(Y;\widehat{Y}) := \mathbb{E}_{p_Y}\left[-\log p_{\widehat{Y}}(Y)\right]$
Conditional CE given $\widehat{Z}$	$\mathcal{H}(Y;\widehat{Y}\|\widehat{Z}) := \mathbb{E}_{p_{\widehat{Z}Y}}\left[-\log p_{\widehat{Y}\|\widehat{Z}}(Y\|\widehat{Z})\right]$
Mutual information between $\widehat{Z}$ and Y	$\mathcal{I}(\widehat{Z};Y) := \mathcal{H}(Y) - \mathcal{H}(Y\|\widehat{Z})$

2 On the two views of the mutual information

The Mutual Information (MI) is a well known-measure designed to quantify the amount of information shared by two random variables. Its formal definition is presented in Table 1. Throughout this work, we will be particularly interested in $\mathcal{I}(\widehat{Z};Y)$ which represents the MI between learned features $\widehat{Z}$ and labels Y. Due to its symmetry property, the MI can be written in two ways, which we will refer to as the *discriminative view* and *generative view* of MI:

$$\mathcal{I}(\widehat{Z};Y) = \underbrace{\mathcal{H}(Y) - \mathcal{H}(Y|\widehat{Z})}_{\text{discriminative view}} = \underbrace{\mathcal{H}(\widehat{Z}) - \mathcal{H}(\widehat{Z}|Y)}_{\text{generative view}} \tag{1}$$

While being analytically equivalent, these two views present two different, complementary interpretations. In order to maximize $\mathcal{I}(\widehat{Z};Y)$, the discriminative view conveys that the labels should be balanced (out of our control) and easily identified from the features. On the other hand, the generative view conveys that the features learned should spread as much as possible in the feature space, while keeping samples sharing the same class close to each other. Hence, the discriminative view is more focused on label identification, while the generative view focuses on more explicitly shaping the distribution of the features learned by the model. Therefore, the MI enables us to draw links between classification losses (*e.g.* cross-entropy) and feature-shaping losses (including all the well-known pairwise metric learning losses).

3 Pairwise losses and the generative view of the MI

In this section, we study four pairwise losses used in the DML community: center loss [39], contrastive loss [6], Scalable Neighbor Component Analysis (SNCA) loss [40] and Multi-Similarity (MS) loss [37]. We show that these losses can be interpreted as proxies for maximizing the generative view of mutual information $\mathcal{I}(\widehat{Z}; Y)$. We begin by analyzing the specific example of contrastive loss, establishing its tight link to the MI, and further generalize our analysis to the other pairwise losses (see Table 2). Furthermore, we show that these pairwise metric-learning losses can be explicitly linked to one another via bound relationships.

3.1 The example of contrastive loss

We start by analyzing the representative example of contrastive loss [6]. For a given margin $m \in \mathbb{R}^+$, this loss is formulated as:

$$\mathcal{L}_{\text{contrast}} = \underbrace{\frac{1}{n} \sum_{i=1}^{n} \sum_{j:y_j=y_i} D_{ij}^2}_{T_{\text{contrast}}} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} \sum_{j:y_j \neq y_i} [m - D_{ij}]_+^2}_{C_{\text{contrast}}} \tag{2}$$

where $[x]_+ = \max(0, x)$. This loss naturally breaks down into two terms: a *tightness* part T_{contrast} and a *contrastive* part C_{contrast}. The tightness part encourages samples from the same class to be close to each other and form *tight* clusters. As for the *contrastive* part, it forces samples from different classes to stand far apart from one another in the embedded feature space. Let us analyze these two terms from a mutual-information perspective.

As shown in the next subsection, the tightness part of contrastive loss is equivalent to the tightness part of the center loss [39]: $T_{\text{contrast}} \stackrel{\text{c}}{=} T_{\text{center}} = \frac{1}{2} \sum_{i=1}^{n} \|z_i - c_{y_i}\|^2$, where $c_k = \frac{1}{|\mathcal{Z}_k|} \sum_{z \in \mathcal{Z}_k} z$ denotes the mean of feature points from class k in embedding space $\mathcal{Z}$ and symbol $\stackrel{\text{c}}{=}$ denotes equality up to a multiplicative and/or additive constant. Written in this way, we can interpret T_{contrast} as a conditional cross entropy between $\widehat{Z}$ and another random variable $\bar{Z}$, whose conditional distribution given Y is a standard Gaussian centered around c_Y: $\bar{Z}|Y \sim \mathcal{N}(c_Y, I)$:

$$T_{\text{contrast}} \stackrel{\text{c}}{=} \mathcal{H}(\widehat{Z}; \bar{Z}|Y) = \mathcal{H}(\widehat{Z}|Y) + \mathcal{D}_{KL}(\widehat{Z}||\bar{Z}|Y) \tag{3}$$

As such, T_{contrast} is an upper bound on the conditional entropy that appears in the mutual information:

$$T_{\text{contrast}} \geq \mathcal{H}(\widehat{Z}|Y) \tag{4}$$

This bound is tight when $\widehat{Z}|Y \sim \mathcal{N}(c_Y, I)$. Hence, minimizing T_{contrast} can be seen as minimizing $\mathcal{H}(\widehat{Z}|Y)$, which exactly encourages the encoder $\phi_{\mathcal{W}}$ to produce low-entropy (=compact) clusters in the feature space for each given class. Notice

that using this term only will inevitably lead to a trivial encoder that maps all data points in $\mathcal{X}$ to a single point in the embedded space $\mathcal{Z}$, hence achieving a global optimum.

To prevent such a trivial solution, a second term needs to be added. This second term – that we refer to as the *contrastive* term – is designed to push each point away from points that have a different label. In this term, only pairs such that $D_{ij} \leq m$ produce a cost. Given a pair (i, j), let us define $x = D_{ij}/m$. Given that $x \in [0, 1]$, one can show the following: $1 - 2x \leq (1 - x)^2 \leq 1 - x$. Using linear approximation $(1 - x)^2 \approx 1 - 2x$ (with error at most x), we obtain:

$$C_{\mathrm{contrast}} \overset{\mathrm{c}}{\approx} -\frac{2m}{n} \sum_{i=1}^{n} \sum_{j:y_j \neq y_i} D_{ij} = -\frac{2m}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} + \frac{2m}{n} \sum_{i=1}^{n} \sum_{j:y_j = y_i} D_{ij} \quad (5)$$

While the second term in Eq. 5 is redundant with the tightness objective, the first term is close to the differential entropy estimator proposed in [36]:

$$\widehat{\mathcal{H}}(\widehat{Z}) = \frac{d}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} \log D_{ij}^2 \overset{\mathrm{c}}{=} \sum_{i=1}^{n} \sum_{j=1}^{n} \log D_{ij} \quad (6)$$

Both terms measure the spread of $\widehat{Z}$, even though they present different gradient dynamics. All in all, minimizing the whole contrastive loss can be seen as a proxy for maximizing the MI between the labels Y and the embedded features $\widehat{Z}$:

$$\mathcal{L}_{\mathrm{contrast}} = \underbrace{\frac{1}{n} \sum_{i=1}^{n} \sum_{j:y_j = y_i} (D_{ij}^2 + 2mD_{ij})}_{\propto \mathcal{H}(\widehat{Z}|Y)} - \underbrace{\frac{2m}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij}}_{\propto \mathcal{H}(\widehat{Z})} \quad \propto \; -\mathcal{I}(\widehat{Z}; Y) \quad (7)$$

3.2   Generalizing to other pairwise losses

A similar analysis can be carried out on other, more recent metric learning losses. More specifically, they can also be broken down into two parts: a *tightness* part that minimizes intra-class distances to form compact clusters, which is related to the *conditional entropy* $\mathcal{H}(\widehat{Z}|Y)$, and a second *contrastive* part that prevents trivial solutions by maximizing inter-class distances, which is related to the *entropy* of features $\mathcal{H}(\widehat{Z})$. Note that, in some pairwise losses, there might be some redundancy between the two terms, *i.e.*, the tightness term also contains some contrastive subterm, and vice-versa. For instance, the cross-entropy loss is used as the contrastive part of the center-loss but, as we show in Section 4.2, the cross-entropy, used alone, already contains both tightness (conditional entropy) and contrastive (entropy) parts. Table 2 presents the split for four DML losses. The rest of the section is devoted to exhibiting the close relationships between several pairwise losses and the tightness and contrastive terms (*i.e.*, T and C).

Links between losses: In this section, we show that the tightness and contrastive parts of the pairwise losses in Table 2, even though different at first sight, can actually be related to one another.

Table 2. Several well-known and/or recent DML losses broken into a *tightness* term and a *contrastive* term. Minimizing the cross-entropy corresponds to an approximate bound optimization of PCE.

Loss	Tightness part $\propto \mathcal{H}(\widehat{Z}|Y)$	Contrastive part $\propto \mathcal{H}(\widehat{Z})$
Center [39]	$\dfrac{1}{2}\sum_{i=1}^{n}\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|^2$	$-\dfrac{1}{n}\sum_{i=1}^{n}\log p_{iy_i}$
Contrast [6]	$\dfrac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j=y_i}D_{ij}^2$	$\dfrac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j\neq y_i}[m - D_{ij}]_+^2$
SNCA [40]	$-\dfrac{1}{n}\sum_{i=1}^{n}\log\left[\sum_{j:y_j=y_i}\exp\dfrac{D_{ij}^{\cos}}{\sigma}\right]$	$\dfrac{1}{n}\sum_{i=1}^{n}\log\left[\sum_{k\neq i}\exp\dfrac{D_{ik}^{\cos}}{\sigma}\right]$
MS [37]	$\dfrac{1}{n}\sum_{i=1}^{n}\dfrac{1}{\alpha}\log\left[1 + \sum_{j:y_j=y_i}e^{-\alpha(D_{ij}^{\cos}-m)}\right]$	$\dfrac{1}{n}\sum_{i=1}^{n}\dfrac{1}{\beta}\log\left[1 + \sum_{j:y_j\neq y_i}e^{\beta(D_{ij}^{\cos}-m)}\right]$
PCE Prop. 1	$-\dfrac{1}{2\lambda n^2}\sum_{i=1}^{n}\sum_{j:y_j=y_i}\boldsymbol{z}_i^{\mathsf{T}}\boldsymbol{z}_j$	$\dfrac{1}{n}\sum_{i=1}^{n}\log\left[\sum_{k=1}^{K}\exp\left[\dfrac{1}{\lambda n}\sum_{j=1}^{n}p_{jk}\boldsymbol{z}_i^{\mathsf{T}}\boldsymbol{z}_j\right]\right] - \dfrac{1}{2K^2\lambda^2}\sum_{k=1}^{K}\|\boldsymbol{c}_k^s\|^2$

Lemma 1. *Let T_A denote the tightness part of the loss from method A. Assuming that features are ℓ_2-normalized, and that classes are balanced, the following relations between Center [39], Contrastive [6], SNCA [40] and MS [37] losses hold:*

$$T_{SNCA} \overset{c}{\leq} T_{Center} \overset{c}{=} T_{Contrastive} \overset{c}{\leq} T_{MS} \tag{8}$$

Where $\overset{c}{\leq}$ stands for lower than, up to a multiplicative and an additive constant, and $\overset{c}{=}$ stands for equal to, up to a multiplicative and an additive constant.

The detailed proof of Lemma 1 is deferred to the supplemental material. As for the contrastive parts, we show in the supplemental material that both C_{SNCA} and C_{MS} are lower bounded by a common contrastive term that is directly related to $\mathcal{H}(\hat{Z})$. We do not mention the *contrastive* term of center-loss, as it represents the cross-entropy loss, which is exhaustively studied in Section 4.

4 Cross-entropy does it all

We now completely change gear to focus on the widely used *unary* classification loss: cross-entropy. On the surface, the cross-entropy may seem unrelated to metric-learning losses as it does not involve pairwise distances. We show that a close relationship exists between these pairwise losses widely used in deep metric

learning and the cross-entropy classification loss. This link can be drawn from two different perspectives, one is based on an explicit optimization insight and the other is based on a discriminative view of the mutual information. First, we explicitly demonstrate that the cross-entropy is an upper bound on a new pairwise loss, which has a structure similar to all the metric-learning losses listed in Table 2, *i.e.*, it contains a tightness term and a contrastive term. Hence, minimizing the cross-entropy can be seen as an approximate *bound-optimization (or Majorize-Minimize)* algorithm for minimizing this pairwise loss. Second, we show that, more generally, minimization of the cross-entropy is actually equivalent to maximization of the mutual information, to which we connected various DML losses. These findings indicate that the cross-entropy represents a proxy for maximizing $\mathcal{I}(\widehat{Z}, Y)$, just like pairwise losses, without the need for dealing with the complex sample mining and optimization schemes associated to the latter.

4.1   The pairwise loss behind unary cross-entropy

Bound optimization: Given a function $f(\mathcal{W})$ that is either intractable or hard to optimize, bound optimizers are iterative algorithms that instead optimize auxiliary functions (upper bounds on f). These auxiliary functions are usually more tractable than the original function f. Let t be the current iteration index, then a_t is an auxiliary function if:

$$
\begin{aligned}
f(\mathcal{W}) &\leq a_t(\mathcal{W}) \quad , \forall\ \mathcal{W} \\
f(\mathcal{W}_t) &= a_t(\mathcal{W}_t)
\end{aligned}
\tag{9}
$$

A bound optimizer follows a two-step procedure: first an auxiliary function a_t is computed, then a_t is minimized, such that:

$$
\mathcal{W}_{t+1} = \underset{\mathcal{W}}{\arg\min}\, a_t(\mathcal{W})
\tag{10}
$$

This iterative procedure is guaranteed to decrease the original function f:

$$
f(\mathcal{W}_{t+1}) \leq a_t(\mathcal{W}_{t+1}) \leq a_t(\mathcal{W}_t) = f(\mathcal{W}_t)
\tag{11}
$$

Note that bound optimizers are widely used in machine learning. Examples of well-known bound optimizers include the concave-convex procedure (CCCP) [45], expectation maximization (EM) algorithms or submodular-supermodular procedures (SSP) [18]. Such optimizers are particularly used in clustering [30] and, more generally, in problems involving latent-variable optimization.

   Pairwise Cross-Entropy: We now prove that minimizing cross-entropy can be viewed as an approximate bound optimization of a more complex pairwise loss.

Proposition 1. *Alternately minimizing the cross-entropy loss $\mathcal{L}_{CE}$ with respect to the encoder's parameters $\mathcal{W}$ and the classifier's weights $\boldsymbol{\theta}$ can be viewed as an*

approximate bound-optimization of a Pairwise Cross-Entropy (PCE) loss, which we define as follows:

$$\mathcal{L}_{PCE} = \underbrace{-\frac{1}{2\lambda n^2} \sum_{i=1}^{n} \sum_{j:y_j=y_i} \boldsymbol{z}_i^{\mathsf{T}} \boldsymbol{z}_j}_{\text{TIGHTNESS PART}} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} e^{\frac{1}{\lambda n} \sum_{j=1}^{n} p_{jk} \boldsymbol{z}_i^{\mathsf{T}} \boldsymbol{z}_j} - \frac{1}{2\lambda} \sum_{k=1}^{K} \|\boldsymbol{c}_k^s\|^2}_{\text{CONTRASTIVE PART}}$$

$$(12)$$

Where $\boldsymbol{c}_k^s = \frac{1}{n} \sum_{i=1}^{n} p_{ik} \boldsymbol{z}_i$ represents the soft-mean of class k, p_{ik} represents the softmax probability of point z_i belonging to class k, and $\lambda \in \mathbb{R}, \lambda > 0$ depends on the encoder $\phi_{\mathcal{W}}$.

The full proof of Proposition 1 is provided in the supplemental material. We hereby provide a quick sketch. Considering the usual softmax parametrization for our model's predictions $\widehat{Y}$, the idea is to break the cross-entropy loss in two terms, and artificially add and remove the regularization term $\frac{\lambda}{2} \sum_{k=1}^{K} \boldsymbol{\theta}_k^{\mathsf{T}} \boldsymbol{\theta}_k$:

$$\mathcal{L}_{\mathrm{CE}} = \underbrace{-\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\theta}_{y_i}^{\mathsf{T}} \boldsymbol{z}_i + \frac{\lambda}{2} \sum_{k} \boldsymbol{\theta}_k^{\mathsf{T}} \boldsymbol{\theta}_k}_{f_1(\boldsymbol{\theta})} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} e^{\boldsymbol{\theta}_k^{\mathsf{T}} \boldsymbol{z}_i} - \frac{\lambda}{2} \sum_{k=1}^{K} \boldsymbol{\theta}_k^{\mathsf{T}} \boldsymbol{\theta}_k}_{f_2(\boldsymbol{\theta})} \quad (13)$$

By properly choosing $\lambda \in \mathbb{R}$ in Eq. (13), both f_1 and f_2 become convex functions of $\boldsymbol{\theta}$. For any class k, we then show that the optimal values of $\boldsymbol{\theta}_k$ for f_1 and f_2 are proportional to, respectively, the hard mean $\boldsymbol{c}_k = \frac{1}{|\mathcal{Z}_k|} \sum_{i:y_i=k} \boldsymbol{z}_i$ and the soft mean $\boldsymbol{c}_k^s = \frac{1}{n} \sum_{i=1}^{n} p_{ik} \boldsymbol{z}_i$ of class k. By plugging-in those optimal values, we can lower bound f_1 and f_2 individually in Eq. 13 and get the result.

Proposition 1 casts a new light on the cross-entropy loss by explicitly relating it to a new pairwise loss (PCE), following the intuition that the optimal weights $\boldsymbol{\theta}^*$ of the final layer, *i.e.*, the linear classifier, are related to the centroids of each class in the embedded feature space $\mathcal{Z}$. Specifically, finding the optimal classifier's weight $\boldsymbol{\theta}^*$ for cross-entropy can be interpreted as building an auxiliary function $a_t(\mathcal{W}) = \mathcal{L}_{CE}(\mathcal{W}, \boldsymbol{\theta}^*)$ on $\mathcal{L}_{PCE}(\mathcal{W})$. Subsequently minimizing cross-entropy w.r.t. the encoder's weights $\mathcal{W}$ can be interpreted as the second step of bound optimization on $\mathcal{L}_{PCE}(\mathcal{W})$. Similarly to other metric learning losses, PCE contains a *tightness* part that encourages samples from the same classes to align with one another. In echo to Lemma 1, this tightness term, noted T_{PCE}, is equivalent, up to multiplicative and additive constants, to T_{center} and T_{contrast}, when the features are assumed to be normalized:

$$T_{\mathrm{PCE}} \stackrel{\mathrm{c}}{=} T_{\mathrm{center}} \stackrel{\mathrm{c}}{=} T_{\mathrm{contrast}} \quad (14)$$

PCE also contains a *contrastive* part, divided into two terms. The first pushes all samples away from one another, while the second term forces soft means c_k^s far from the origin. Hence, minimizing the cross-entropy can be interpreted as implicitly minimizing a pairwise loss whose structure appears similar to the well-established metric-learning losses in Table 2.

Simplified Pairwise Cross-Entropy: While PCE brings interesting theoretical insights, the computation of the parameter λ at every iteration requires computating the eigenvalues of a $d \times d$ matrix at every iteration (cf. full proof in supplemental material), which makes the implementation of PCE difficult in practice. In order to remove the dependence upon λ, one can plug in the same $\boldsymbol{\theta}$ for both f_1 and f_2 in Eq. 13. We choose to use $\boldsymbol{\theta}_1^* = \arg\min_{\boldsymbol{\theta}} f_1(\boldsymbol{\theta}) \propto [\boldsymbol{c}_1, ..., \boldsymbol{c}_K]^\mathsf{T}$. This yields a simplified version of PCE, that we call SPCE:

$$\mathcal{L}_{SPCE} = \underbrace{-\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j:y_j=y_i}\boldsymbol{z}_i^\mathsf{T}\boldsymbol{z}_j}_{\text{TIGHTNESS}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\log\sum_{k=1}^{K}\exp\left(\frac{1}{n}\sum_{j:y_j=k}\boldsymbol{z}_i^\mathsf{T}\boldsymbol{z}_j\right)}_{\text{CONTRASTIVE}} \qquad (15)$$

SPCE and PCE are similar (the difference is that PCE was derived after plugging in the soft means instead of hard means in f_2). Contrary to PCE, however, SPCE is easily computable, and the preliminary experiments we provide in the supplementary material indicate that CE and SPCE exhibit similar behaviors at training time. Interestingly, our derived SPCE loss has a form similar to contrastive learning losses in unsupervised representation learning [1, 20, 31].

4.2   A discriminative view of mutual information

Lemma 2. *Minimizing the conditional cross-entropy loss, denoted by $\mathcal{H}(Y; \widehat{Y}|\widehat{Z})$, is equivalent to maximizing the mutual information $\mathcal{I}(\widehat{Z}; Y)$.*

The proof of Lemma 2 is provided in the supplementary material. Such result is compelling. Using the discriminative view of mutual information allows to show that minimizing cross-entropy loss is equivalent to maximizing the mutual information $\mathcal{I}(\widehat{Z}; Y)$. This information theoretic argument reinforces our conclusion from Proposition 1 that cross-entropy and the previously described metric learning losses are essentially doing the same job.

4.3   Then why would cross-entropy work better?

We showed that cross-entropy essentially optimizes the same underlying mutual information $\mathcal{I}(\widehat{Z}; Y)$ as other DML losses. This fact alone is not enough to explain why the cross-entropy is able to consistently achieve better results than DML losses as shown in Section 5. We argue that the difference is in the optimization process. On the one hand, pairwise losses require careful sample mining and weighting strategies to obtain the most informative pairs, especially when considering minibatches, in order to achieve convergence in a reasonable amount of time, using a reasonable amount of memory. On the other hand, optimizing cross-entropy is substantially easier as it only implies minimization of unary terms. Essentially, cross-entropy does it all without dealing with the difficulties of pairwise terms. Not only it makes optimization easier, but also it simplifies the implementation, thus increasing its potential applicability in real-world problems.

Table 3. Summary of the datasets used for evaluation in metric learning.

Name	Objects	Categories	Images
Caltech-UCSD Birds-200-2011 (CUB) [32]	Birds	200	11 788
Cars Dataset [12]	Cars	196	16 185
Stanford Online Products (SOP) [27]	House furniture	22 634	120 053
In-shop Clothes Retrieval [14]	Clothes	7 982	52 712

5 Experiments

5.1 Metric

The most common metric used in DML is the recall. Most methods, especially recent ones, use the cosine distance to compute the recall for the evaluation. They include ℓ_2 normalization of the features in the model [4,16,19,21,23,35,37,42–44, 46], which makes cosine and Euclidean distances equivalent. Computing cosine similarity is also more memory efficient and typically leads to better results [24]. For these reasons, the Euclidean distance on non normalized features has rarely been used for both training and evaluation. In our experiments, ℓ_2-normalization of the features during training actually hindered the final performance, which might be explained by the fact that we add a classification layer on top of the feature extractor. Thus, we did not ℓ_2-normalize the features during training and reported the recall with both Euclidean and cosine distances.

5.2 Datasets

Four datasets are commonly used in metric learning to evaluate the performances. These datasets are summarized in Table 3. CUB [32], Cars [12] and SOP [27] datasets are divided into train and evaluation splits. For the evaluation, the recall is computed between each sample of the evaluation set and the rest of the set. In-Shop [14] is divided into a query and a gallery set. The recall is computed between each sample of the query set and the whole gallery set.

5.3 Training specifics

Model architecture and pre-training: In the metric learning literature, several architectures have been used, which historically correspond to the state-of-the-art image classification architectures on ImageNet [3], with an additional constraint on model size (*i.e.*, the ability to train on one or two GPUs in a reasonable time). These include GoogLeNet [28] as in [11], BatchNorm-Inception [29] as in [37] and ResNet-50 [7] as in [43]. They have large differences in classification performances on ImageNet, but the impact on performances over DML benchmarks has rarely been studied in controlled experiments. As this is not the focus of our paper, we use ResNet-50 for our experiments. We concede that one may obtain better performances by modifying the architecture (*e.g.*, reducing model

stride and performing multi-level fusion of features). Here, we limit our comparison to standard architectures. Our implementation uses the PyTorch [22] library, and initializes the ResNet-50 model with weights pre-trained on ImageNet.

Sampling: To the best of our knowledge, all DML papers – including [46] – use a form of pairwise sampling to ensure that, during training, each mini-batch contains a fixed number of classes and samples per class (*e.g.* mini-batch size of 75 with 3 classes and 25 samples per class in [46]). Deviating from that, we use the common random sampling among all samples (as in most classification training schemes) and set the mini-batch size to 128 in all experiments (contrary to [37] in which the authors use a mini-batch size of 80 for CUB, 1 000 for SOP and did not report for Cars and In-Shop).

Data Augmentation: As is common in training deep learning models, data augmentation improves the final performances of the methods. For CUB, the images are first resized so that their smallest side has a length of 256 (*i.e.*, keeping the aspect ratio) while for Cars, SOP and In-Shop, the images are resized to 256×256. Then a patch is extracted at a random location and size, and resized to 224×224. For CUB and Cars, we found that random jittering of the brightness, contrast and saturation slightly improves the results. All of the implementation details can be found in the publicly available code.

Cross-entropy: The focus of our experiments is to show that, with careful tuning, it is possible to obtain similar or better performance than most recent DML methods, while using only the cross-entropy loss. To train with the cross-entropy loss, we add a linear classification layer (with bias) on top of the feature extraction – similar to many classification models – which produces logits for all the classes present in the training set. Both the weights and biases of this classification layer are initialized to **0**. We also add dropout with a probability of 0.5 before this classification layer. To further reduce overfitting, we use label smoothing for the target probabilities of the cross-entropy. We set the probability of the true class to $1 - \epsilon$ and the probabilities of the other classes to $\frac{\epsilon}{K-1}$ with $\epsilon = 0.1$ in all our experiments.

Optimizer: In most DML papers, the hyper-parameters of the optimizer are the same for Cars, SOP and In-Shop whereas, for CUB, the methods typically use a smaller learning rate. In our experiments, we found that the best results were obtained by tuning the learning rate on a per dataset basis. In all experiments, the models are trained with SGD with Nesterov acceleration and a weight decay of 0.0005, which is applied to convolution and fully-connected layers' weights (but not to biases) as in [9]. For CUB and Cars, the learning rate is set to 0.02 and 0.05 respectively, with 0 momentum. For both SOP and In-Shop, the learning rate is set to 0.003 with a momentum of 0.99.

Batch normalization: Following [37], we freeze all the batch normalization layers in the feature extractor. For Cars, SOP and In-Shop, we found that adding batch normalization – without scaling and bias – on top of the feature extractor improves our final performance and reduces the gap between ℓ_2 and cosine distances when computing the recall. On CUB, however, we obtained the best recall without this batch normalization.

Table 4. Performance on CUB200, Cars-196, SOP and In-Shop datasets. d refers to the distance used to compute the recall when evaluating.

	Method	d	Architecture	Recall at					
				1	2	4	8	16	32
Caltech-UCSD Birds-200-2011	Lifted Structure [27]	ℓ_2	GoogLeNet	47.2	58.9	70.2	80.2	89.3	93.2
	Proxy-NCA [16]	cos	BN-Inception	49.2	61.9	67.9	81.9	-	-
	HTL [4]	cos	GoogLeNet	57.1	68.8	78.7	86.5	92.5	95.5
	ABE [11]	cos	GoogLeNet	60.6	71.5	79.8	87.4	–	–
	HDC [44]	cos	GoogLeNet	60.7	72.4	81.9	89.2	93.7	96.8
	DREML [42]	cos	ResNet-18	63.9	75.0	83.1	89.7	–	–
	EPSHN [43]	cos	ResNet-50	64.9	75.3	83.5	–	–	–
	NormSoftmax [46]	cos	ResNet-50	65.3	76.7	85.4	91.8	–	–
	Multi-Similarity [37]	cos	BN-Inception	65.7	77.0	86.6	91.2	95.0	97.3
	D&C [23]	cos	ResNet-50	65.9	76.6	84.4	90.6	–	–
	Cross-Entropy	ℓ_2	ResNet-50	67.6	78.1	85.6	91.1	94.7	97.2
		cos		69.2	79.2	86.9	91.6	95.0	97.3
				1	2	4	8	16	32
Stanford Cars	Lifted Structure [27]	ℓ_2	GoogLeNet	49.0	60.3	72.1	81.5	89.2	92.8
	Proxy-NCA [16]	cos	BN-Inception	73.2	82.4	86.4	88.7	–	–
	HTL [44]	cos	GoogLeNet	81.4	88.0	92.7	95.7	97.4	99.0
	EPSHN [43]	cos	ResNet-50	82.7	89.3	93.0	–	–	–
	HDC [44]	cos	GoogLeNet	83.8	89.8	93.6	96.2	97.8	98.9
	Multi-Similarity [37]	cos	BN-Inception	84.1	90.4	94.0	96.5	98.0	98.9
	D&C [23]	cos	ResNet-50	84.6	90.7	94.1	96.5	–	–
	ABE [11]	cos	GoogLeNet	85.2	90.5	94.0	96.1	–	–
	DREML [42]	cos	ResNet-18	86.0	91.7	95.0	97.2	–	–
	NormSoftmax [46]	cos	ResNet-50	89.3	94.1	96.4	98.0	–	–
	Cross-Entropy	ℓ_2	ResNet-50	89.1	93.7	96.5	98.1	99.0	99.4
		cos		89.3	93.9	96.6	98.4	99.3	99.7
				1	10	100	1000		
Stanford Online Product	Lifted Structure [27]	ℓ_2	GoogLeNet	62.1	79.8	91.3	97.4		
	HDC [44]	cos	GoogLeNet	70.1	84.9	93.2	97.8		
	HTL [4]	cos	GoogLeNet	74.8	88.3	94.8	98.4		
	D&C [23]	cos	ResNet-50	75.9	88.4	94.9	98.1		
	ABE [11]	cos	GoogLeNet	76.3	88.4	94.8	98.2		
	Multi-Similarity [37]	cos	BN-Inception	78.2	90.5	96.0	98.7		
	EPSHN [43]	cos	ResNet-50	78.3	90.7	96.3	–		
	NormSoftmax [46]	cos	ResNet-50	79.5	91.5	96.7	–		
	Cross-Entropy	ℓ_2	ResNet-50	80.8	91.2	95.7	98.1		
		cos		81.1	91.7	96.3	98.8		
				1	10	20	30	40	50
In-Shop Clothes Retrieval	HDC [44]	cos	GoogLeNet	62.1	84.9	89.0	91.2	92.3	93.1
	DREML [42]	cos	ResNet-18	78.4	93.7	95.8	96.7	–	–
	HTL [4]	cos	GoogLeNet	80.9	94.3	95.8	97.2	97.4	97.8
	D&C [23]	cos	ResNet-50	85.7	95.5	96.9	97.5	–	98.0
	ABE [11]	cos	GoogLeNet	87.3	96.7	97.9	98.2	98.5	98.7
	EPSHN [43]	cos	ResNet-50	87.8	95.7	96.8	–	–	–
	NormSoftmax [46]	cos	ResNet-50	89.4	97.8	98.7	99.0	–	–
	Multi-Similarity [37]	cos	BN-Inception	89.7	97.9	98.5	98.8	99.1	99.2
	Cross-Entropy	ℓ_2	ResNet-50	90.6	97.8	98.5	98.8	98.9	99.0
		cos		90.6	98.0	98.6	98.9	99.1	99.2

5.4   Results

Results for the experiments are reported in Table 4. We also report the architecture used in the experiments as well as the distance used in the evaluation to compute the recall. ℓ_2 refers to the Euclidean distance on non normalized features while *cos* refers to either the cosine distance or the Euclidean distance on ℓ_2-normalized features, both of which are equivalent.

On all datasets, we report state-of-the-art results except on Cars, where the only method achieving similar recall uses cross-entropy for training. We also notice that, contrary to common beliefs, using Euclidean distance can actually be competitive as it also achieves near state-of-the-art results on all four datasets. These results clearly highlight the potential of cross-entropy for metric learning, and confirm that this loss can achieve the same objective as pairwise losses.

6   Conclusion

Throughout this paper, we revealed non-obvious relations between the cross-entropy loss, widely adopted in classification tasks, and pairwise losses commonly used in DML. These relations were drawn under two different perspectives. First, cross-entropy minimization was shown equivalent to an approximate bound-optimization of a pairwise loss, introduced as Pairwise Cross-Entropy (PCE), which appears similar in structure to already existing DML losses. Second, adopting a more general information theoretic view of DML, we showed that both pairwise losses and cross-entropy were, in essence, maximizing a common mutual information $\mathcal{I}(\hat{Z}, Y)$ between the embedded features and the labels. This connection becomes particularly apparent when writing mutual information in both its *generative* and *discriminative* views. Hence, we argue that most of the differences in performance observed in previous works come from the optimization process during training. Cross-entropy only contains unary terms, while traditional DML losses are based on pairwise terms optimization, which requires substantially more tuning (*e.g.* mini-batch size, sampling strategy, pair weighting). And while we acknowledge that some losses have better properties than others regarding optimization, we empirically showed that the cross-entropy loss was also able to achieve state-of-the-art results when fairly tuned, highlighting the fact that most improvements have come from enhanced training schemes (*e.g.* data augmentation, learning rate policies, batch normalization freeze) rather than intrinsic properties of pairwise losses. We strongly advocate that cross-entropy should be carefully tuned to be compared against as a baseline in future works.

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning (ICML) (2020)
2. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the International Conference on Machine Learning (ICML) (2007)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
4. Ge, W.: Deep metric learning with hierarchical triplet loss. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
5. Goldberger, J., Hinton, G.E., Roweis, S.T., Salakhutdinov, R.R.: Neighbourhood components analysis. In: Advances in Neural Information Processing Systems (NeurIPS) (2005)
6. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2006)
7. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
8. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
9. Jia, X., Song, S., He, W., Wang, Y., Rong, H., Zhou, F., Xie, L., Guo, Z., Yang, Y., Yu, L., et al.: Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. arXiv preprint arXiv:1807.11205 (2018)
10. Kedem, D., Tyree, S., Sha, F., Lanckriet, G.R., Weinberger, K.Q.: Non-linear metric learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2012)
11. Kim, W., Goyal, B., Chawla, K., Lee, J., Kwon, K.: Attention-based ensemble for deep metric learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
12. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops (2013)
13. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
14. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
15. Lowe, D.G.: Similarity metric learning for a variable-kernel classifier. Neural Computation (1995)
16. Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
17. Musgrave, K., Belongie, S., Lim, S.N.: A metric learning reality check. arXiv preprint arXiv:2003.08505 (2020)
18. Narasimhan, M., Bilmes, J.: A submodular-supermodular procedure with applications to discriminative structure learning. In: Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI) (2005)

19. Oh Song, H., Jegelka, S., Rathod, V., Murphy, K.: Deep metric learning via facility location. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
20. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
21. Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Bier-boosting independent embeddings robustly. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS). Curran Associates, Inc. (2019)
23. Sanakoyeu, A., Tschernezki, V., Buchler, U., Ommer, B.: Divide and conquer the embedding space for metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
24. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
25. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: Advances in Neural Information Processing Systems (NeurIPS) (2004)
26. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Advances in Neural Information Processing Systems (NeurIPS) (2016)
27. Song, H.O., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
28. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
29. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
30. Tang, M., Marin, D., Ben Ayed, I., Boykov, Y.: Kernel cuts: Kernel and spectral clustering meet regularization. International Journal of Computer Vision (2019)
31. Tschannen, M., Djolonga, J., Rubenstein, P.K., Gelly, S., Lucic, M.: On mutual information maximization for representation learning. In: International Conference on Learning Representations (2020)
32. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
33. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Processing Letters (2018)
34. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
35. Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y.: Deep metric learning with angular loss. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)

36. Wang, M., Sha, F.: Information theoretical clustering via semidefinite programming. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AIStats) (2011)
37. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
38. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research (JMLR) (2009)
39. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
40. Wu, Z., Efros, A.A., Yu, S.X.: Improving generalization via scalable neighborhood component analysis. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
41. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: Advances in Neural Information Processing Systems (NeurIPS) (2003)
42. Xuan, H., Souvenir, R., Pless, R.: Deep randomized ensembles for metric learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
43. Xuan, H., Stylianou, A., Pless, R.: Improved embeddings with easy positive triplet mining. In: The IEEE Winter Conference on Applications of Computer Vision (WACV) (2020)
44. Yuan, Y., Yang, K., Zhang, C.: Hard-aware deeply cascaded embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
45. Yuille, A.L., Rangarajan, A.: The concave-convex procedure (cccp). In: Advances in neural information processing systems (NeurIPS) (2002)
46. Zhai, A., Wu, H.Y.: Classification is a strong baseline for deep metric learning. In: British Machine Vision Conference (BMVC) (2019)
47. Zheng, W., Chen, Z., Lu, J., Zhou, J.: Hardness-aware deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)