Quadruplet-wise Image Similarity Learning

Marc T. Law Nicolas Thome Matthieu Cord

LIP6, UPMC - Sorbonne University, Paris, France
{Marc.Law, Nicolas.Thome, Matthieu.Cord}@lip6.fr

Abstract

This paper introduces a novel similarity learning framework. Working with inequality constraints involving quadruplets of images, our approach aims at efficiently modeling similarity from rich or complex semantic label relationships. From these quadruplet-wise constraints, we propose a similarity learning framework relying on a convex optimization scheme. We then study how our metric learning scheme can exploit specific class relationships, such as class ranking (relative attributes), and class taxonomy. We show that classification using the learned metrics gets improved performance over state-of-the-art methods on several datasets. We also evaluate our approach in a new application to learn similarities between webpage screenshots in a fully unsupervised way.

1. Introduction

Similarity learning is useful in many Computer Vision applications, such as image classification [6, 10, 17], image retrieval [6], face verification or person re-identification [12, 18]. The key ingredients of similarity learning framework are (i) the data representation including both the feature space and the similarity function, (ii) the learning framework which includes: training data, type of labels and relations, the optimization formulation and solvers.

The usual way to learn similarities is to consider binary labels on image pairs [29]. For instance, in the context of face verification [12], binary labels establish whether two images should be considered equivalent or not. Metrics are learned with training data to minimize dissimilarities between similar pairs while separating dissimilar ones. Many different metrics have been considered in Euclidean space or using kernel embedding [18].

Recently, some attempts have been made to go beyond learning metrics with pairwise constraints generated from binary class membership labels. On the one hand, triplet-wise constraints have been considered to learn metrics [6, 15, 28]. Triplet constraints may be generated from class labels or they can be inferred from richer relationships. For example, Verma *et al.* [26] learn a similarity that depends on a class hierarchy: an image should be closer to another image from a sibling class than to any image from a distant class in the hierarchy. Other methods exploit specific rankings between classes. For instance, relative attributes have been introduced in [20]: different classes (*e.g.* ”celebrity”) are ranked with respect to different concepts or attributes (*e.g.* ”smile”), see Fig. 1 (top). Pairwise relations are extracted: *e.g.* face images from class (x) smile more than (or as much as) face images from class (y). In [20], it is shown that learning relative features can help significantly boost classification performances.

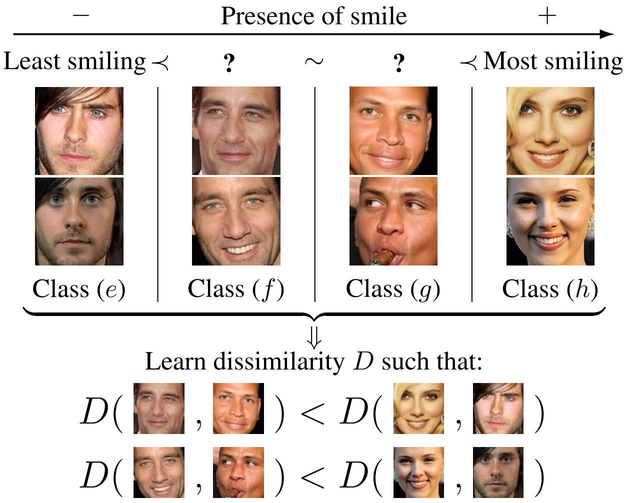

$-$ Presence of smile $+$
Least smiling $\prec$? $\sim$? $\prec$ Most smiling
Class (e)
Class (f)
Class (g)
Class (h)
$\Downarrow$
Learn dissimilarity D such that:
$D($, $) < D($, $)$
$D($, $) < D($, $)$

Figure 1. Quadruplet-wise (Qwise) strategy on 4 face classes ranked according to the degree of presence of smile. Instead of working on pairwise relations that present some flaws (see text), Qwise strategy defines quadruplet-wise constraints to express that dissimilarities between examples from (f) and (g) should be smaller than dissimilarities between examples from (e) and (h).

In this paper, we focus on these rich contexts for learning similarity metrics. Instead of pairwise or triplet-wise techniques, we propose to investigate relations between quadruplets of images. We claim that, in many contexts, considering relations such as *two images are more similar than two other images* may be useful to learn a similarity. Our motivation is illustrated in Fig. 1 in which enforcing strong pairwise equivalence constraints as in [20] may be problematic: (second row) Owen (f) is smiling more than Rodriguez (g) although their classes are annotated as smiling as much as each other. To overcome this limitation, one can consider relations on quadruplets: noting that the difference between the surrounding classes (e) and (h) is always greater than between (f) and (g), we express inequality constraints on dissimilarities (Fig. 1 (bottom part)).

Based on this quadruplet-wise (Qwise) approach, we propose in this paper a generic framework to learn metrics. To get efficient optimization, it is based on Mahalanobis-like metrics embedded in a convex optimization scheme. Section 2 positions the paper with respect to related works. Section 3 details our metric learning framework. We then demonstrate the advantage of our approach for image classification with respect to pairwise and triplet-wise strategies (Sections 4 and 5), and also for a new emerging context about webpage visual screenshot comparison (Section 6).

2. Related Work

Image representation for classification has been deeply investigated in recent years [7, 19]. The traditional Bag of Words representation [22] has been extended for the coding step [11, 30] as well as for the pooling [2], or with bio-inspired models [21, 24]. However, all these approaches focus on image representation. Similarity functions are also important to compare, classify and retrieve images. The Mahalanobis-like distance $D_{\mathbf{W}}$ is definitively the most investigated metric for metric learning:

$$D^2_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j), \mathbf{W} \succeq 0 \qquad (1)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a symmetric positive semi-definite (PSD) matrix ($\mathbf{W} \succeq 0$) and $(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}^d \times \mathbb{R}^d$ are representations of images p_i and p_j. As explained in [28], one can work on the elements of the matrix $\mathbf{W}$ or learn the linear transformation of the input space parameterized by a matrix $\mathbf{L}$ such that $\mathbf{W} = \mathbf{L}^\top \mathbf{L}$ and $D^2_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2$.

We focus in this work on supervised learning methods. The learning strategy is usually driven by the application. When supervision is considered, the way the dataset is labeled, *e.g.* binary labels on pairwise or triplet-wise rankings, greatly affects the optimization problem formulation.

2.1. Pairwise optimization framework

In pairwise approaches [18, 29], the problem is formulated as learning $\mathbf{W}$ such that the distance $D^2_{\mathbf{W}}$ is optimized on a training set composed of a subset $\mathcal{S}$ of similar images and a subset $\mathcal{D}$ of dissimilar images. For instance, [29] define the resulting convex objective function:

$$\min_{\mathbf{W}} \sum_{\mathcal{S}} D_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j) \quad s.t. \sum_{\mathcal{D}} D_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j) \geq 1 \ , \ W \succeq 0 \qquad (2)$$

A regularization term may be added. A hinge loss or a generalized logistic loss function may be used to express all the constraints (over $\mathcal{S}$ and $\mathcal{D}$) in a single functional [18]. This learning process may be extended to kernel functions [14, 18].

Many supervised approaches have been proposed recently to get training sets $\mathcal{S}$ and $\mathcal{D}$. Most of those approaches use binary similarity labels: two images represent the same object or not [12, 29], two images belong to the same class or not [18].

2.2. Triplets and extensions

Another way to exploit labeled datasets is to consider triplets of images (p_i, p_i^+, p_i^-) where the dissimilarity $D_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_i^+)$ between (p_i, p_i^+) is smaller than $D_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_i^-)$ between (p_i, p_i^-). This type of constraints is easy to generate in classification: (p_i, p_i^+) are sampled from the same class and (p_i, p_i^-) from different classes [10, 15, 25, 28]. For instance, Large Margin Nearest Neighbor algorithm (LMNN) [28] learns a Mahalanobis distance for k-Nearest Neighbors (k-NN) approach using these triplet-wise training sets. More precisely, LMNN uses a scheme similar to Eq. (2) with a hinge loss function to enforce $D^2_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_i^-)$ to be larger than $D^2_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_i^+)$.

In image retrieval, the Online Algorithm for Scalable Image Similarity (OASIS) [6] uses a non-PSD square matrix $\mathbf{W}$ in the bilinear function $\mathbf{x}_i^\top \mathbf{W} \mathbf{x}_j$ as a similarity between images p_i and p_j. For any triplet, a safety margin constraint is defined: $\mathbf{x}_i^\top \mathbf{W}(\mathbf{x}_i^+ - \mathbf{x}_i^-) \geq 1$. As explained by the authors [6], OASIS requires images represented as sparse vectors to be computationally efficient.

Other approaches investigate different dataset labels or semantic relationships to build pairwise or triplet-wise metric learning schemes. For instance, in [27], a class taxonomy is used in order to get elements of related classes, close to each other. Verma *et al.* [26] extend this work by learning for each class a local Mahalanobis distance. Hwang *et al.* [13] learn discriminative visual representations while exploiting external semantic knowledge about object category relationships. In [20], complex relations between classes are used. They consider totally ordered sets of classes that describe relations among classes. Based on these rich relations, they learn image representations by exploiting only pairwise class relations.

We propose to explore this type of data knowledge in metric learning for image comparison. Noting that pairwise or triplet-wise approaches may, sometimes, be limited (see Section 1), our learning framework is based on constraints on quadruplets.

3. Qwise Similarity Learning Framework

As illustrated in Fig. 1, pair or triplet constraints may be noisy or irrelevant, leading to less than optimal learning scheme when provided at a class level. On the other hand, working on dissimilarities between quadruplets of images limits the risk of incorporating misleading annotations. One can find other Computer Vision applications where pairwise dissimilarities D_{ij} might be hard or meaningless for humans to annotate and quadruplet-wise easy or meaningful. For instance, perceptual color spaces have been proposed using experiments to compare D_{ij} and D_{kl}.

We are interested in comparing pairs of dissimilarities (D_{ij}, D_{kl}) that involve up to four different images (p_i, p_j, p_k, p_l). We are given a set $\mathcal{P}$ of images p_i, and the target dissimilarity function $D : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ between pairs of images (p_i, p_j), we note $D(p_i, p_j) = D_{ij}$. Two types of relations $\mathcal{R}$ are considered between D_{ij} and D_{kl}: (1) strict inequality between dissimilarities: $D_{ij} < D_{kl}$, (2) non-strict inequality: $D_{ij} \leq D_{kl}$. Note that $D_{ij} = D_{kl}$ can be rewritten as two relations $D_{ij} \leq D_{kl}$ and $D_{ij} \geq D_{kl}$. We then define two training sets: $\mathcal{A}$, composed of quadruplets (p_i, p_j, p_k, p_l) satisfying $D_{ij} < D_{kl}$ (D_{kl} strict upper bound of D_{ij}) and $\mathcal{B}$, composed of quadruplets (p'_i, p'_j, p'_k, p'_l) satisfying $D_{i'j'} \leq D_{k'l'}$ (non-strict upper bound).[1]

3.1. Metric formulation

Following Eq. (1), the dissimilarity considered for learning in this paper is:

$$D^2_{\mathbf{W}}(p_i, p_j) = \Phi(p_i, p_j)^\top \mathbf{W} \, \Phi(p_i, p_j) \tag{3}$$

with $\mathbf{W} \in \mathbb{R}^{d \times d}$ PSD and $\Phi(p_i, p_j) \in \mathbb{R}^d$ the aggregation in a single vector of d elementary dissimilarity functions ϕ_k where $\forall k \in \{1, \dots, d\}$, $\phi_k : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$. The basic choice for Φ is $\Phi(p_i, p_j) = \mathbf{x}_i - \mathbf{x}_j$ corresponding to Eq. (1).

Working in $\mathbb{R}^d$, the optimization may be done using a global loss on Eq. (3) over a training set, as proposed in [6, 28]. However, it is usually computationally expensive to guarantee $\mathbf{W}$ PSD.[2] An option is to optimize over $\mathbf{L} \in \mathbb{R}^{n \times d}$ *s.t.* $\mathbf{L}^\top \mathbf{L} = \mathbf{W}$ (usually $n < d$). This reduces the number of parameters from d^2 to nd, and thus may overcome overfitting issues, as discussed in [18]. However, this can lead to a non-convex optimization problem. As a consequence, [28] alternate their optimization steps w.r.t. $\mathbf{W}$ and $\mathbf{L}$.

Instead, we focus on two contexts where the optimization may be done efficiently and with a relatively small number of parameters to avoid overfitting:

- $\mathbf{W}$ is diagonal: $Diag(\mathbf{W}) = \mathbf{w}$ and $D^2_{\mathbf{W}}(p_i, p_j) = \mathbf{w}^\top \Phi^2(p_i, p_j)$ with $\Phi^2(p_i, p_j) = \Phi(p_i, p_j) \circ \Phi(p_i, p_j)$ where $\circ$ is the Hadamard product (element-by-element product). $\mathbf{W}$ is PSD (*i.e.* $D^2_{\mathbf{W}}$ is a squared Mahalanobis distance) iff $\mathbf{w} \geq \mathbf{0}$ (the elements of $\mathbf{w}$ are non-negative). We then restrict $\mathbf{w} \geq \mathbf{0}$ in this case.
- Optimization over the rows of $\mathbf{L}$: if the annotations provide M different dissimilarity functions, where each of them represents a relative ordering focused on a given criterion (*e.g.* p_i is more smiling than p_j, p_i is younger than p_j...), each row of the matrix $\mathbf{L} \in \mathbb{R}^{M \times d}$ can be learned independently. The m^{th} row of $\mathbf{L}$ (denoted $\mathbf{w}_m^\top$) satisfies the ordering of the m^{th} function $D_{\mathbf{w}_m}(p_i, p_j) = \mathbf{w}_m^\top \Phi(p_i, p_j)$.

In both cases, the learning problem may be expressed as a linear combination of the $\mathbf{w}$ parameters. Without loss of generality, we then consider optimizing the following dissimilarity function (with Ψ equal to Φ or Φ^2):

$$D_{\mathbf{w}}(p_i, p_j) = \mathbf{w}^\top \, \Psi(p_i, p_j) \tag{4}$$

3.2. Learning scheme

Let $D_{\mathbf{w}}$ be the family of dissimilarities that we consider. Our goal is to learn the parameters of $D_{\mathbf{w}}$ such that the maximum number of the following constraints is satisfied:

$$\forall (p_i, p_j, p_k, p_l) \in \mathcal{A} : D_{\mathbf{w}}(p_k, p_l) \geq D_{\mathbf{w}}(p_i, p_j) + 1 \tag{5}$$

$$\forall (p_i, p_j, p_k, p_l) \in \mathcal{B} : D_{\mathbf{w}}(p_k, p_l) \geq D_{\mathbf{w}}(p_i, p_j) \tag{6}$$

This means that $D_{\mathbf{w}}$ fulfills in Eq. (5) the constraint $D_{kl} > D_{ij}$ with a safety margin of 1, and $D_{\mathbf{w}}$ fulfills in Eq. (6) the constraint $D_{kl} \geq D_{ij}$. Eq. (5) is similar to the constraints used in triplet-wise approaches [10, 25, 28] with the exception that we use quadruplets of images.

The problem of determining a $\mathbf{w}$ that maximizes the number of satisfied constraints in Eq. (5) and Eq. (6) (corresponding to minimizing a global loss with a $0/1$ loss function per constraint) is NP-hard [20]. Instead, we use a surrogate loss function that is convex and differentiable. We first rewrite these equations using Eq. (4). Let $\mathbf{z}_q$ be the vector of differences of quadruplet $q = (p_i, p_j, p_k, p_l)$: $\mathbf{z}_q = \mathbf{z}_{ijkl} = \Psi(p_k, p_l) - \Psi(p_i, p_j)$. We have $D_{\mathbf{w}}(p_k, p_l) - D_{\mathbf{w}}(p_i, p_j) = \mathbf{w}^\top \mathbf{z}_q$, and Eq. (5) and Eq. (6) may be rewritten:

$$\forall q \in \mathcal{A} : \mathbf{w}^\top \mathbf{z}_q \geq 1 \tag{7}$$

$$\forall q \in \mathcal{B} : \mathbf{w}^\top \mathbf{z}_q \geq 0 \tag{8}$$

As explained in Section 2, we can use a loss function over the training set $\mathcal{A} \cup \mathcal{B}$ to define our objective function. Since the constraints over $\mathcal{A}$ and $\mathcal{B}$ are different, we first define the loss function L^h_1 over quadruplets in $\mathcal{A}$ w.r.t.

[1] Although quadruplet-wise constraints can be inferred from pairwise approaches [8, 18], *i.e.* $(p_i, p_j) \in \mathcal{S}, (p_k, p_l) \in \mathcal{D} \Rightarrow D_{ij} < D_{kl}$, the reverse is not true. Quadruplet-wise constraints can be generated even if (p_i, p_j) and (p_k, p_l) are both similar or dissimilar. Only the order of similarity between (p_i, p_j) and (p_k, p_l) is required.

[2] Gradient descent over $\mathbf{W}$ usually requires projecting the solution onto the cone of PSD matrices at each iteration using spectral decomposition.

Eq. (7). We note $t = \mathbf{w}^\top \mathbf{z}_q$ ($\forall q \in \mathcal{A}$) and use the following differentiable loss function[3]:

$$L_1^h(t) = \begin{cases} 0 & \text{if} \quad t > 1+h \\ \frac{(1+h-t)^2}{4h} & \text{if} \quad |1-t| \leq h \\ 1-t & \text{if} \quad t < 1-h \end{cases} \tag{9}$$

We define L_0^h as an adaptation of L_1^h that considers the absence of safety margin in Eq. (8). $\forall q \in \mathcal{B}$, $t = \mathbf{w}^\top \mathbf{z}_q$:

$$L_0^h(t) = \begin{cases} 0 & \text{if} \quad t > 0 \\ \frac{t^2}{4h} & \text{if} \quad |-h-t| \leq h \\ -h-t & \text{if} \quad t < -2h \end{cases} \tag{10}$$

To avoid overfitting, we introduce a regularization over $\mathbf{w}$ (term $\|\mathbf{w}\|_2^2$). We then get our optimization problem:

$$\min_{\mathbf{w}} \sum_{q \in \mathcal{A}} L_1^h(\mathbf{w}^\top \mathbf{z}_q) + \sum_{q \in \mathcal{B}} L_0^h(\mathbf{w}^\top \mathbf{z}_q) + \lambda \|\mathbf{w}\|_2^2 \tag{11}$$

where the parameter λ weighs the regularization.

By choosing such a regularization, our scheme may be compared to a ranking SVM [5], except that the loss functions work on quadruplets. Therefore, the optimization problem defined in Eq. (11) is convex. We solve the above primal problem using Newton's method [5]. The complexity of our optimization is linear in the size of $\mathcal{A} \cup \mathcal{B}$. It can be solved efficiently even with a large number of constraints. In the three applications that we consider, the learning can be achieved on a single computer in less than one hour. The number of parameters to learn is small and grows linearly with the input space dimension, limiting overfitting [18]. It can also be extended to kernels [5].

The two key ingredients of our approach are the formulation of Ψ and the generation of the training set $\mathcal{A} \cup \mathcal{B}$. We present in the next sections different ways to choose Ψ, $\mathcal{A}$ and $\mathcal{B}$ depending on the application context.

4. Metric learning and Relative Attributes

Relative attributes have been introduced in [20]. Attributes are human-nameable concepts used to describe images. In Fig. 1 the attribute a_m = "Presence of smile" allows to rank 4 celebrity classes from the least to the most smiling. Instead of considering attributes as boolean values (the concept is present in the image or not), Parikh and Grauman [20] learn for each attribute a_m a vector $\mathbf{w}_m \in \mathbb{R}^d$ so that the score $\mathbf{w}_m^\top \mathbf{x}_i$ represents the degree of presence of a_m in p_i ($\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of p_i).

To learn $\mathbf{w}_m$, they use original training sets about relative ordering between classes such as the one presented in Fig. 1: $(e) \prec (f) \sim (g) \prec (h)$. [20] only consider pairwise relations for learning: $(e) \prec (f)$ meaning that images of class (f) have stronger presence of attribute a_m than images of class (e), and $(f) \sim (g)$ meaning that images of (f) and (g) have similar relative strengths of attribute a_m.

By considering the signed dissimilarity $D_{\mathbf{w}_m}(p_i, p_j) = \mathbf{w}_m^\top(\mathbf{x}_i - \mathbf{x}_j)$, their constraints are of the form $D_{\mathbf{w}_m}(p_i, p_j) \geq 1$ and $D_{\mathbf{w}_m}(p_i, p_j) = 0$.

4.1. Ψ, $\mathcal{A}$: Qwise strategy

Following our formalism defined in Section 3, we consider $D_{\mathbf{w}_m}(p_i, p_j) = \mathbf{w}_m^\top \Psi(p_i, p_j)$, and $\Psi(p_i, p_j) = \mathbf{x}_i - \mathbf{x}_j$.

As explained in Section 1, the learning information is provided at a class level: pairwise constraints may be noisy or irrelevant, leading to less than optimal learning scheme. Considering triplet-wise constraints (class (x) is more similar to (y) than to (z)) could be helpful but still generates inconsistent constraints in some cases: in Fig. 1 (second row), Owen (f) seems to be more similar to Johansson (h) than to Rodriguez (g). To further exploit the available ordered set of classes and overcome these limitations, we consider relations on quadruplets. Two types of Qwise constraints may be derived from the training set. The first one is: $(e) \prec (f) \prec (g) \prec (h)$. We then do the following assumption: any image pair from the extreme border classes (e) and (h) are more dissimilar than any image pair from the intermediate classes (f) and (g):

$$\forall (p_i, p_j, p_k, p_l) \in (g) \times (f) \times (h) \times (e) \;\; D_{kl} > D_{ij} \tag{12}$$

By working with pairs of dissimilarities, the risk of incorporating misleading annotations into our process is limited. By sampling such quadruplets from the whole set of relative orderings on classes (*e.g.* Table 1, see experiments for details), we build our Qwise set $\mathcal{A}$.

The second type of relations is: $(e) \prec (f) \sim (g) \prec (h)$, meaning that the pair (p_i, p_j) are similar. We then use a slightly different assumption: $D_{kl} > |D_{ij}|$ to take into account the fact that p_i and p_j are not ranked. In order to have a convex problem, we rewrite it as two constraints:[4]

$$\begin{cases} D_{kl} \geq D_{ij} + 1 \\ D_{kl} \geq D_{ji} + 1 \end{cases} \tag{13}$$

We thus generate two quadruplets in $\mathcal{A}$ from Eq. (13). Note that $\mathcal{B}$ remains empty in this application. Once the optimal weight vectors $\mathbf{w}_m$ are learned for all a_m, each image p_i is described by a high level feature representation: $\mathbf{h}_i = [\mathbf{w}_1^\top \mathbf{x}_i, \ldots, \mathbf{w}_m^\top \mathbf{x}_i, \ldots, \mathbf{w}_M^\top \mathbf{x}_i]^\top \in \mathbb{R}^M$ where M is the number of attributes. This corresponds to learning a linear transformation parameterized by $\mathbf{L} \in \mathbb{R}^{M \times d}$ such that $\mathbf{h}_i = \mathbf{L}\mathbf{x}_i$ where the m-th row of $\mathbf{L}$ is $\mathbf{w}_m^\top$.

[3] As described in [4], L_1^h is a differentiable approximation of the hinge loss when $h \to 0$. It is inspired by the Huber Loss function. Usually $h \in [0.01, 0.5]$. In all our experiments, h is set to 0.05.

[4] It is not necessary to discuss the sign of D_{kl} since p_k was annotated to have stronger presence of a_m than p_l. We infer $D_{kl} > 0$.

OSR Attributes	Relative Ordering of Classes
Natural	$\mathrm{T} \prec \mathrm{I} \sim \mathrm{S} \prec \mathrm{H} \prec \mathrm{C} \sim \mathrm{O} \sim \mathrm{M} \sim \mathrm{F}$
Open	$\mathrm{T} \prec \mathrm{F} \prec \mathrm{I} \sim \mathrm{S} \prec \mathrm{M} \prec \mathrm{H} \sim \mathrm{C} \sim \mathrm{O}$
Perspective	$\mathrm{O} \prec \mathrm{C} \prec \mathrm{M} \sim \mathrm{F} \prec \mathrm{H} \prec \mathrm{I} \prec \mathrm{S} \prec \mathrm{T}$
Large-Objects	$\mathrm{F} \prec \mathrm{O} \prec \mathrm{M} \prec \mathrm{I} \sim \mathrm{S} \prec \mathrm{H} \sim \mathrm{C} \prec \mathrm{T}$
Diagonal-Plane	$\mathrm{F} \prec \mathrm{O} \prec \mathrm{M} \prec \mathrm{C} \prec \mathrm{I} \sim \mathrm{S} \prec \mathrm{H} \prec \mathrm{T}$
Close-Depth	$\mathrm{C} \prec \mathrm{M} \prec \mathrm{O} \prec \mathrm{T} \sim \mathrm{I} \sim \mathrm{S} \sim \mathrm{H} \sim \mathrm{F}$

Table 1. Relative orderings used in [20] for OSR (categories: coast (C), forest (F), highway (H), inside-city (I), mountain (M), open-country (O), street (S) and tall-building (T)).

4.2. Classification Experiments

To evaluate and compare our Qwise scheme, we follow a classification framework inspired from [20] for scene and face recognition on the OSR [19] and Pubfig [16] datasets.

Datasets: We experiment with the two datasets used in [20]: Outdoor Scene Recognition (OSR) [19] containing 2688 images from 8 scene categories and a subset of Public Figure Face (PubFig) [16] containing 771 images from 8 face categories. We use the image features made publicly available by [20]: a 512-dimensional GIST [19] descriptor for OSR and a concatenation of the GIST descriptor and a 45-dimensional Lab color histogram for PubFig. Another information is also available for both datasets: relative orderings of classes according to some semantic attributes (see Table 1 for OSR).

Baselines: We use three baselines: (1) the linear transformation learned with LMNN [28] that uses only class membership information[5], (2) the relative attribute learning problem of Parikh and Grauman [20] that uses relative attribute annotations on classes (*e.g.* Table 1), unlike LMNN, to generate and exploit only pairwise constraints, (3) a combination of the first two baselines that first uses relative attribute annotations to learn a representation of images in attribute space, and second, learns a metric in attribute space with LMNN. We call this baseline **RA + LMNN**. We use the publicly available codes of [20] and [28].

Qwise Method: We use for **OSR** and **Pubfig** the Qwise constraints defined in Section 4.1. The Qwise scheme only uses relative attribute information to learn a linear transformation. This linear transformation can be exploited by other linear transformation learning methods that use class membership information. We chose LMNN [28]: the high level features $\mathbf{h}_i$ learned with our method are used as input of LMNN. We call this strategy **Qwise + LMNN**.

Learning setup: We use the same experimental setup as [20] to learn our Qwise metric. $N = 30$ training images are used per class, the rest is for testing. To learn the projection direction $\mathbf{w}_m$ of attribute a_m, we select pairs of classes. From each selected pair of classes, we extract $N \times N$ image pairs or quadruplets to create training constraints. To carry out fair comparisons, we generate one Qwise constraint for each pairwise constraint generated by [20] using the strategies described in Section 4.1. We then have the same number of constraints. Once all the M projection directions $\mathbf{w}_m$ are learned, a Gaussian distribution is learned for each class c_s of images: the mean $\boldsymbol{\mu}_s \in \mathbb{R}^M$ and covariance matrix $\mathbf{\Sigma}_s \in \mathbb{R}^{M \times M}$ are estimated using the $\mathbf{h}_i$ of all training images $p_i \in c_s$. A test image p_t is then assigned to the class corresponding to the highest likelihood. The performance is measured as the average classification accuracy across all classes over 10 random train/test splits.

	OSR	Pubfig
Parikh's code [20]	$71.3 \pm 1.9\%$	$71.3 \pm 2.0\%$
LMNN-G	$70.7 \pm 1.9\%$	$69.9 \pm 2.0\%$
LMNN	$71.2 \pm 2.0\%$	$71.5 \pm 1.6\%$
RA + LMNN	$71.8 \pm 1.7\%$	$74.2 \pm 1.9\%$
Qwise	$74.1 \pm 2.1\%$	$74.5 \pm 1.3\%$
Qwise + LMNN-G	$\mathbf{74.6} \pm \mathbf{1.7}\%$	$76.5 \pm 1.2\%$
Qwise + LMNN	$74.3 \pm 1.9\%$	$\mathbf{77.6} \pm \mathbf{2.0}\%$

Table 2. Test classification accuracies on the OSR and Pubfig datasets for different methods.

Results: Table 2 reports the classification scores for the three baselines, Qwise, and Qwise+LMNN. A k-NN classifier is used for the LMNN methods (since LMNN is designed for k-NN classification) whereas Gaussian models are used for the LMNN-G methods to have the same classifier as [20]. On OSR and Pubfig, our method reaches an accuracy of 74.1% and 74.5%, respectively. It outperforms the first two baselines on both datasets with a margin of 3% accuracy, reaching state-of-the-art results in this original setup [20]. Moreover, performance is further improved when combining Qwise and LMNN. Particularly, an improvement of about 3% is obtained on Pubfig, reaching 77.6%. Relative attribute annotations (used for Qwise learning) and class membership information (used for LMNN) then seem complementary.

We also investigated how our quadruplets are sampled from the set of ordering relations. For instance, if we have $(k) \prec (i) \prec (e) \prec (f) \sim (g) \prec (h) \prec (j) \prec (l)$ and focus on the class pair $(f) \sim (g)$, in all our experiments, we only sampled quadruplets from the 4 classes $(e) \prec (f) \sim (g) \prec (h)$ (step 1). We experimented with increasing the step: *e.g.* $(i) \prec (f) \sim (g) \prec (j)$ (step 2) or $(k) \prec (f) \sim (g) \prec (l)$ (step 3). There are no significant differences in the results, our method is very robust and always better than baselines.

[5] For each image, LMNN tries to satisfy the condition that members of a predefined set of target neighbors (of the same class) are closer than samples from other classes. In [28], those neighbors are chosen using the ℓ_2-distance in the input space.

5. Hierarchical Metric Learning

Another classification context with rich annotations is metric learning using a semantic taxonomy structure. We study in this section how our model can exploit complex relations from a class hierarchy as proposed in [26] . Our objective is to learn a metric such that images from close (sibling) classes with respect to the class semantic hierarchy are more similar than images from more distant classes.

5.1. Ψ, $\mathcal{A}$: Qwise formulation

Given a semantic taxonomy expressed by a tree of classes, let us consider two sibling classes c_a and c_b and one of their cousin classes c_d. We generate two types of quadruplet-wise constraints in order to:

- Enforce the dissimilarity between two images from the same class to be smaller than between two others from sibling classes. If (p_i, p_j) are both sampled from c_a, and (p_k, p_l) from $c_a \times c_b$, this means we want $D_{ij} < D_{kl}$.
- Enforce the dissimilarity between two images from sibling classes to be smaller than between two images from cousin classes. If (p_i, p_j) are sampled from $c_a \times c_b$ and (p_k, p_l) from $c_a \times c_d$, we want $D_{ij} < D_{kl}$.

All these quadruplets form the set $\mathcal{A}$. We use the diagonal PSD matrix learning framework described in Section 3. We formulate our distance $D_{\mathbf{w}}(p_i, p_j) = \mathbf{w}^\top \Psi(p_i, p_j)$ where $\Psi(p_i, p_j) = (\mathbf{x}_i - \mathbf{x}_j) \circ (\mathbf{x}_i - \mathbf{x}_j)$ and $\mathbf{w} = Diag(\mathbf{W})$. Once the diagonal PSD matrix $\mathbf{W} \geq \mathbf{0}$ is learned, we project the input space using the linear transformation parameterized by the diagonal matrix $\mathbf{W}^{1/2} = \mathbf{L} \in \mathbb{R}^{d \times d}$ such that $\forall i \in \{1, \dots, d\}, \mathbf{L}_{ii} = \sqrt{\mathbf{W}_{ii}}$ (note that $\mathbf{L}^\top \mathbf{L} = \mathbf{W}$).

5.2. Experiments

To validate the Qwise ability to learn a powerful metric using a class hierarchy, we focus on the local subtree classification task described in [26]. We use the same 9 datasets as in [26] (which are all subsets of ImageNet [9]). The goal is to discriminate classes (leafs of a hierarchical subtree) amongst a hierarchical subtree that contains all the considered classes. The training sets (mentioned in Table 3) contain from 8 to 40 different classes and from 8000 to 54000 images per subtree. We use the train, validation and test sets defined in [26], and the same publicly available features[6] as in [26]: 1000 dimensional SIFT-based Bag-of-Bords (BoW).

We compare our model to Verma *et al.* [26] which also use class taxonomy information to learn hierarchical similarity metrics. It is worth mentioning that they learn a local metric for each class (leaf of the subtree), parameterized by a full PSD matrix. Our Qwise-learning model is simpler since we learn a global metric for each subtree and use a diagonal matrix, for which the number of parameters only grows linearly with the input space dimension. In [26], they use an ad hoc classifier specifically designed for their local metric. Instead, since we learn a global metric, we use a standard classifier (linear SVM) to perform classification.

Subtree Dataset	Verma *et al.* [26]	Qwise
Amphibian	41%	**43.5**%
Fish	39%	**41**%
Fruit	**23.5**%	21.1%
Furniture	46%	**48.8**%
Geological Formation	52.5%	**56.1**%
Musical Instrument	32.5%	**32.9**%
Reptile	22%	**23.0**%
Tool	**29.5**%	26.4%
Vehicle	27%	**34.7**%
Global Accuracy	34.8%	**36.4**%

Table 3. Standard classification accuracy for the various datasets.

Test classification accuracies are reported in Table 3. LMNN and a polynomial SVM are reported in [26] to perform a global accuracy of 24.4% and 33.2%, respectively. Our method reaches a global accuracy of 36.4%, which is 1.6% better than [26]. It outperforms all the reported methods, globally and on each dataset except Fruit and Tool. Even if our metric learning strategy is not significantly better than a SVM scheme alone, the results are encouraging. The sampling strategy to get useful quadruplet constraints from these hierarchies has to be further investigated.

6. Temporal Metric Learning for Webpages

Inspired from the study of dynamics in webpages [1], a novel application of our formalism is proposed. For Web crawling purpose, it is useful to understand the change behavior of websites over time [3]. Significant changes between successive versions of a same webpage mean that a robot has to revisit the page and index it.

In this study, we focus on news websites, where advertisements or menus are not significant whereas the news content is significant. In this context, having a metric able to properly identify significant changes between webpage versions is crucial. An important aspect is the localization of these changes inside pages [1, 23]: each site has a semantical spatial structure important to capture. Using many manual annotations, [23] learn region weights inside pages. [3] exploit this strategy to detect significant changes using the source code of pages.

We propose to model this problem in a fully unsupervised way, exploiting temporal information to define a visual quadruplet-wise similarity learning scheme. We intend to learn (1) a semantical dissimilarity between versions, and (2) region weights that help interpreting the results.

[6] http://www.image-net.org/challenges/LSVRC/2010/

Websites	CNN			NPR			New York Times			BBC		
Measures	AP$_S$	AP$_D$	MAP	AP$_S$	AP$_D$	MAP	AP$_S$	AP$_D$	MAP	AP$_S$	AP$_D$	MAP
Euclidian dist.	68.1	85.9	77.0	96.3	89.5	92.9	69.8	79.5	74.6	91.1	76.7	83.9
LMNN dist.	78.8	91.7	85.2	98.0	92.5	95.2	83.2	89.1	86.1	92.5	**80.1**	**86.3**
Qwise dist.	**82.7**	**94.6**	**88.6**	**98.6**	**94.3**	**96.5**	**85.5**	**92.3**	**88.9**	**92.8**	79.3	86.1

Table 4. Webpage metric learning results: Similar (AP_S), Dissimilar (AP_D) and MAP (in %) on CNN, NPR, NYTimes and BBC.

6.1. Ψ, $\mathcal{A}$, $\mathcal{B}$: Qwise formulation

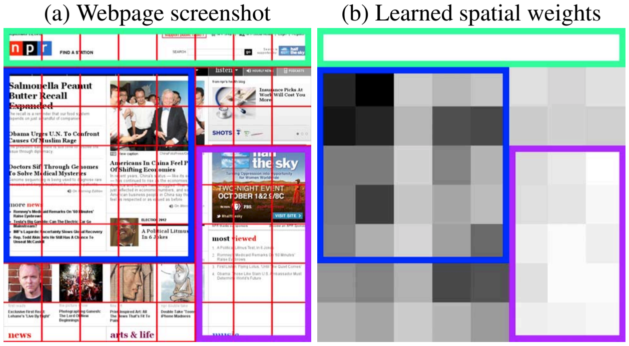
Let p_i be here a screen capture of a specific webpage at time i. Following Section 3, we use a diagonal matrix metric model and express our metric as: $D_{\mathbf{w}}(p_i, p_j) = \mathbf{w}^\top \Psi(p_i, p_j)$ with $\Psi(p_i, p_j) = \Phi^2(p_i, p_j)$. In practice, we use a non-overlapping grid of regions, and the d^{th} term in $\Phi^2(p_i, p_j)$ is the squared ℓ_2 distance between GIST descriptors in the d^{th} regions of p_i and p_j (see section 6.2 for details). The vector $\mathbf{w} \geq \mathbf{0}$ directly weights spatial regions in a webpage, allowing to learn a semantical spatial structure.

To generate our constraints, we assume that the dissimilarity between two successive screen captures p_t and p_{t+1} is smaller than the dissimilarity between a previous (p_r) and a later (distant enough) ($p_{r+\gamma}$) versions: $D(p_r, p_{r+\gamma}) > D(p_t, p_{t+1})$ if $r \leq t \leq r + \gamma - 1$. The parameter γ defines the period beyond which a strict inequality holds. In this case, the quadruplet $(p_t, p_{t+1}, p_r, p_{r+\gamma})$ lies in $\mathcal{A}$. Moreover, we also consider dissimilarities between closer versions using a $\gamma' < \gamma$, but we relax the inequalities to keep consistent constraints: $D(p_r, p_{r+\gamma'}) \geq D(p_t, p_{t+1})$. This kind of quadruplet $(p_t, p_{t+1}, p_r, p_{r+\gamma'})$ belongs to $\mathcal{B}$.

To help understanding what the constraints in $\mathcal{A}$ and $\mathcal{B}$ encode, we give two examples of region types that our Qwise scheme is able to learn. First, quadruplets in $\mathcal{B}$ that violate the constraint (*i.e.* the content between p_r and p_s is more similar than their intermediate versions (p_t, p_{t+1})) help to ignore regions where random and periodic changes occur. Typically, this happens for advertisements. Second, quadruplets in $\mathcal{A}$ that violate their corresponding constraint penalize content that does not change much in some region, although a change in the whole page is expected. These static regions correspond for example to menus: the algorithm learns to ignore these areas.

6.2. Experimental Results

Dataset: To evaluate the Qwise learning scheme, we provide a new webpage dataset. For this, we crawled CNN, BBC, NPR and New York Times homepages[7] for about 50 days. The crawling is performed each hour, as done in [1, 3]: p_{t+1} is visited one hour after p_t. For an evaluation purpose, we manually annotate successive versions (p_t, p_{t+1}) as dissimilar if a (semantical) significant change occurs between p_t and p_{t+1}, and as similar otherwise.[8]



(a) Webpage screenshot

(b) Learned spatial weights

Figure 2. Important change map for NPR. (a) Webpage screenshot, with relevant area (news) in blue, irrelevant parts (menu and advertisement) in green and purple, respectively. (b) Spatial weights learned by Qwise (larger values are darker).

Evaluation Process: To evaluate the quality of the metric, we use the Average Precision (AP). Since AP is not symmetric, we compute AP$_S$ and AP$_D$ for the similar and dissimilar classes, and report the Mean Average Precision (MAP). For each website, we split the dataset in 10 train/test subsets. For each split, a distance is learned on versions crawled during 5 successive days, and the successive versions of the 45 remaining days are used for testing.

Visual descriptors: We consider screen captures of page versions as images. Only the visible part of pages is considered since it generally contains the most useful information [23]. Our images thus have a maximal height of 1000 pixels, and a width usually of about 1000 pixels. We then use GIST descriptors [19] with an $m \times m$ grid over images. Thus, $\Psi(p_i, p_j) \in \mathbb{R}^{m^2}$ is a vector where each element corresponds to the squared ℓ_2-distance between between bins that fall into the same cell of the grids of p_i and p_j. We set $m = 10$ for results in Table 4.

Parameters to generate Qwise constraints: To generate constraints, we sample version quadruplets (p_t, p_{t+1}, p_r, p_s) in a temporal window (of 5 days) where t varies and so that $r \geq t - 6$, $s \leq t + 7$, $\gamma = 4$.

Baselines: We consider two baselines: (1) the Euclidean distance ($\mathbf{W} = I_d$ in Eq. (3)), (2) a learned metric using

[7] http://www.cnn.com, http://www.bbc.co.uk, http://www.npr.org, http://www.nytimes.com

[8] Some pages have been annotated as ambiguous if a clear annotation could not be decided. They are ignored from test evaluations.

LMNN, for which the set $\mathcal{A}$ is used to generate triplets.

Results and Discussion: Table 4 gives quantitative results on the ranking of distances of test similar/dissimilar page version pairs (p_t, p_{t+1}). Qwise favorably compares to the Euclidean distance and LMNN metrics. It is worth mentioning that the results have been obtained with the same γ parameter for all webpages. The performance gain is particularly noticeable compared to the Euclidean distance, especially in New York Times $(+14.3\%)$ and CNN $(+11.6\%)$. This illustrates the Qwise ability to focus on relevant areas (news) and to ignore "noisy" regions (advertisements or menus). In BBC, the gain is smaller $(+2.2\%)$ because some irrelevant changes take place from time to time in the only region where relevant changes usually occur. In addition, Qwise has an edge over LMNN, with a stable improvement, except in BBC where the results are similar. Fig 2 (b) illustrates region weights learned by our Qwise approach. The map plots the relative values of the learned $\mathbf{w} \geq \mathbf{0}$. The highest values, represented by dark regions, match with the significant content of the page (blue region). Menus and advertisements are ignored by the map as expected.

7. Conclusion and Perspectives

In this paper, we introduce our Qwise framework to learn similarities from quadruplets of images. It is specifically adapted to incorporate knowledge from rich or complex semantic label relations. The proposed metric parameterization makes the approach robust to overfitting, and the convexity of the objective function makes the learning effective. Our Qwise approach has been successfully evaluated in three different scenarios: relative attribute learning, metric learning on class hierarchy, and study of webpage changes.

Acknowledgments. This work was partially supported by the SCAPE Project cofunded by the European Union under FP7 ICT2009.4.1 (Grant Agreement nb 270137).

References

[1] E. Adar, J. Teevan, and S. Dumais. Resonance on the web: web dynamics and revisitation patterns. In *CHI*, 2009.

[2] S. Avila, N. Thome, M. Cord, E. Valle, and A. d. A. Araújo. Pooling in image representation: The visual codeword point of view. *CVIU*, 117(5):453–465, 2013.

[3] M. Ben Saad and S. Gançarski. Archiving the Web using Page Changes Pattern: A Case Study. In *JCDL*, 2011.

[4] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.

[5] O. Chapelle and S. S. Keerthi. Efficient algorithms for ranking with svms. *Inf. Retrieval*, 13(3):201–215, 2010.

[6] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010.

[7] M. Cord and P. Cunningham. *Machine learning techniques for multimedia*. Springer, 2008.

[8] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[10] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.

[11] H. Goh, N. Thome, M. Cord, and J. Lim. Unsupervised and supervised visual codes with restricted boltzmann machines. In *ECCV*, 2012.

[12] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, 2009.

[13] S. J. Hwang, K. Grauman, and F. Sha. Learning a tree of metrics with disjoint visual features. In *NIPS*, 2011.

[14] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, 2008.

[15] M. Kumar, P. Torr, and A. Zisserman. An invariant large margin nearest neighbour classifier. In *ICCV*, 2007.

[16] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[17] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Metric learning for large-scale image classification: generalizing to new classes at near-zero cost. In *ECCV*, 2012.

[18] A. Mignon and F. Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012.

[19] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[20] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[21] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *PAMI*, 29(3):411–426, 2007.

[22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[23] R. Song, H. Liu, J. Wen, and W. Ma. Learning block importance models for web pages. In *WWW*, 2004.

[24] C. Theriault, N. Thome, and M. Cord. Extended coding and pooling in the hmax model. *IEEE Transactions on Image Processing*, 22(2):764–777, 2013.

[25] L. Torresani and K. Lee. Large margin component analysis. In *NIPS*, 2007.

[26] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair. Learning hierarchical similarity metrics. In *CVPR*, 2012.

[27] K. Weinberger and O. Chapelle. Large margin taxonomy embedding with an application to document categorization. In *NIPS*, 2008.

[28] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.

[29] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2002.

[30] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.