Inter-Concept Distance Measurement with Adaptively Weighted Multiple Visual Features

Kazuaki Nakamura and Noboru Babaguchi

Graduate School of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka, 565-0871 Japan

Abstract. Most of the existing methods for measuring the inter-concept distance (ICD) between two concepts from their image instances use only a single kind of visual feature extracted from each instance. However, a single kind of feature is not enough for appropriately measuring ICDs due to a wide variety of perspectives for similarity evaluation (e.g., color, shape, size, hardness, heaviness, and functions); the relationships between different concept pairs are more appropriately modeled from different perspectives provided by multiple kinds of features. In this paper, we propose extracting two or more kinds of visual features from each image instance and measuring ICDs using these multiple features. Moreover, we present a method for adaptively weighting the visual features on the basis of their appropriateness for each concept pair. Experiments demonstrated that the proposed method outperformed a method using only a single kind of visual feature and one combining multiple kinds of features with a fixed weight.

1 Introduction

Inter-concept distance measurement (ICDM), which is a problem of computing the distance between two concepts, plays an important role in various vision applications including image retrieval, automatic image annotation, indexing, and clustering. For instance, for image retrieval, query suggestion and expansion based on ICDM can help to bridge the *intention gap* [1] between user's search intent and input queries. Also, for automatic image annotation, consistency of annotation tags associated to an image can be improved by ICDM.

In the past decade, several methods for measuring inter-concept distances (ICD) have been proposed [2–9]. These methods can be roughly divided into three categories: ontology-based, text-based, and image instance-based. The last one, on which we focus in this paper, has been most actively studied in recent years. The general procedure of image instance-based ICDM for two concepts u and v is as follows: First, a set of images annotated with u and another annotated with v are gathered. Then visual models of u and v are constructed by using the respective image set. Finally, the dissimilarity score between the two visual models is calculated and used as the ICD between u and v.

In the above procedure, a visual model of each concept is generally constructed using visual features extracted from each image instance. Local descriptors including scale invariant feature transform (SIFT) [10] and speeded

up robust features (SURF) [11] are often used for this purpose, but any kind of visual feature such as color, shape, edge, and texture can also be used. Nevertheless, most image instance-based methods use only a single kind of visual feature for all concept pairs [7, 8]. Actually, there are various perspectives for measuring ICDs. For instance, the ICDs between concrete concepts such as object (e.g., animals and vehicles) can be measured from the perspectives of color, shape, size, heaviness, functions, and so on. ICDM methods should take into account these various perspectives as widely as possible. However, a single kind of visual feature can only provide a single perspective.

Humans adaptively select appropriate perspectives for measuring the ICD for each concept pair. For instance, humans would give a low distance to the concept pair (*cat*, *tiger*) from the perspective of shape, although the sizes of cats and tigers are quite different. This fact indicates that only using multiple kinds of visual features is not enough for measuring good ICDs; it is necessary to adaptively select or weight the features in accordance with their appropriateness for each concept pair. Fan et al. have proposed an ICDM method with multiple kinds of visual features [9], but it equally uses all kinds of visual features with the same level of importance.

In this paper, we propose the combined use of multiple kinds of visual features extracted from each image instance for ICDM. Moreover, we propose a method for adaptively weighting each kind of visual feature based on its appropriateness for each concept pair. This work makes following two novel contributions.

(1) A variety of perspectives are considered due to the use of multiple kinds of visual features.
(2) The most appropriate perspective for measuring the ICD is automatically selected for each concept pair due to the adaptively weighted combination of the multiple visual features.

The remainder of this paper is organized as follows: Section 2 briefly reviews existing ICDM methods, especially focusing on image instance-based ones. Section 3 describes the principle of image instance-based ICDM with the simplest case, where only a single kind of visual feature is used. Section 4 describes the proposed method in detail, which uses an adaptively weighted combination of multiple visual features. Section 5 shows some experimental results, and finally Section 6 concludes this paper.

2 Related Works

As mentioned in Section 1, existing ICDM methods can be roughly divided into three types: ontology-based, text-based, and image instance-based. In every type, each concept is described by a corresponding text word, but their measuring procedures differ completely. We briefly review each type of methods in this section. In the remainder of this paper, we refer to the ICD between two concepts u and v as $\mathrm{ICD}(u, v)$.

2.1 Ontology-based ICDM

The ontology-based ICDM [2, 3] uses a human-defined ontology in which semantic hierarchical relationships between various text words are represented as a tree structure and regards the length of the path between two text words u and v in the tree as $\mathrm{ICD}(u, v)$. One of the most common ontologies used for this purpose is WordNet[1]. ICDs computed by WordNet-based methods are close to human perception since the WordNet is structured by human experts. However, the ontology-based methods can handle only a limited number of concepts, i.e., those that are included in the ontology. To overcome this drawback, the text-based and the image-instance based ICDM use web-scale data sources.

2.2 Text-based ICDM

The text-based ICDM [4–6] counts the co-occurrence frequency of two text words u and v in a web-scale corpus and regards the inverse of the co-occurrence frequency as $\mathrm{ICD}(u, v)$. Since each concept is described by a text word, we can say the text-based methods are more direct than the image instance-based methods. However, text-based methods cannot handle some inter-concept relationships that are trivial for people to handle [8]; two concepts that have a trivial relationship do not always co-occur as text words in a corpus because such relationships have no need to be verbalized. To overcome this drawback, image instance-based ICDM have been studied more actively in recent years as they are expected to be able to handle various kinds of inter-concept relationships implicitly contained in images.

2.3 Image Instance-based ICDM

The image instance-based ICDM is generally achieved by the following two steps:

i. For each concept c, construct its visual model $\mathcal{M}(c)$ by using a set of images annotated with c.
ii. For each concept pair (u, v), compute the dissimilarity score between $\mathcal{M}(u)$ and $\mathcal{M}(v)$ and regard it as $\mathrm{ICD}(u, v)$.

Step i. is the process called "Visual Concept Learning (VCL)."

There have been a number of previous studies on VCL [9, 12–16]. Most of them define a VCL task as a problem of learning a classifier for classifying the presence or absence of each concept c in images. In general, the learning process for each concept c is performed independently of other concepts in a supervised or semi-supervised manner using a number of positive images (those tagged with c) and negative images (those not tagged with c). Discriminative models such as Support Vector Machines (SVM) are often applied for this purpose. Sjöberg et al. use a linear SVM which is computationally light and fast, aiming at real-time applications [15]. Yang et al. use a non-linear SVM with per-sample Multiple

[1] WordNet 3.0, http://wordnet.princeton.edu/

Kernel Learning (MKL), which is an expansion of traditional MKL, for achieving better performance in VCL [13]. Zhu et al. also use SVMs and learn a classifier for each concept in the space spanned by concatenating different kinds of visual features [12]. This allows the classifier to implicitly give an adaptive weight to each kind of visual features. Like Zhu's method, combining multiple kinds of visual features with adaptively determined weights is widely used for VCL.

However, the above framework for VCL is not suitable for ICDM. IDCM is a task of computing the distance between two concepts. Therefore, the appropriateness of each kind of visual features is determined not for each concept c but for each concept pair (u, v). In other words, the appropriateness of a certain feature for the concept u would vary depending on the counterpart concept v. For instance, the distance between *leaf* and *tree* seems to be low because *leaf* is obviously a meronym of *tree*, and it would be appropriately measured with color feature because the color of leaves and trees are both green in most cases. On the other hand, the distance between *leaf* and *flower* also seems to be low because both are organs of plants, but it cannot be appropriately measured with color feature because flowers have a wide variety of colors in contrast to leaves. Rather than the color feature, local texture feature would be more appropriate for the concept pair (*leaf*, *flower*). Nevertheless, a visual model of *leaf* constructed by the above VCL framework always gives the same weights to each kind of visual features, regardless of relations with any other concepts including *tree* and *flower*.

There are a few VCL methods taking into account inter-dependence between concepts [9, 14], but these methods generally do not consider the use of adaptively weighted combination of multiple visual features. For instance, Qi et al. have proposed a technique of cross-category transfer learning for VCL, which learns a classifier

$$f_u(\boldsymbol{x}; v) = \frac{1}{|\mathcal{X}^v|} \sum_{\boldsymbol{x}_i^v \in \mathcal{X}^v} z_i^v \boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}_i^v \tag{1}$$

for a target concept u using an image instance set $\mathcal{X}^v = \{\boldsymbol{x}_i^v | i = 1, 2, \cdots\}$ of a source concept $v \neq u$, where $\boldsymbol{x}_i^v$ is a visual feature extracted from the i-th image instance of the concept v, $z_i^v \in \{1, -1\}$ is the ground truth label of $\boldsymbol{x}_i^v$, and S is a parametric matrix optimized in the learning process [14]. S can be interpreted as the correlation matrix between the target concept u and the source concept v, so it would be related to $\mathrm{ICD}(u, v)$ implicitly. However, since the relation between S and $\mathrm{ICD}(u, v)$ is non-trivial, it is not straightforward to compute $\mathrm{ICD}(u, v)$ based on S. Moreover, their method only uses a single kind of visual feature, i.e., bag of SIFT descriptors. Fan et al. directly compute inter-concept similarity $\gamma(u, v)$ for each concept pair (u, v) and add it to the classifier learning process of SVM for increasing the discrimination power of the classifier [9]. Their inter-concept similarity score $\gamma(u, v)$ can be easily applied to ICDM by computing $\mathrm{ICD}(u, v)$ as $1/\gamma(u, v)$. However, their algorithm for computing $\gamma(u, v)$, in which multiple kinds of visual features are extracted for each image, equally uses all kinds of visual features with the same weight.

One more drawback of the above VCL framework for ICDM is to use discriminative models which often provide complex discriminant hypersurfaces (e.g. non-linear SVM). Actually, it is not a straightforward problem to compute the dissimilarity score between two complex hypersurfaces. Therefore, generative models are more suitable for ICDM. There are a few VCL methods using not discriminative models but generative models. One of them is the method proposed by Zhuang et al. [16], in which semi-supervised Latent Dirichlet Allocation is used instead of SVM. Their method constructs visual models of all concepts as a full probability distribution $p(\boldsymbol{x}, c)$ over concept c and visual feature vector $\boldsymbol{x}$. This construction way for visual models is also adopted in many existing image-instance based ICDM methods [7, 8]. More precisely, the existing ICDM methods construct a visual model of each concept c as $p(\boldsymbol{x}|c) = p(\boldsymbol{x}, c)/p(c)$ with the assumption that prior $p(c)$ is constant. Wu et al. model the $p(\boldsymbol{x}|c)$ as

$$p(\boldsymbol{x}|c) = \sum_h p(\boldsymbol{x}|h)p(h|c) \tag{2}$$

with probabilistic Latent Sematic Analysis (pLSA), where h is a hidden state [8]. Kawakubo et al. also use pLSA, but they model each concept c with not $p(\boldsymbol{x}|c)$ but $p(h|c)$ [7]. In the method of Kawakubo et al., a set of hidden states is shared by all concepts. In these methods, the dissimilarity score between two visual models $p(\boldsymbol{x}|u)$ and $p(\boldsymbol{x}|v)$ is easily computed by information divergence measure. However, in the existing ICDM methods [7, 8], only a single kind of visual feature is used as the vector $\boldsymbol{x}$.

In contrast to the existing methods, we introduce an adaptively weighted combination of multiple visual features into image instance-based ICDM, which is widely considered in many VCL methods as mentioned above.

3 Image Instance-based ICDM with a Single Kind of Visual Feature

To comprehensively describe the principle of image instance-based ICDM, we start with the simplest case, where only a single kind of visual feature is used.

3.1 Notation

Let $\mathcal{C}$ denote the finite set of concepts on which we focus. For each concept $c \in \mathcal{C}$, let I_i^c $(i = 1, \cdots, n_c)$ denote the i-th image instance of concept c, where n_c is the total number of image instances of concept c. Each image is converted into the same kind of d-dimensional feature vector $\boldsymbol{x} \in \mathbb{R}^d$ by some feature extractor. Let $\boldsymbol{x}_i^c$ denote the actual value of feature vector $\boldsymbol{x}$ extracted from image I_i^c. In addition, let $\mathcal{X}^c$ denote the set of feature vectors for concept c; that is, $\mathcal{X}^c = \{\boldsymbol{x}_i^c | i = 1, \cdots, n_c\}$.

3.2 Visual Concept Learning with a Generative Model

As mentioned in Section 2.3, most existing ICDM methods use the following two-step algorithm:

i. For each concept $c \in \mathcal{C}$, construct its visual model as conditional distribution $p(\boldsymbol{x}|c)$ by using $\mathcal{X}^c$.
ii. For each concept pair $(u, v) \in \mathcal{C}^2$, compute the dissimilarity score between $p(\boldsymbol{x}|u)$ and $p(\boldsymbol{x}|v)$ by information divergence measure.

In step i., the probability $p(\boldsymbol{x}|c)$ is generally assumed to be well modeled by a parametric distribution such as a mixture of Gaussians or a categorical distribution. We therefore use $p(\boldsymbol{x}|\boldsymbol{\Theta}^c)$ instead of $p(\boldsymbol{x}|c)$, where $\boldsymbol{\Theta}^c$ is the parameter vector of the parametric distribution.

According to probability theory, modeling each concept c as $p(\boldsymbol{x}|\boldsymbol{\Theta}^c)$ boils down to estimating parameter $\boldsymbol{\Theta}^c$ from the feature vector set $\mathcal{X}^c$. On the basis of MAP estimation, $\boldsymbol{\Theta}^c$ can be estimated as

$$\boldsymbol{\Theta}^c = \mathrm{argmax}_{\boldsymbol{\Theta}} \left[p\left(\boldsymbol{\Theta} \,|\mathcal{X}^c\right) \right] \tag{3}$$

by using parameter variable $\boldsymbol{\Theta}$.

According to Bayes' rule, $p\left(\boldsymbol{\Theta} \,|\mathcal{X}^c\right)$ can be decomposed as

$$p\left(\boldsymbol{\Theta} \,|\mathcal{X}^c\right) = \frac{1}{\lambda} p(\boldsymbol{\Theta}) p\left(\mathcal{X}^c|\, \boldsymbol{\Theta}\right), \tag{4}$$

where λ is the normalization constant for satisfying condition $\int p\left(\boldsymbol{\Theta} \,|\mathcal{X}^c\right) d\boldsymbol{\Theta} = 1$. For convenience of computation, we assume that all image instances are independently observed. This assumption means that each feature vector $\boldsymbol{x}_i^c$ is independent of any other feature vector, so

$$p\left(\mathcal{X}^c|\, \boldsymbol{\Theta}\right) = \prod_{i=1}^{n_c} p(\boldsymbol{x}_i^c|\boldsymbol{\Theta}) \ . \tag{5}$$

On the basis of Formulas (3), (4), and (5), $\boldsymbol{\Theta}^c$ can be estimated as

$$\boldsymbol{\Theta}^c = \mathrm{argmax}_{\boldsymbol{\Theta}} \left[p(\boldsymbol{\Theta}) \prod_{i=1}^{n_c} p(\boldsymbol{x}_i^c|\boldsymbol{\Theta}) \right]. \tag{6}$$

Note that λ can be ignored in the above maximization since it is constant with respect to $\boldsymbol{\Theta}$. This formula can be rewritten as

$$\boldsymbol{\Theta}^c = \mathrm{argmax}_{\boldsymbol{\Theta}} \left[\log p(\boldsymbol{\Theta}) + \sum_{i=1}^{n_c} \log p(\boldsymbol{x}_i^c|\boldsymbol{\Theta}) \right] \tag{7}$$

because of the monotonicity of the log function.

In order to estimate parameter $\boldsymbol{\Theta}^c$ for each concept c based on Formula (7), we model $p(\boldsymbol{\Theta})$ and $p(\boldsymbol{x}|\boldsymbol{\Theta})$ with some parametric distributions.

First, we model $p(\boldsymbol{x}|\boldsymbol{\Theta})$ with a K-dimensional categorical distribution, quantizing feature vector $\boldsymbol{x}$ into K representative values $\{\boldsymbol{y}_k|k=1,\cdots,K\}$ by using a vector quantization method. Actually, we can employ more sophisticated approaches for modeling $p(\boldsymbol{x}|\boldsymbol{\Theta})$ like Kawakubo et al. [7] and Wu et al. [8], but it is not our focus, so that we employ a categorical distribution in this study. In a K-dimensional categorical distribution, model parameter $\boldsymbol{\Theta}$ is defined as K-dimensional real vector $(\theta_1 \ \cdots \ \theta_K)^T$, each of whose element θ_k is corresponding to the probability of $\boldsymbol{x}=\boldsymbol{y}_k$, that is, $p(\boldsymbol{x}=\boldsymbol{y}_k|\boldsymbol{\Theta})=\theta_k$. Trivially,

$$\sum_{k=1}^{K}\theta_k \ = \ 1 \tag{8}$$

is satisfied. Distribution $p(\boldsymbol{x}|\boldsymbol{\Theta})$ can be formulated as

$$p(\boldsymbol{x}|\boldsymbol{\Theta})=\prod_{k=1}^{K}(\theta_k)^{\delta(\boldsymbol{x},\boldsymbol{y}_k)} \ , \tag{9}$$

where δ is Kronecker's delta. Taking the log of both sides, the above Formula (9) is rewritten as

$$\log p(\boldsymbol{x}|\boldsymbol{\Theta}) \ = \ \sum_{k=1}^{K}\delta(\boldsymbol{x},\boldsymbol{y}_k)\log\theta_k \ . \tag{10}$$

Next, we model $p(\boldsymbol{\Theta})$ with a K-order Dirichlet distribution, which is the conjugate prior of the K-dimensional categorical distribution. With a parameter $\boldsymbol{\alpha}=(\alpha_1 \ \cdots \ \alpha_K)^T$, distribution $p(\boldsymbol{\Theta})$ is formulated as

$$p(\boldsymbol{\Theta})=\frac{1}{Z}\prod_{k=1}^{K}(\theta_k)^{(\alpha_k-1)} \ , \tag{11}$$

where Z is the normalization constant for satisfying condition $\int p(\boldsymbol{\Theta})d\boldsymbol{\Theta}=1$. Taking the log of both sides, the above Formula (11) is rewritten as

$$\log p(\boldsymbol{\Theta}) \ = \ -\log Z+\sum_{k=1}^{K}(\alpha_k-1)\log\theta_k \ . \tag{12}$$

On the basis of Formulas (7), (10), and (12), $\boldsymbol{\Theta}^c$ is finally estimated as

$$\boldsymbol{\Theta}^c \ = \ \mathrm{argmax}_{\boldsymbol{\Theta}}\left[\sum_{k=1}^{K}\log\theta_k\left\{\alpha_k-1+\sum_{i=1}^{n_c}\delta(\boldsymbol{x}_i^c,\boldsymbol{y}_k)\right\}\right] \ . \tag{13}$$

Note that $-\log Z$ can be ignored in the above maximization since it is constant with respect to θ_k. Maximization problem (13) subject to constraint (8) can be solved by Lagrange multiplier method, whose solution $\boldsymbol{\Theta}^c=(\theta_1^c \ \cdots \ \theta_K^c)^T$ is explicitly described as

$$\theta_k^c=\frac{1}{\xi}\left\{\alpha_k-1+\sum_{i=1}^{n_c}\delta(\boldsymbol{x}_i^c,\boldsymbol{y}_k)\right\} \tag{14}$$

for each $k \in \{1, \cdots, K\}$, where ξ is the normalization constant for satisfying the constraint (8).

3.3 Dissimilarity Score between Two Visual Models

The dissimilarity score between two distributions $p(\boldsymbol{x}|\boldsymbol{\Theta}^u)$ and $p(\boldsymbol{x}|\boldsymbol{\Theta}^v)$ is generally computed by information divergence measures. A commonly used one is Kullback-Leibler (KL) divergence, which is defined as

$$D_{\mathrm{KL}}[\phi||\psi] = \int p(\boldsymbol{x}|\phi) \log \frac{p(\boldsymbol{x}|\phi)}{p(\boldsymbol{x}|\psi)} d\boldsymbol{x} \tag{15}$$

for two distributions $p(\boldsymbol{x}|\phi)$ and $p(\boldsymbol{x}|\psi)$. However, KL divergence has an undesirable property as a dissimilarity measure, i.e., asymmetry. Therefore, Jensen-Shannon (JS) divergence [17], which is symmetric, has also been used as a dissimilarity measure [7, 8]. The JS divergence between $p(\boldsymbol{x}|\phi)$ and $p(\boldsymbol{x}|\psi)$ is defined as

$$D_{\mathrm{JS}}[\phi||\psi] = \frac{1}{2} \left(D_{\mathrm{KL}}[\phi||\eta] + D_{\mathrm{KL}}[\psi||\eta] \right), \tag{16}$$

where

$$p(\boldsymbol{x}|\eta) = \frac{1}{2} \left\{ p(\boldsymbol{x}|\phi) + p(\boldsymbol{x}|\psi) \right\}. \tag{17}$$

We also use JS divergence and compute the dissimilarity score between two concepts u and v as

$$\mathrm{Dissim}(u, v) = D_{\mathrm{JS}}[\boldsymbol{\Theta}^u||\boldsymbol{\Theta}^v] \ . \tag{18}$$

More specifically, for two model parameters $\boldsymbol{\Theta}^u = (\theta_1^u \ \cdots \ \theta_K^u)^T$ and $\boldsymbol{\Theta}^v = (\theta_1^v \ \cdots \ \theta_K^v)^T$, we compute the dissimilarity score $\mathrm{Dissim}(u, v)$ as

$$\mathrm{Dissim}(u, v) \ = \ \sum_{k=1}^{K} \left\{ \frac{\theta_k^u}{2} \log \frac{2\theta_k^u}{\theta_k^u + \theta_k^v} + \frac{\theta_k^v}{2} \log \frac{2\theta_k^v}{\theta_k^u + \theta_k^v} \right\} \ . \tag{19}$$

4 Image Instance-based ICDM with Multiple Kinds of Visual Features

Here we consider the case in which multiple kinds of visual features are used in conjunction with each other. We use the symbol $\mathcal{S}$ to represent a set of visual feature extractors such as SIFT descriptor and color histogram descriptor. The vector $\boldsymbol{x}_i^c$ is rewritten as $\boldsymbol{x}_i^{c,s}$, which represents the feature vector extracted from image instance I_i^c by extractor $s \in \mathcal{S}$, and the feature vector set $\mathcal{X}^c$ is redefined as $\mathcal{X}^{c,s} = \{\boldsymbol{x}_i^{c,s} | i = 1, \cdots, n_c\}$. The scoring function $\mathrm{Dissim}(u, v)$ in Formula (19) is redefined as $\mathrm{Dissim}(u, v; s)$, which represents the dissimilarity score between concepts u and v computed using feature vector sets $\mathcal{X}^{u,s}$ and $\mathcal{X}^{v,s}$.

4.1 Distance as Weighted Sum of Dissimilarity Scores

There are two popular strategies to fuse multiple kinds of visual features: feature-level fusion and decision-level fusion. In the context of ICDM, the feature-level fusion is achieved by defining a new large vector

$$\boldsymbol{X}_i^c \;=\; \left(w(s_1)\,(\boldsymbol{x}_i^{c,s_1})^{\mathrm{T}} \quad w(s_2)\,(\boldsymbol{x}_i^{c,s_2})^{\mathrm{T}} \quad \cdots\right)^{\mathrm{T}} \tag{20}$$

with a weight set $\mathcal{W} = \{w(s_j)|j = 1, 2, \cdots\}$ for all $c \in \mathcal{C}$ and $i \in \{1, \cdots, n_c\}$, where $s_j \in \mathcal{S}$ for all j, and applying the method of Section 3 on a set of the new vectors $\{\boldsymbol{X}_i^c|i = 1, \cdots, n_c\}$. However, this strategy is not suitable for adaptive weight control, because the weight set $\mathcal{W}$ only can affect the scale of each dimension of the new vector $\boldsymbol{X}$; the general shape of the distribution $p(\boldsymbol{X}|c)$ is not affected by $\mathcal{W}$, which means the information divergence between $p(\boldsymbol{X}|u)$ and $p(\boldsymbol{X}|v)$ for any pair $(u, v) \in \mathcal{C}^2$ does not change with $\mathcal{W}$. Therefore, in this paper, we employ the other strategy, the decision-level fusion, which is achieved by computing $\mathrm{Dissim}(u, v; s)$ separately for each $s \in \mathcal{S}$ and fusing the set $\{\mathrm{Dissim}(u, v; s)|s \in \mathcal{S}\}$ into a single distance measure. For this purpose, we use a weighted-sum method, a simple yet commonly used fusion method.

As shown in Figure 1, our fused distance measure gives $\mathrm{ICD}(u, v)$ as

$$\mathrm{ICD}(u, v) = \sum_{s\in\mathcal{S}} w(s; u, v)\mathrm{Dissim}(u, v; s) \tag{21}$$

by using the weighted sum scheme, where $w(s; u, v)$ denotes the weight for $\mathrm{Dissim}(u, v; s)$. The weight set $\{w(s; u, v)|s \in \mathcal{S}\}$ always satisfies

$$\sum_{s\in\mathcal{S}} w(s; u, v) = 1, \tag{22}$$

but it is determined differently for each concept pair (u, v). If the feature vector set extracted using extractor s is more appropriate for modeling the relationship between u and v, a larger value is assigned to $w(s; u, v)$.

4.2 Weight Determination based on Dissimilarity Itself

Now the problem is how to determine weight $w(s; u, v)$ for each concept pair $(u, v) \in \mathcal{C}^2$ and each extractor $s \in \mathcal{S}$. In other words, the problem is how to evaluate the appropriateness of each extractor s for modeling the relationship between concepts u and v.

In the context of VCL, "appropriateness" can be simply defined as "degree of correlation." This is because if the feature vectors extracted by extractor s are deeply correlated with the presence or absence of concept c, the extractor s is appropriate for the concept c, and vice versa. Hence the appropriateness of each extractor s for each concept c is implicitly given in the process of supervised learning for the concept c. However, in the context of ICDM, the "appropriateness" is determined not for each concept but for each concept pair, as mentioned

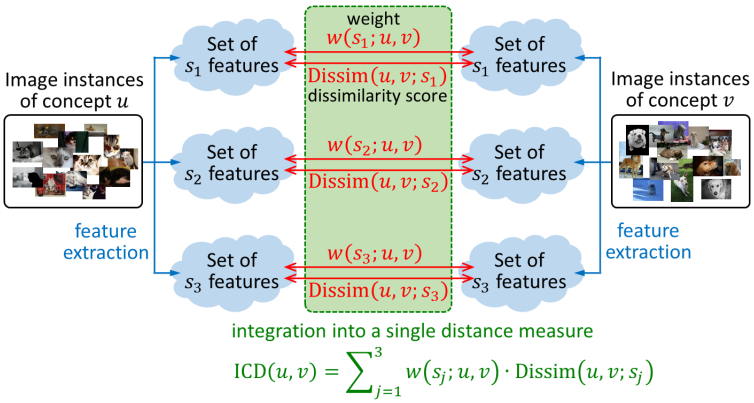

Fig. 1. Overview of method proposed for measuring ICDs

in Section 2.3. In this case, "appropriateness" differs from "degree of correlation," and therefore it can hardly be given by supervised learning. We therefore employ a non-learning based approach in this study.

As mentioned in Section 1, there are a wide variety of perspectives for evaluating the similarity between two concepts. This means that the similarity score between two concepts would vary depending on the perspective. For instance, the similarity between *car* and *train* would likely be high from the perspective of function, but it would likely be low from the perspective of shape. On the other hand, the similarity between *cat* and *tiger* would likely be high from the perspective of shape, but it would likely be low from the perspectives of size and color. For such concept pairs, how should the semantic distance between them be determined? Generally speaking, humans would give a low distance to such concept pairs. *Car* and *train* would likely be considered similar in spite of their dissimilarity in shape, and *cat* and *tiger* would likely be similar in spite of their dissimilarity in size and color. Analogizing from these instances, we hypothesize that the extractor s having lower $\mathrm{Dissim}(u, v; s)$ is more important for modeling the relationship between u and v. In other words, $\mathrm{Dissim}(u, v; s)$ itself can be used as the indicator for evaluating the appropriateness of extractor s for concept pair (u, v).

In our method, we define appropriateness indicator $a(s; u, v)$ of extractor s for concept pair (u, v) as

$$a(s; u, v) = \frac{1}{\epsilon + \mathrm{Dissim}(u, v; s)} \ , \tag{23}$$

where ϵ (> 0) is a regularization term for avoiding division by zero. The lower the $\mathrm{Dissim}(u, v; s)$, the higher the $a(s; u, v)$ due to their inverse relationship. If $a(s; u, v)$ is high, extractor s is considered to be appropriate for modeling the

relationship between u and v. Using $a(s;u,v)$, we determine weight $w(s;u,v)$ as

$$w(s;u,v) = \frac{w'(s;u,v)}{\sum_{s\in\mathcal{S}} w'(s;u,v)} \ , \tag{24}$$

where

$$w'(s;u,v) = (a(s;u,v))^{\beta} \ . \tag{25}$$

Parameter β (≥ 0) is used to adjust the balance between weights; the smaller the β, the more uniform the weights. If $\beta = 0$, the same weight is given to all descriptors. If $\beta = +\infty$, $w(\hat{s};u,v) = 1$ is given to descriptor $\hat{s}$ such that $\hat{s} = \mathrm{argmax}_{s\in\mathcal{S}}\{a(s;u,v)\}$, and zero weight is given to all other descriptors.

5 Evaluation

We evaluated the effectiveness of the proposed method experimentally in comparison with two other methods: (A) using only a single kind of visual feature and (B) combining multiple kinds of visual features with the same fixed-weight used for all descriptors. We refer to the comparative method (B) as "Simple avg." method in this section. The performance of each method was evaluated on the basis of correlation coefficient with JCN [2], which we refer to as CCJ in this section. JCN is a representative ontology-based ICDM method using WordNet and can provide ICDs close to human perception; so JCN has been often used as the basis for performance evaluation in previous works [6, 8].

5.1 Experimental Setting

We first gathered 2,000 commonly used English nouns from the *Corpus of Contemporary American English*[2] as a candidate set of concepts. The nouns not included in WordNet were excluded. Next, the 400 nouns that had been most frequently used as annotation tags in Flickr[3] were selected from the candidate set and used to construct concept set $\mathcal{C}$. Then, for each of the 400 concepts, the corresponding image instances were collected from Flickr. The number of collected images ranged from 249 to 1868 per concept, and the total number was 379,294. From each of the collected images, we extracted four kinds of visual feature: HSV color histogram (HSV Histo), GIST [18], GLCM [19], and bag of visual words (BoVW) using the local descriptors proposed by Wu et al. [8]. Note that the comparative method using only BoVW simulates the result of Wu's method. Using these 4 kinds of visual features, we computed the ICDs for all combination of the 400 concepts. The total number of the considered concept pairs was $(400 \times 399)/2 = 79800$.

The proposed method has four parameters: K, $\boldsymbol{\alpha} = (\alpha_1 \ \cdots \ \alpha_K)^T$, β, and ϵ. Empirically, K was set to 100, β was set to 8, and ϵ was set to 0.01 in this experiment. As to $\boldsymbol{\alpha}$, we assign the same value $\bar{\alpha}$ to all α_k and set the $\bar{\alpha}$ to 50.

[2] Corpus of Contemporary American English, http://www.wordfrequency.info/
[3] Flickr, http://www.flickr.com/

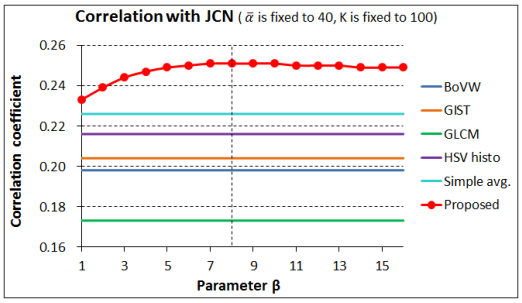

Fig. 2. Experimental results with several settings of β

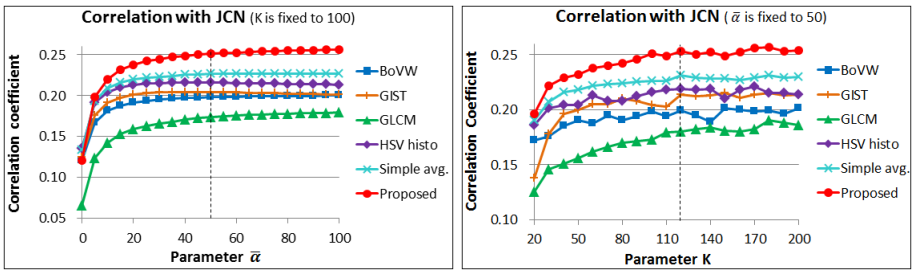

Fig. 3. Results of additional experiments with several settings of K and $\bar{\alpha}$. K is fixed to 100 in the left figure and $\bar{\alpha}$ is fixed to 50 in the right figure.

5.2 Results and Discussion

The CCJ of the proposed method and that of each comparative method are plotted in Figure 2, in which the horizontal axis represents parameter β. Since β is not related to comparative methods (A) and (B), their results are represented as lines parallel to the horizontal axis. The CCJs of comparative methods (A) ranges from 0.17 to 0.21. These were outperformed by Simple avg. method, whose CCJ is 0.226. This indicates that it is important for ICDM to consider a variety of perspectives which can be provided by the use of multiple kinds of visual features. Moreover, Simple avg. method was outperformed by the proposed method, whose CCJ is 0.251 for $\beta = 8$. This demonstrates that using an adaptively weighted combination of multiple visual features is further effective than the simple combination with the same fixed-weight.

The CCJ of the method using only BoVW, which simulates the result of the method of Wu et al. [8], is 0.198. This is much lower than the result reported in [8]. This is because the method described in Section 3 does not adopt a complex way for modeling $p(\boldsymbol{x}|\boldsymbol{\Theta})$ unlike Wu's study. Using more sophisticated approaches for modeling $p(\boldsymbol{x}|\boldsymbol{\Theta})$ for each kind of visual feature would improve the effectiveness of the proposed method.

Table 1. Average rank of 80 concept pairs that are judged as similar by humans

BoVW	GIST	GLCM	HSV histo	Simple avg.	Proposed
11183.2	10946.6	12132.8	10730.0	9940.7	**9438.9**

We examined the effects of parameters K and $\bar{\alpha}$ in additional experiments. Figure 3 (left) shows the CCJ of each method with several settings of $\bar{\alpha}$ and fixed K. All of the shown CCJ is seriously decreased for small $\bar{\alpha}$, especially for $\bar{\alpha} \leq 20$. This indicates the possibility that the size of collected image set was not enough. The method described in Section 3 tends to cause the overfitting of $\boldsymbol{\Theta}^c$ to image set $\mathcal{X}^c$ if the size of $\mathcal{X}^c$ is too small. Large $\bar{\alpha}$ is helpful for avoiding this overfitting problem. Hence, the CCJ of each method increases with larger $\bar{\alpha}$. However, if the size of $\mathcal{X}^c$ is enough large, the overfitting problem does not occur so seriously. On the other hand, Figure 3 (right) shows the CCJ of each method with several settings of K and fixed $\bar{\alpha}$. Roughly speaking, the larger the K, the more improved the performance of each method. However, for $K \geq 120$, we can see little change in the CCJ of each method. This indicates that K should be set to more than 1/8 of the size of a training set since the number of collected image instances per concept in this experiment is 948 on an average. Note that the proposed method outperformed the comparative methods (A) and (B) for any settings of parameters K and $\bar{\alpha}$.

In the above experiments, all methods including the proposed one had not high CCJ: at most 0.26. This is mainly because the image instance sets gathered from Flickr included a non-negligible number of junk images, i.e., ones unrelated to the annotation tags. Removing such junk images from the image instance sets [20] as a pre-process would improve the effectiveness of the proposed method.

5.3 Comparison with Human Perception

Although JCN can provide ICDs close to human perception, there is not complete agreement between them. Therefore we also experimentally compared the result of each method with human perception.

In this experiment, we first picked up 783 concept pairs from all the 79800 concept pairs mentioned in Section 5.1. Next, we showed the 783 pairs to 4 people and instructed them to judge whether each pair is semantically similar or not. As a result, 80 pairs out of the 783 pairs were judged as similar by all the 4 people. On the other hand, we ranked the original 79800 concept pairs in order of ICD computed by the proposed method, and created a ranked list. For comparative methods (A) and (B), we created ranked lists in the same way. Then we calculated the average rank of the above 80 concepts in each ranked list. The average rank would be low if the corresponding method is effective. Table 1 shows the result.

The result shown in Table 1 strongly supports the discussion in Section 5.2. The average rank of Simple avg. method is lower than that of comparative

methods (A), and that of the proposed method is lowest. This also demonstrates that using an adaptively weighted combination of multiple visual features is effective.

6 Conclusion

We proposed measuring image instance-based inter-concept dissimilarity by using multiple kinds of visual features and combining them into a single distance measure using adaptively determined weights. Analogous to how people judge ICDs, it determines the weight for each feature s in accordance with the dissimilarity score calculated using s itself. The experimental results show that the proposed method outperforms a method using only a single kind of feature and one combining multiple kinds of features with a fixed weight. We will further improve its performance by devising and using a method for removing junk images from the image instance sets.

Acknowledgement. This work was supported in part by a Grants-in-Aid for Scientific Research from the Japan Society for the Promotion of Science.

References

1. Zha, Z.J., Yang, L., Mei, T., Wang, M., Wang, Z., Chua, T.S., Hua, X.S.: Visual query suggestion: Towards capturing user intent in internet image search. ACM Transactions on Multimedia Computing, Communications, and Applications **6** (2010)
2. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet::similarity - measuring the relatedness of concepts. In: Proceedings of 5th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04). (2004) 38–41
3. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics **32** (2006) 13–47
4. Cilibrasi, R.L., Vitányi, P.M.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering **19** (2007) 370–383
5. Liu, D., Hua, X.S., Yang, L., Wang, M., Zhang, H.J.: Tag ranking. In: Proceedings of the 18th International Conference on World Wide Web. (2009) 351–360
6. Mousselly-Sergieh, H., Döller, M., Egyed-Zsigmond, E., Gianini, G., Kosch, H., Pinon, J.M.: Tag relatedness using laplacian score feature selection & adapted jensen-shannon divergence. In: Proceedings of the 20th International Conference on MultiMedia Modeling. (2014) 159–171
7. Kawakubo, H., Akima, Y., Yanai, K.: Automatic construction of a folksonomy-based visual ontology. In: Proceedings of the 6th IEEE International Workshop on Multimedia Information Processing and Retrieval. (2010) 330–335
8. Wu, L., Hua, X.S., Yu, N., Ma, W.Y., Li, S.: Flickr distance: A relationship measure for visual concepts. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 863–875
9. Fan, J., He, X., Zhou, N., Peng, J., Jain, R.: Quantitative characterization of semantic gaps for learning complexity estimation and inference model selection. IEEE Transactions on Multimedia **14** (2012) 1414–1428

10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Internal Journal of Computer Vision **60** (2004) 91–110
11. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: Proceedings of the 9th European Conference on Computer Vision. (2006) 404–417
12. Zhu, S., Wang, G., Ngo, C.W., Jiang, Y.G.: On the sampling of web images for learning visual concept classifiers. In: Proceedings of the 9th ACM International Conference on Image and Video Retrieval. (2010) 50–57
13. Yang, J., Li, Y., Tian, Y., Duan, L.Y., Gao, W.: Per-sample multiple kernel approach for visual concept learning. EURASIP Journal on Image and Video Processing **2010** (2010) 1–14
14. Qi, G.J., Aggarwal, C., Rui, Y., Tian, Q., Chang, S., Huang, T.: Towards cross-category knowledge propagation for learning visual concepts. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Rrecognition. (2011) 897–904
15. Sjöberg, M., Koskela, M., Ishikawa, S., Laaksonen, J.: Real-time large-scale visual concept detection with linear classifiers. In: Proceedings of the 21st International Conference on Pattern Recognition. (2012) 421–424
16. Zhuang, L., Gao, H., Luo, J., Lin, Z.: Regularized semi-supervised latent dirichlet allocation for visual concept learning. Neurocomputing **119** (2013) 26–32
17. Fuglede, B., Topsøe, F.: Jensen-shannon divergence and hilbert space embedding. In: Proceedings of the 2004 International Symposium on Information Theory. (2004) 30
18. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. In: Progress in Brain Research. Volume 155. (2006) 23–26
19. Soh, L.K., Tsatsoulis, C.: Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. IEEE Transactions on Geoscience and Remote Sensing **37** (1999) 780–795
20. Zhu, S., Ngo, C.W., , Juang, Y.G.: Sampling and ontologically pooling web images for visual concept learning. IEEE Transactions on Multimedia **14** (2012) 1068–1078