**ProbSim-Annotation**: A novel image annotation algorithm using a probability-based similarity measure

Chunsheng Fang, Anca Ralescu

Machine Learning and Computational Intelligence Laboratory
Department of Computer Science, ML 0030
University of Cincinnati, Cincinnati, Ohio, 45220
fangcg@email.uc.edu, Anca.Ralescu@uc.edu

**Abstract**

*ProbSim-Annotation* is an image annotation algorithm for heterogeneous data driven by a probability-based similarity assessment. Image annotation consists of associating to an image a description in terms of labels (or words) from a dictionary. This association rests on the premise that similar images have similar annotations. Evaluation of an annotation algorithm is conveyed by the relevance of retrieval from an image database when the query is described by labels of the dictionary. Previous studies have shown the probability-based similarity to be very useful in assessing similarity between heterogeneous data by mapping heterogeneous distances into their probability distributions. These can then be estimated from the training set. In this paper, for practical use, the empirical CDF of distances is approximated by polynomial series. Combining probability-based similarity across multiple attributes/dimensions leads to an overall similarity. This serves as an important cue to transfer annotations from the training set to a test set using a kNN algorithm. Experimental results performed on Corel5K benchmark dataset show that *ProbSim-Annotation* is a promising image annotation algorithm.

**Introduction**

One of the important problems in computer based image understanding is mapping low-level physical features of an image into an image description using a higher level language. One can say that the ultimate goal is for the higher level language to coincide with natural language. The goal of this mapping is to achieve an “understanding” of the image similar to that of a subject looking at the image. Moreover, for practical use these description must be accurate enough to support meaningful retrieval results whenever they are used as queries from an image data set. Interest in image annotation has increased drastically in the wake of on-line information processing and retrieval. However, research in *linguistic image understanding* goes back several years. Indeed this was the objective of a series of studies on image understanding carried out at the Laboratory for International Fuzzy Engineering Research (LIFE) in Yokohama, Japan, during the years 1991-1995 (see for example (Ralescu 1995) and references therein). Most of the current state-of-the-art models are exploiting multiple image features such as *color*, *texture*, and hoping to find a good similarity measurement to combine these heterogeneous features. The probability-based similarity assessment (Le and Ho 2004), (Ralescu, Popovici, and Ralescu 2008), (Popovici 2008), (Fang et al. 2008), consists of two distinct steps: (1) first similarity along each image feature is computed as a probability of certain event in the domain corresponding to that feature; (2) similarities (in fact probabilities) obtained in step (1) are combined across image features. A training set is used to estimate the distribution underlying the domains of each feature. Multiple (often redundant) image features are usually employed to better measure the image similarity in annotation task (Duygulu et al. 2002) (Makadia, Pavlovic, and Kumar 2008). Similarity assessment is one of the most critical parts in image annotation. Its appeal for image annotation rests on the following assumptions:

- Similar images are more likely to share the same keywords.
- Multiple and heterogeneous image features are more likely to effectively capture meaningful similarity between two images.

**Standard distance combining technique**

$L_1$, $L_2$ distances are widely used as distance measures. However, for multi-dimensional data, the heterogeneity problem is more critical since we are willing to combine data channels that are from totally different sources, e.g. different image features.

Two standard distance combining techniques are reported by (Makadia, Pavlovic, and Kumar 2008). Since these are used for comparison purposes with the proximity measure proposed here, a review of these distances follows.

**Joint Equal Contribution (JEC) (Duygulu et al. 2002)**

Each feature contributes equally towards the image distance, and JEC scales each feature distance by their empirically estimated upper and lower bounds into within $[0, 1]$. Combined distance between image $I_i$ and $I_j$ is denoted as $d(i, j)$, original distance as $d^k_{(i,j)}$, and scaled distance as $\tilde{d}^k_{(i,j)}$.

\[
\tilde{d}^k_{(i,j)} = \frac{d^k_{(i,j)}}{\max_{i,j} (d^k_{(i,j)})} \in [0, 1],
\]

\[
d(i, j) = \frac{1}{N} \sum_{k=1}^{N} \tilde{d}^k_{(i,j)},
\]
where \( k \) denotes different image features, \( k = 1, \ldots, N \).

**L1-Penalized Logistic Regression (Lasso)** Lasso (Tibshirani 1996), (Hastie, Tibshirani, and Friedman 2001) is another approach based on \( L_1 \) to capture image similarity by applying regression on the created training set containing pairs of similar and dissimilar images, as positive and negative samples.

In the remaining sections, we briefly illustrate the idea of probability based similarity measure for heterogeneous data. Then an annotation baseline method is introduced, followed by experiment settings and benchmarking.

### Probability-based Similarity Measures

Probability-based Similarity is obtained from the probability distribution of the training data, more precisely, from the probability distribution of pairwise distances, and is defined as follows:

\[
Sim_F(a, b) = P(d(X, Y) > d(a, b))
\]

where \( X, Y \) are independent observations from a domain \( D \), with distribution \( F \), and \( a, b \) are two particular values from \( D \) whose similarity is being assessed, and \( d \) is a distance measure defined on this domain. In words, equation (3) states that the similarity between \( a \) and \( b \) is the probability of an arbitrary pair \( (X, Y) \) such that the distance between \( X \) and \( Y \) is greater than the distance between \( a \) and \( b \). From equation (3) it follows

\[
Sim_F(a, b) = 1 - P(d(X, Y) \leq d(a, b)) = 1 - F_{d(X,Y)}(d(a, b))
\]

where \( F_{d(X,Y)} \) denotes the distribution function of \( d(X, Y) \) induced by the distribution \( F \) on \( D \). \( F_{d(X,Y)}(d(a, b)) \) can be computed for various \( F \), the distribution function of \( X \) and \( Y \), distances \( d \). For instance is \( d(a, b) = |a - b| \)

\[
F_{d(X,Y)}(d(a, b)) = F_{X-Y}(|a - b|) - F_{X-Y}(-|a - b|)
\]

where \( F_{X-Y} \) is the distribution function of \( X - Y \) when \( X, Y \) are iid \( F \). In particular if \( F \) is the cumulative distribution function for Normal distribution with some mean \( \mu \) and variance \( \sigma^2 \), then \( X - Y \) is distributed according to Normal distribution mean 0 and variance \( 2\sigma^2 \), and hence

\[
Sim_{N(\mu, \sigma^2)}(a, b) = 1 - F_{N(0, 2\sigma^2)}(|a - b|) + F_{N(0, 2\sigma^2)}(-|a - b|)
\]

As shown in (Fang et al. 2008), even for two identical domains with distribution coming from the same family of distributions (e.g. Normal) but with different parameters, the similarity measure between pairs of values from each of these domains are different.

It is easy to see from the above that knowing the distribution (CDF) is critical in each heterogeneous feature. However, often, in practice, this is not known. Instead the Empirical Cumulative Distribution Function (ECDF) can be obtained from the data which, in turn can be used to approximate a close form CDF.

![PDF of L1 distance for RGB feature](image343x587 to 534x738)

**Polynomial Approximation of CDF**

One way to approximate the distribution underlying the data, or of the pairwise distances between data points, is by using a mixtures of Gaussian distributions, in which case the density function (PDF) and distribution function can be written as a weighted sum of the various Gaussian PDFs and CDF respectively.

In this paper, we pursue an alternative approach of polynomial approximation of the CDF. Taylor series (Equation 4) can approximate any infinitely differentiable function \( F(x) \) in a neighborhood of value \( a \), so we could apply polynomial fitting to estimate the probability distributions of all pairwise distances.

\[
F(x) = \sum_{n=0}^{\infty} \frac{F^{(n)}(a)}{n!} (x - a)^n
\]

Therefore we can nicely approximate the ECDF using Least Square polynomial fitting with a small Root Mean Square Error (RMSE). It turns out we only need a 9th-degree polynomial to well approximate with a small error \( RMSE \approx 0.0016 \). It means that we could obtain an Approximated CDF (ACDF) of the form like Equation 5.

\[
F(x) \approx \sum_{k=0}^{8} p_k x^k
\]

In Figure 2, we illustrate the ECDF and the 9th-degree polynomial approximation for all pairwise \( L_1 \) distances of RGB histogram features in Corel5K dataset.

### Combining probability-based similarity across features

After we map the distance into a probability, we are facing the question: "How to combine these probability together?" We investigate here two different ways of combination across features. Assume that the similarities along \( \delta \) features are computed as \( S_1, \ldots, S_\delta \) (recall that there are actually probabilities). Assuming independence among features (an assumption which would be satisfied for such features as color and texture), the over all similarity between
two data points \( a \) and \( b \) would be obtained by multiplying these individual similarities. That is

\[
ProbSim_{\Pi}(a, b) = \prod_{i=1}^{k} S_i(a, b) \tag{6}
\]

where \( a = (a_1, \ldots, a_k), b = (b_1, \ldots, b_k) \), and \( S_i = \text{Sim}_F(a_i, b_i) \). The properties of \( ProbSim_{\Pi} \) follow from those of \( S_i \).

Note that from equation (6) it follows that

\[
ProbSim_{\Pi}(a, b) = 1 \iff S_i(a, b) = 1, \forall i = 1, \ldots, k.
\]

That is, in words, this means that \( a \) and \( b \) are similar if and only if \( a \) and \( b \) are similar (with similarity 1) across all features. Likewise \( ProbSim_{\Pi}(a, b) = 0 \iff \exists i_0 \text{ such that } S_{i_0} = 0 \). Since the quantities \( 0 \leq S_i \leq 1 \), depending on the value \( k \), their product can rapidly approach 0, and become eventually quite useless.

The usual alternative in such cases is to replace \( \prod_{i=1}^{k} S_i(a, b) \) by \( \alpha \log(\prod_{i=1}^{k} S_i(a, b)) \) where \( \alpha < 0 \) ensures that the result is a positive value. With \( \alpha = -2 \), further algebra leads to the expression \( \sum_{i=1}^{k} \log(\frac{1}{S^2_i}) \), which is the Fisher transform, whose distribution is \( \chi^2(2k) \). I.e. the \( \chi^2 \) distribution with \( 2k \) degrees of freedom. The overall similarity value is calculated as the \( p \)-value of this distribution corresponding to the quantity \( \sum_{i=1}^{k} \log(\frac{1}{S^2_i}) \) computed from the data. We denote the result as \( ProbSim_{\chi^2}(a, b) \). That is,

\[
ProbSim_{\chi^2}(a, b) = P(\chi^2_{2k} \geq \sum_{i=1}^{k} \log(\frac{1}{S^2_i})) \tag{7}
\]

Note that the case when one of \( S_i = 0 \) is treated by replacing it by a very small value \( \epsilon \) (see (Popovici 2008) for a discussion of this case). It should be noted here that in this approach underlying normal distribution, and independence of features are assumed.

**Image annotation baseline method**

As in the recent paper (Makadia, Pavlovic, and Kumar 2008), the baseline method formulates image annotation as a label transfer problem using k-Nearest Neighbor. The neighborhood structure is modeled using different low-level image features.

**Features**

Some simple image features are used to describe the low-level image statistics. Since color and textures are the most common low-level visual cues for image representation, they are also used in this study. We use the following features:

- **Color:** RGB histogram, \( 1 \times 48 \) vector (each channel has 16 bins); HSV histogram, \( 1 \times 48 \) vector;
- **Texture:** Haar wavelet, \( 1 \times 32 \) vector; Gabor wavelet, \( 1 \times 32 \) vector;

**Similarity measures**

In (Makadia, Pavlovic, and Kumar 2008), JEC and Lasso are proposed as distance combining techniques. Our proposed Probability-based similarity can be plugged in after the \( L_1 \) pairwise computation, and using Equation 3, we can have the \( ProbSim \) which is also pairwise.

**kNN Label Transfer**

A greedy 2-stage label transfer baseline algorithm is proposed in (Makadia, Pavlovic, and Kumar 2008), which is as follows:

- For query image \( I_q \), k-Nearest Neighbor images are retrieved based on Probability based similarity, which are denoted as \( \{I_1, I_2, \ldots I_k\} \), where \( I_1 \) is the most similar image to \( I_q \);
- Label Transfer from the annotations for retrieval set \( \{I_1, I_2, \ldots I_k\} \) to query image \( I_q \)
  - Sort annotations of \( I_1 \) based on their global frequency estimated from training set;
  - For \( |I_1| \), transfer the \( n \) out of \( |I_1| \) highest annotations to \( I_q \). If \( I_1 < n \), proceed to next step; otherwise done;
  - For annotations for \( \{I_2, \ldots I_k\} \), select the highest ranking \( n - |I_1| \) annotations based on their local frequency.

We apply this simple but efficient baseline algorithm for our image annotation.

**Experiments and Discussion**

**Experiment Setup**

We evaluate \( ProbSim \)-annotation algorithm on de-facto benchmark dataset, Corel5K (Duygulu et al. 2002) which consists of 5,000 annotated images collected from the larger Corel CD set. Among these images 4,500 samples are randomly selected as training set, and 500 are left as testing set. Each result reported here is the average of 5 rounds runs. Some statistics for Corel5K are listed below:

- Each image has on average 3.5 and no more than 5 annotations.
• The dictionary contains 374 words but only 260 words are used.
• 5,000 images fall into 50 folders, each of which contains 100 conceptually similar images.

Performance Evaluation

We evaluate ProbSim-annotation algorithm based on 4 metrics:

• **Precision (P)**: The annotation precision for a keyword is defined as the number of images assigned the keyword correctly divided by the total number of images predicted to have the keyword.

\[
P = \frac{\# \text{ images assigned keyword correctly}}{\# \text{ images predicted to have the keyword}}
\]

• **Recall (R)**: The annotation recall is defined as the number of images assigned the keyword correctly, divided by the number of images assigned the keyword in the ground-truth annotation.

\[
R = \frac{\# \text{ images assigned keyword correctly}}{\# \text{ images assigned the keyword in ground-truth}}
\]

• **Classification (C)**: The folder id can be roughly taken as classes. Therefore we can approximately evaluate the classification performance as dividing the number of images predicted as most similar by their real folder id. Note that this metric is not standard in image annotation literature, we use it only as another point of view to our system performance.

• **Recalled Keywords (N_{+})**: Number of keywords recalled by the system is measuring the ability to cover the whole dataset.

Table 1 shows the results for precision, recall, classification and recalled key words when each of the combination methods $\text{ProbSim}_{\chi^2}$ and $\text{ProbSim}_{\prod}$ are used. We compare these results with the results reported in (Makadia, Pavlovic, and Kumar 2008) focusing on three different aspects:

- Performance along each features separately.
- Performance reported in recent publication (Makadia, Pavlovic, and Kumar 2008)
- Our proposed $\text{ProbSim}$ with $\chi^2$ combination, $\text{ProbSim}_{\chi^2}$, and the product combination, $\text{ProbSim}_{\prod}$.

We can see from Table 1 that:

- $\text{ProbSim}_{\prod}$ is slightly better than $\text{ProbSim}_{\chi^2}$.
- $\text{ProbSim}$ has competitive or similar precision and recalled keywords performance
- $\text{ProbSim}$ has outperformed others with much better recall performance.

![Figure 3: kNN neighborhood size and Precision, Recall, Classification rate](image)

**Dependence on the kNN neighborhood size**

kNN is employed to select most important "Candidates" from the query image’s neighborhood, therefore, the size of the neighborhood may influence the annotation result. Figure 3 shows the relationship between kNN neighborhood size and Precision, Recall, Classification rate.

We can see that as the size of neighborhood grows:

- Recall rate becomes higher because more coverage is achieved by adding more neighbors;
- Precision rate falls a little, because some incorrect annotations may be brought in by wrong neighbors;
- The classification rate seems independent of size of neighbors.

**ProbSim-Annotation results**

Comprehensive experiments and results are illustrated in this section. In each of the figures below, above the image, is the file ID in Corel5K dataset. The first line below the image consists of predicted annotations, while the second line consists of ground-truth annotations.

**Comparing ProbSim with JEC:** Though $\text{ProbSim}$ and JEC have similar performance in precision, it is still worthy to compare them through different scenarios. In Figure 4, $\text{ProbSim}$ is better than JEC since JEC brings in the totally
unrelated word "boats". In Figure 5, both have the same predictions, while the image neighbors are a little different.

In Figure 6, ProbSim and JEC both malfunction, bringing some totally unrelated keywords such as "jet", "birds".

Comparing ProbSim using only color or texture features with all features: In Figure 7 and 8, the retrieved neighbors are not good enough to transfer the annotations to the query image. Using all of them, figure 9, we obtain a higher quality annotation result, noting that "water" is brought in.

More results by ProbSim: Figures 10, 11, and 12 show more experimental results obtained using ProbSim.

Figure 4: ProbSim outperform JEC

Figure 5: ProbSim and JEC have different neighbors but same predicted annotations

Figure 6: ProbSim and JEC have wrong predicted annotations

Figure 7: Using only color features

Figure 8: Using only texture features

Figure 9: Using all color and texture features
Conclusion and Future Work

In this paper, we propose a ProbSim-Annotation, a probability-based image annotation algorithm. We illustrate its performance on a series of experiments on the Corel5K data set. This paper can be further extended in several aspects including (1) design of ProbSim image annotation system architecture; (2) High dimensional distance distribution where more advanced mathematical tools e.g. manifold can be employed; (3) Investigate deeper the effect of independence (or lack of) between features on the overall result, and design combination methods that can take this into account.

Acknowledgments

This work was partially supported by the Department of the Navy, Grant ONR N000140710438 and Ohio Board of Regent (OBR) Graduate Scholarship. The author also appreciate Dr. Kobus Barnard (Univ. of Arizona) sharing Corel5K dataset and valuable comments, Shaoting Zhang (Rutgers Univ.) sharing some of his Matlab codes for benchmarking, and Junjun Yu’s discussions (University of Wisconsin-Madison) about approximation technique.

References


Figure 10: ProbSim Annotation results 1

Figure 11: ProbSim Annotation results 2

Figure 12: ProbSim Annotation results 3