Bayesian extensions to person ReID: a discussion document.

Jeremy L. Wyatt and Ferdian Jovan

October 28, 2017

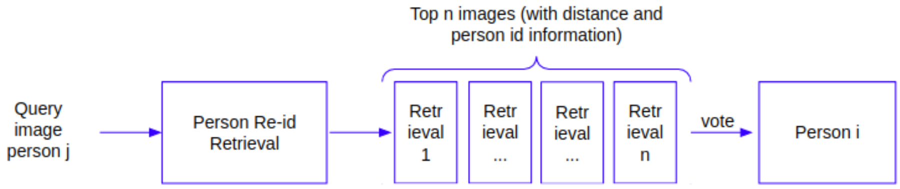
1 Introduction

In this short document we sketch some ideas about how to use simple Bayesian reasoning to use a person ReID system to provide a posterior over the persons. Please note that this is a discussion document. We have given the problem some thought, but we cannot promise that our suggestions are correct, feasible to implement efficiently, or that they will produce improved performance. Nonetheless, this is a place to start.

We first assume that we split the data into a training set, a validation set, and a test set. The training set is used to train the ReID system. The validation set is used to train the observation model for Bayesian inference. The test set is used to ascertain how good the performance of the Bayesian ReID system is. Each observation model gives the conditional probability of the output from the algorithm given the true ID. So the output of the ReID pipeline is termed an observation. Each proposal essentially differs in the observed output of the ReID pipeline. We suggest these alternatives: (i) the winning ID after voting by the retrieved images; (ii) the distribution of votes from the retrieved images over the IDs, and (iii) the vector of distances between the query image and some hypotheses about the ID. We also assume, in all cases, that the number of persons is known and fixed. New persons are classified as instances of a "novel person" class. In each case we have tried to build our extension on top of the existing person ReID system's output, i.e. the retrieval of a series of pictures, with associated IDs, from the training data. We also use the distance metric between instances that is described in the triplet-loss paper [2].

The query image is x. If the output class of a ReID system is i then $o(x) = o_i$, or just o_i. The true person (class) j of the query image is $c(x) = j$, or c_j. There are K persons, including the novel person class. The set of retrieved images is R. The subset of retrieved images corresponding to some person j is R_j. To train the observation models we need examples of novel persons in the validation set, i.e. persons that the ReID system was not trained on.

2 Winning ID as observed output



The first problem is how to massage the output of the baseline retrieval system into an output ID o_i. Here, we simply take the top n retrievals, and they vote. The winning ID is the one with the most votes. The output of the person ReID system is thus as specified in figure above. Therefore, to learn the observation model, we run the ReID retrieval and voting pipeline on the validation set. The observation model is simply derived from the resulting multi-class $(K-1 \times K)$ confusion matrix. Each cell states how likely an input image of person j is–after the voting procedure–to be classified as class i. A trivial application of Bayes' rule gives the posterior over the true class:

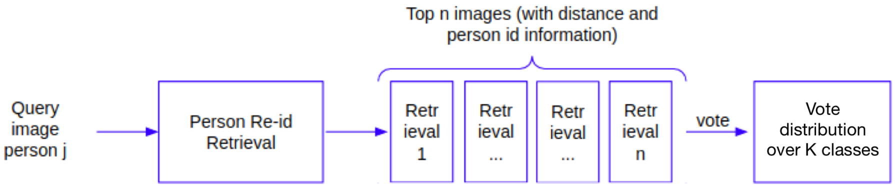
$$P(c_j \mid o_i) = \frac{P(o_i \mid c_j)P(c_j)}{\sum_{k=1}^{K} P(o_i \mid c_k)P(c_k)} \tag{1}$$

This can easily be extended to a series of query images and the output IDs. Some sequence $x^{1:m}$ of m query images, all of person j, is fed to the ReID system. The ReID system produces a corresponding m-dimensional vector of classifications $\overrightarrow{o}$. The probability $P(c_j \mid \overrightarrow{o})$ is:

$$P(c_j \mid \overrightarrow{o}) = \prod_{k=1}^{m} P(c_j^k \mid o^k) \tag{2}$$

with $\overrightarrow{o} = (o^1, \ldots, o^m)$, and $\overrightarrow{c_j} = (c_j^1, \ldots, c_j^m)$. We could either substitute this into Eq. 1 or apply it recursively, i.e. applying the Bayesian update with each element of the product in turn.

3 Votes as observed output



Suppose that, instead of taking the winner of a voting process as the observed output we take the distribution of votes over the categories. This means that we have to take the first n votes only, and n is likely to be far smaller than K the number of categories. The raw vote vector is:

$$\vec{v} = \langle v_1, \ldots, v_K \rangle \tag{3}$$

This will contain mostly zeros. Nevertheless we can normalise the votes. This gives us:

$$\vec{p} = \langle p_1, \ldots, p_K \rangle \tag{4}$$

Then, we can calculate the likelihood of drawing these normalised votes given some true class j. In this case we may estimate the observation model using a kernel estimator where the distance metric underpinning the kernels ϕ is the Hellinger distance.

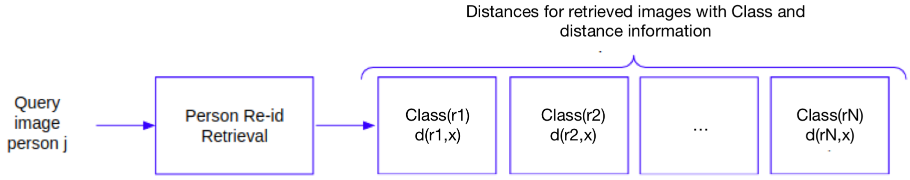
$$P(\vec{p} \mid c_j) \propto \sum_i \phi(\vec{p}, \vec{e}(i)) \tag{5}$$

K is Gaussian kernal defined with respect to the Hellinger distance for a discrete distribution.

$$H(\vec{p}, \vec{e}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{K} (\sqrt{p_k}) - \sqrt{e_k})^2} \tag{6}$$

So, to obtain the KDE estimator, we use a validation set to estimate the density over vote distributions. Each normalised vote distribution constitutes a kernel centre on the K-dimensional simplex. The kernels are Gaussians defined using Hellinger distance. This gives us a kernel density representation of the likelihood of each normalised vote distribution for each class. We then apply Bayes' rule as before.

4 Vector of distances as observed output



Let us now suppose that instead of predicted class, or vote distribution, the ReID system simply outputs the distances between the query and the retrieved images. The ReID system is then as in the figure above. $d(o_k, c_j)$ is the distance value between the image of person j and the retrieved image of person i. We obtain a distance for every retrieved image $k = 1\colon K$.

From a validation set we can learn kernel density estimates of the distributions that give us the likelihoods of the distances conditional on classes. In general, the distance matrix $\mathbf{D}$ across the training or validation set is symmetric. The upper triangle of $\mathbf{D}$ is divided into blocks. These are upper triangular for the distances between images of a person i and other images of that person i, i.e. $d(a_i, b_i) = \mathbf{D}_{\mathbf{a_i},\mathbf{b_i}}$, where a_i and y_i are two different images of person i, numbered a and b respectively. They are rectangular $|C_i| \times |C_j|$ for classes i and j where $i \neq j$.

If we want to estimate the likelihoods we simply use:

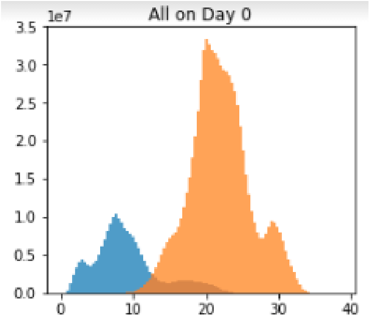


Figure 1: Histogram for queries over distance. Blue is for matched IDs ($j = i$) and orange is for unmatched ($j \neq i$).

$$P(d \mid C_i, C_j) \propto \sum_a \sum_b \phi(d, d(a_i, b_j)) \tag{7}$$

This distribution of distances can be conditioned on the additional event that the true class and output class are identical ($i = j$). It may well be the case that making a sufficiently good estimate of these class conditional distributions from a validation set for data to be trained and used on a single day is too problematic. Of course, we can estimate more generic distributions, by marginalising over all persons within a day, or by additionally marginalising over distance data several days.

$$P(d \mid i \neq j) = \sum_{i=1}^{K} \sum_{j=i+1}^{K} P(d \mid C_i, C_j) \tag{8}$$

$$P(d \mid i = j) = \sum_i P(d \mid C_i, C_i) \tag{9}$$

An example histogram for distances, conditioned on matching and non-matching IDs–from which we can obtain an estimated distribution–is shown in Figure 1). This is from a single day, across all classes. If we gather a validation set over more than one day, and add a conditioning variable for the day, w, we can marginalise this out.

$$P(d \mid i \neq j, \forall w) = \sum_w P(d \mid i \neq j, w) \tag{10}$$

Given these, then the likelihood of the vector of distances associated with the N retrieved images can be written as follows:

$$P(R \mid c_i) = \prod_{r_a \in R_i^c} P(d(x, r_a) \mid i = j) \prod_{r_b \in R_i} P(d(x, r_b) \mid i \neq j) \tag{11}$$

Using this the posterior would be:

$$P(c_i \mid R) \propto P(R \mid c_i) P(c_i) \tag{12}$$

Given particular time interval $0 : t$, where a series of query images of person j is fed to the system, $R_{0:t}$ is the series of retrievals. The posterior is then simply:

$$P(c_i \mid R_{0:t}) = \prod_{k=0}^{t} P(c_i \mid R_k) \tag{13}$$

5 Discussion

There are some papers on Bayesian methods for dealing with ranked data and for information retrieval. The former are typically concerned with inference the distribution of some underlying continuous variable (here d) conditioned on class membership. This value then influences some ranking of classes in a random sample. This has been applied to inferring d from a set of rankings R, e.g. in the TrueSkill system [1]. However, our problem is inferring the posterior over which class is most likely to have won overall, given such a ranking, derived from a distance measure. In which case it seems that the ranking variables are mere intermediates, and can be ignored.

For information retrieval, there are Bayesian methods, but these are concerned with estimating the posterior over whether a document is relevant given a set of search terms that may appear in that document. Thus, while the topic is superficially related, it is different underneath.

Each system proposed above essentially relies on the amount validation data that is possible to gather, and how generic vs specific the observation model is. The more specific the observation model (i.e. the more specific its conditioning variables are) the better a job it will do of making inferences about the posterior. But the smaller the amount of validation data for training the observation model the worse the quality. Thus, there is a temptation to make weaker inferences from more generic observation models. These will thus converge more slowly (the posterior will have a higher variance), but the posterior will be more reliable (it will be closer to the true posterior if we had a perfect observation model).

References

[1] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill^{TM} a Bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576, 2007.

[2] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. 2017.