k Nearest Neighbor using Ensemble Clustering

Loai AbedAllah and Ilan Shimshoni

[1] Department of Mathematics, University of Haifa, Israel
Department of Mathematics, The College of Saknin, Israel
[2] Department of Information Systems, University of Haifa, Israel
loai1984@gmail.com & ishimshoni@mis.haifa.ac.il

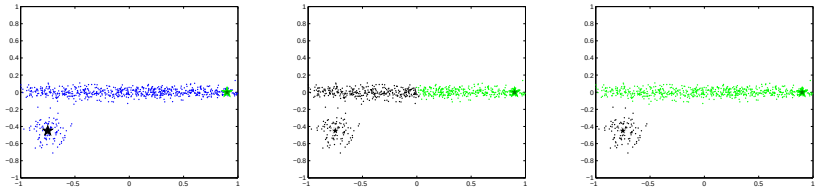
Abstract. The performance of the k Nearest Neighbor (kNN) algorithm depends critically on its being given a good metric over the input space. One of its main drawbacks is that kNN uses only the geometric distance to measure the similarity and the dissimilarity between the objects without using any statistical regularities in the data, which could help convey the inter-class distance. We found that objects belonging to the same cluster usually share some common traits even though their geometric distance might be large. We therefore decided to define a metric based on clustering. As there is no optimal clustering algorithm with optimal parameter values, several clustering runs are performed yielding an ensemble of clustering (EC) results. The distance between points is defined by how many times the objects were not clustered together. This distance is then used within the framework of the kNN algorithm (kNN-EC). Moreover, objects which were always clustered together in the same clusters are defined as members of an equivalence class. As a result the algorithm now runs on equivalence classes instead of single objects. In our experiments the number of equivalence classes is usually one tenth to one fourth of the number of objects. This equivalence class representation is in effect a smart data reduction technique which can have a wide range of applications. It is complementary to other data reduction methods such as feature selection and methods for dimensionality reduction such as for example PCA. We compared kNN-EC to the original kNN on standard datasets from different fields, and for segmenting a real color image to foreground and background. Our experiments show that kNN-EC performs better than or comparable to the original kNN over the standard datasets and is superior for the color image segmentation.

Keywords: Clustering, Classification, Ensemble Clustering, Unsupervised Distance Metric Learning.

1 Introduction

The performance of many learning and data mining algorithms depend critically on there being given a good metric over the input space. Learning a "good" metric from examples may therefore be the key of a successful application of these algorithms. For instance, many researchers have demonstrated that k-nearest neighbor (kNN) [8] classification can be significantly improved by learning a

distance metric from labeled examples [5,10,14,16]. However, like any classifier kNN has some drawbacks. One of its main drawbacks is that most implementations of kNN use only the geometric distance to measure the similarity and the dissimilarity between the objects without using any statistical regularities in the data. Thus, it does not always convey the inter-class distance. The following example illustrates this situation. Given the dataset in Figure 1(a) with two labeled points. When the classifier uses the Euclidean distance, it works "poorly" and many points belonging to the green class were classified to be black (b).



(a) (b) (c)

Fig. 1. Euclidean distance does not reflect the actual similarity.

To overcome this problem we turned to clustering for defining a better metric. As there is no optimal clustering algorithm with optimal parameter values, several clustering runs are performed yielding an ensemble of clustering results. The distance between points is defined by how many times the points were not clustered together. This distance is then used within the framework of the kNN algorithm (kNN-EC). Returning to the previous example in Figure 1(a), we can see that it worked very well (c). Moreover, points that are always clustered together in the same cluster (distance= 0) are defined as members of an equivalence class. As a result, the algorithm now runs on equivalence classes instead of single points. In our experiments the number of equivalence classes is usually less than one tenth to one fourth of the number of points. This equivalence class representation is in effect a novel data reduction technique which can have a wide range of applications. It is complementary to other data reduction methods such as feature selection and methods for dimensionality reduction such as the well known Principal Component Analysis (PCA).

This paper is organized as follows: Related work on distance metric learning is discussed in Section 2. The distance metric using ensemble clustering is described in Section 3. Section 4 describes the ensemble clustering method using the mean shift and the k-means clustering algorithms. Experimental results are presented in Section 5. Finally, our conclusions are presented in Section 6.

2 Related work

A large body of work has been presented on the topic of distance metrics learning, and we will just briefly mention some examples. Most of the work in distance met-

ric learning can be organized into the following two categories: Supervised/semi-supervised distance metric learning and unsupervised distance metric learning.

Most supervised/semi-supervised distance metric learning attempt to learn metrics that keep data points within the same classes close, while separating data points from different classes [5,16]. Goldberger et. al [14] provided a distance metric learning method to improve the classification of the kNN algorithm. They use a gradient decent function to reduce the chance of error under the stochastic neighborhood assignments. Domeniconi et. al [10] proposed to use a locally adaptive distance metric for kNN classification such as the decision boundaries of SVMs. Shalev-Shwartz et. al [22] considered an online method for learning a Mahalanobis distance metric. The goal of their method is to minimize the distance between all similarity labeled inputs by defining margins and inducing hinge loss functions. Recently, a similar method was presented by Weinberger et. al [24] which uses only the similarly labeled inputs that are specified as neighbors in order to minimize the distance.

The unsupervised distance metric learning takes an input dataset, and finds an embedding of it in some space. Many unsupervised distance metric learning algorithms have been proposed. Gonzales and Woods [15] provided the well known PCA which finds the subspace which best maintains the variance of the input data. Tenenbaum et. al [23] proposed a method called ISOMAP which finds the subspace which best maintains the geodesic inter-point distances. Saul et. al [21] provided a locally linear embedding (LLE) method to establish the mapping relationship between the observed data and the corresponding low dimensional data. Belikin et. al [1] presented an algorithm called the Laplacian Eigenamp to focus on the maintenance of local neighbor structure.

Our method falls into the category of the unsupervised distance metric learning. Given an unlabeled dataset, a clustering procedure is applied several times with different parameter values. The distance between points is defined as a function of the number of times the points belonged to different clusters in the different runs.

A clustering based learning method was proposed in Derbeko, El-Yaniv, and Meir [9]. There, several clustering algorithms are run to generate several (unsupervised) models. The learner then utilizes the labeled data to guess labels for entire clusters (under the assumption that all points in the same cluster have the same label). In this way the algorithm forms a number of hypotheses. The one that minimizes the PAC-Bayesian bound is chosen and used as the classifier. They assume that at least one of the clustering runs produces a good classifier and that their algorithm finds it.

Our work is different from these techniques in several ways especially on the assumptions that they made. Unlike other techniques, we only assume that the equivalence classes, which were built by running the clustering algorithm several times, are quite pure. We also did not assume that at least one of the clustering runs produces a good classifier. Rather that the true classifier can be approximated quite well by a set of equivalence classes (i.e the points which always belong to the same clusters in the different clustering iterations will define

an equivalence class) instead of single points and the distance metric defined between these equivalence classes.

3 Distance Metric Learning using Ensemble Clustering

Consider the following paradigm. Let X be the *unlabeled data* - a set of unlabeled instances where each x_i is a vector in some space χ. Instances are assumed to be i.i.d. distributed according to some unknown fixed distribution ρ. Each instance x_i has a label $w_i \in \mathcal{W}$ (where in our case $\mathcal{W} = \{0,1\}$) distributed according to some unknown conditional distribution $P(w|x)$. Let $D = \{\langle x_i, f(x_i)\rangle : x_i \in X, i = 1,...,N_D\}$ be the *training data*—a set of labeled examples already known.

The Euclidean distance does not always reflect the actual similarity or dissimilarity of the objects to be classified. We found that on the other hand points belonging to the same cluster usually share some common traits even though their geometric distance might be large.

The main problem with such an approach is that there is no known method to choose the best clustering. There have been several attempts to try to select the optimal parameters values of the clustering algorithms in supervised and unsupervised manners mainly within the range image and color image domain, but a general solution to this problem has not been found [19,3,25]. We therefore decided to run different clustering algorithms several times with different parameter values. The result of all these runs yields a cluster ensemble [11].

The clustering results are stored in a matrix denoted the *clusters matrix* $C \in Mat_{N\times K}$, where K is the number of times the clustering algorithms were run. The ith row consists of the cluster identities of the ith point in the different runs. This results in a new instance space $\chi_{cl} = \mathbb{Z}^K$ which contains the rows of the *clusters matrix*. Let X_{cl} be an *unlabeled training set*, a set of objects drawn randomly from χ_{cl} according to distribution ρ. Let $D_{cl} = \{\langle x_i, f(x_i)\rangle : x_i \in X, i = 1,...,N_D\}$ be the *training data*—a set of labeled examples from X_{cl}.

The goal now is to adapt the kNN classifier to work with a distance function based on the new instance space. The new distance between points from this space should be defined in such a way as to reflect our intuitive notion on proximity among the corresponding points.

Given two points $x, y \in \chi_{cl}$ we define a new distance function d_{cl} as:

$$d_{cl}(x,y) = \sum_{i=1}^{K} dis(x_i, y_i), \tag{1}$$

where $dis(x_i, y_i) = \begin{cases} 1 & x_i \neq y_i \\ 0 & x_i = y_i \end{cases}$ be the metric of a single feature. This metric is known as the *Hamming distance*. Over this metric we define the following equivalence relation.

Let E be a binary relation on χ_{cl}, where E defined as

$$\forall x, y \in \chi_{cl}, (x, y) \in E \Leftrightarrow d_{cl}(x, y) = 0.$$

The relation E is an equivalence relation on χ_{cl}. By this relation, points which always belong to the same clusters in the different clustering iterations will define an equivalence class $[\cdot]_E$. Thus, all the equivalent points will be represented by a single point in the quotient set and we can work with X/E yielding $C' \in Mat_{M \times K}$, where $M = |X/E|$. On the other hand, points which always belong to different clusters in all the clustering iterations will be infinitely distant (i.e. $d_{cl}(x, y) = \infty$ if and only if x and y always belong to different clusters in all the clustering iterations). Thus, x is a neighbor of y if and only if $d_{cl}(x, y) < \infty$. The set of the neighbors of x will be defined as: $\mathfrak{N}_x = \{y | d_{cl}(x, y) < \infty\}$. For each $x \in X$ $\mathfrak{N}_x \neq \emptyset$, since by using the reflexive property of E, we get $d_{cl}(x, x) = 0 < \infty$, thus, $x \in \mathfrak{N}_x$.

This new metric is used in the kNN classifier instead of the Euclidean distance. In this setting all the unlabeled points in X_{cl} will be labeled according to a given training dataset D_{cl}. Experiments using this method are presented in Section 5.

The main presumption made by the algorithm is that *equivalent points have the same label* but this assumption does not always hold in practice. To overcome this hurdle several possible options exist. One possibility is that for each equivalence class $x_{cl} \in D_{cl}$ several points from its equivalence class are labeled and x_{cl} will then be labeled according to the majority voting. Another option is to label x_{cl} according to its center point. Thus, with high probability a point will be selected from the majority class of the equivalence class. It is also possible to run the clustering algorithms more times, increasing the number of the equivalence classes yielding smaller but hopefully purer equivalence classes.

4 Ensemble clustering using mean shift and k means algorithms

As mentioned above the main problem with an approach based on clustering is that there is no known method to choose the best clustering. It is unknown how many clusters should be, their shapes, which clustering algorithm is best, and which parameter values should be used? We therefore decided to run different clustering algorithms several times with different parameter values.

Our algorithm however is general and any good clustering algorithm could be used. We decided to work with the well known k-means algorithm [18] and the mean shift clustering algorithm [13,7] in order to build the clusters matrix.

For completeness we will now give a short overview of the mean shift algorithm. Mean shift is a non-parametric iterative clustering algorithm. The fact that mean shift does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters, makes it ideal for handling clusters of arbitrary shape and number. It is also an iterative technique, but instead of

the means, it estimates the modes of the multivariate distribution underlying the feature space. The number of clusters is obtained automatically by finding the centers of the densest regions in the space (the modes). The density is evaluated using kernel density estimation which is a non-parametric way to estimate the density function of a random variable. This is also called the Parzen window technique. Given a kernel K, bandwidth parameter h, which is a smoothing parameter of the estimated density function, the kernel density estimator for a given set of d-dimensional points is:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{2}$$

For each data point, a gradient ascent process is performed on the local estimated density until convergence. The convergence points represent the modes of the density function. All points associated with the same convergence point belong to the same cluster.

We worked with the two mean shift algorithm types; the simple, and the adaptive (more details in [13,7]). The simple mean shift works with a fixed bandwidth h. We chose 80 different values of h with fixed intervals from 0.1 to 0.9 of the space size. The adaptive mean shift algorithm is given the number of neighbors k as a parameter and the bandwidth is determined for each point in the data as the distance to its k'th neighbor. We chose 30 different values of k with fixed intervals between 1% to 30% of N (for more details see Section 5).

Some clustering algorithms work with continuous parameters, like the mean shift algorithm described above, or with continuous weights over the features, like the EDISON program which will be discussed in Section 5.2. In these cases the differences between two consecutive iterations might be small. There are two possibilities to deal with these similar clusterings: The first one is to eliminate the similar clustering results or simply take all of them. We preferred the second one because if a set of samples were together in several clustering runs it means that they might have some common features. So if we eliminate them we stand to loose this information. However, it is not efficient to preserve similar clustering runs. Therefore, we decided to join them, as a result the dimensionality of the data is reduced. We use the Rand index [20] which is a measure of similarity between two data clusterings. Let $C1, C2$ be two clustering iterations, then the measure between them is:

$$R(C1, C2) = \frac{\alpha + \beta}{\alpha + \beta + \gamma + \delta} = \frac{\alpha + \beta}{\binom{n}{2}}, \tag{3}$$

where α describes the number of pairs of elements in the instance space that are in the same set (i.e cluster) in $C1$ and in the same set in $C2$, β describes the number of pairs of elements in the instance space that are in the different set in $C1$ and in the different set in $C2$, γ describes the number of pairs of elements in the instance space that are in the same set in $C1$ and in the different set in $C2$ and δ describes the number of pairs of elements in the instance space that are in the different set in $C1$ and in the same set in $C2$.

Similar clusterings are represented by a single column, weighted by the number of clustering it represents. Accordingly, the metric function has become:

$$d_{cln}(x,y) = \sum_{i=1}^{q} n_i dis(x_i, y_i), \tag{4}$$

where $x, y \in \chi_{cln}$ are two points in the new weighted space, q is the dimension of χ_{cln}, and n_i is the weight of each representative column.

The advantage of this method is that it maintains the relation between the samples according to the clustering results, while maintaining a relatively small dimension of the clustering matrix.

This method worked quite well for mean shift clustering as the bandwidth acts as a smoothing parameter for the density estimation. However, for k-means the differences between consecutive runs of the algorithm were significant and thus columns could not be joined.

5 Experiments

To validate the efficiency of kNN-EC we conducted a series of experiments using standard datasets from different fields. An additional experiment was conduct on a color image, where the mission is to classify each pixel as a foreground or background pixel. We compare the performance of kNN-EC to that of kNN on the same datasets. Both algorithms were implemented in Matlab.

5.1 Datasets

In order to evaluate the performance of kNN-EC we ran experiments on four datasets; the image segmentation dataset (UCI Machine Learning Repository [12]), the breast cancer dataset (LIBSVM library [4]), Leo Breiman's ringnorm [2], and a real color image. The image segmentation dataset contains 2310 instances, which are divided into 7 classes. Since we choose to work with a binary kNN, the classes were joined to create two class labels (as was done in [17]) one corresponding to BRICKFACE, SKY and FOLIAGE and the other corresponding to CEMENT, WINDOW, PATH and GRASS. The breast cancer dataset contains 683 instances, which are divided into two class labels, such that 444 points are from the first class and the rest are from the second. Leo Breiman's ring norm dataset contains 7400 instances, two-class classification problem. Each class is drawn from a multivariate normal distribution. The last dataset is a color image. More details on this experiment will be described in Section 5.2. All these datasets were labeled, but this knowledge was used only to evaluate the quality rate of the resulting classifier. In all experiments the algorithm assumes that these datasets are unlabeled.

The mean shift algorithm was run with the k or h values, described above. For the breast cancer and ring norm datasets the mean shift algorithm did not yield good clustering (i.e one cluster or the same clustering for all runs). So

we use the k-means algorithm for these two datasets. For the breast cancer the k-means algorithm was run with $k = 3..15$ and for the ring norm dataset the k-means algorithm was run with $k = 3..30$. The results are stored in the clusters matrix C. The equivalence relation E was employed to build the equivalence matrix C'. As can be seen in Table 1, the new space is usually smaller than the original space without the equivalence classes. The ratio between the sizes of the two spaces is given in the fourth column. Our algorithm assumes that points belonging to the same equivalence class have the same label. However, as can be seen from the last column the equivalence classes are not perfectly pure.

Table 1. Numerical Datasets properties

Dataset	Dataset size	Cluster matrix size	Equivalence matrix size	Ratio %	Purity
Image Segmentation	2310×19	2310×38	548×38	24%	97%
Breast Cancer	683×8	683×13	160×13	23%	98%
Ring Norms	7400×20	7400×28	7400×28	100%	100%

At the first stage of each algorithm a training set of size 20% to 40% of the dataset is randomly drawn and labeled. For each training dataset the algorithms run with different numbers of neighbor values (i.e $k = 3, 5, 7$). For each k the quality was evaluated by the ability of the classifier to label the rest of the unlabeled points. The results are averaged over 10 different runs on each dataset. A resulting curve was constructed for each dataset which evaluated how well the algorithm performed.

Results As can be seen from Figure 2, the kNN-EC performs better than or comparable to the kNN with the Euclidean distance. The learning curves, which describe the accuracy for each classifier by computing the ratio of the correct classified instances to the whole unlabeled data, of the image segmentation dataset and that of the Breast Cancer datasets show that kNN-EC is comparable to the kNN while glancing at the learning curves of the ring norm dataset depict the superiority of kNN-EC. As Figure 2 shows the quality of kNN-EC is about 85% while the kNN quality is about 65%. Moreover we compute the runtime of the two algorithms when the training dataset includes 30% of the points, and $k = 5$. The runtime results are shown in Table 2.

Table 2. Runtime of the algorithms in seconds

Dataset	kNN	kNN-EC
Image Segmentation	0.45	0.15
Breast Cancer	0.15	0.01
Ring Norms	2.4	3.9

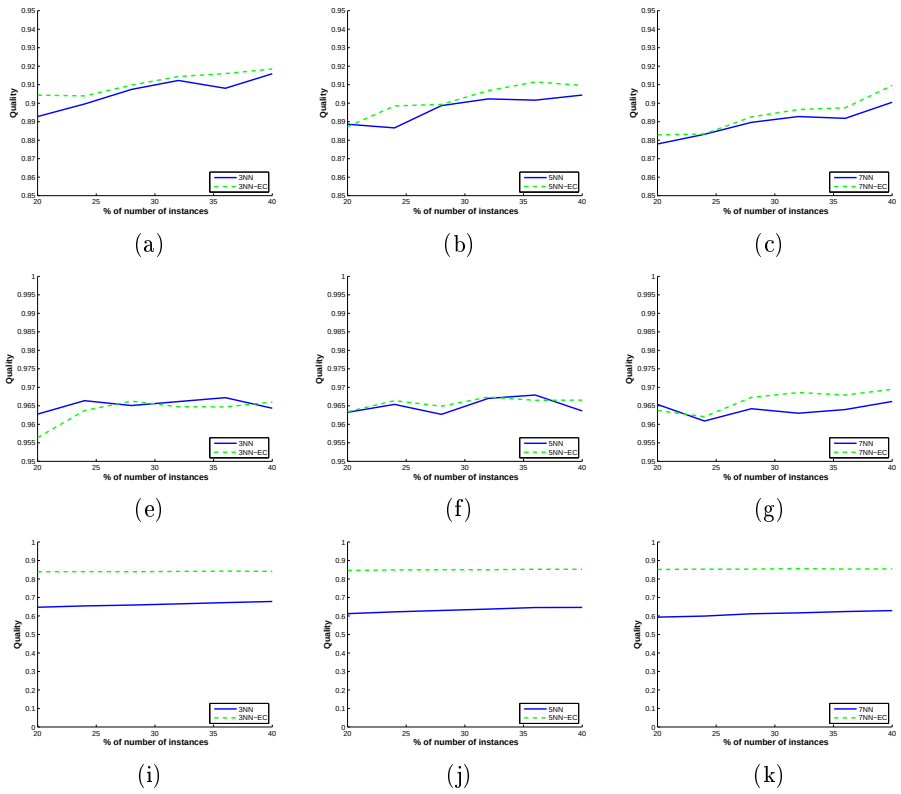
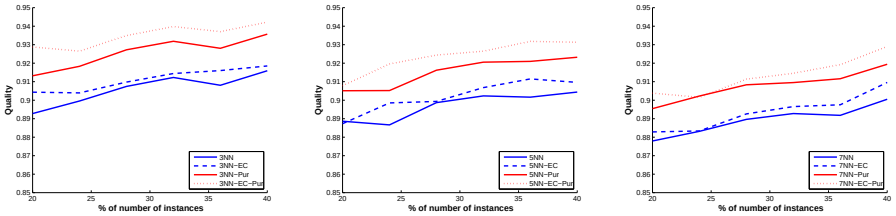


Fig. 2. Results of kNN and kNN-EC for the three datasets. The first row shows the learning curves of the Image Segmentation dataset, the second shows Breast Cancer dataset, and the third one shows the Ring Norm dataset. The columns show the learning curves for the different k's values.

The effect of the purity of the equivalence classes As shown in the previous subsection, the performance of the kNN-EC is not always superior. In this section we carried out an experiment in order to determine how much our algorithm depends on the purity of the equivalence classes. In the original experiments the equivalence classes were not pure, for instance the purity of the image segmentation dataset was 97%. In this experiment the classes were changed until the equivalence classes were pure (i.e 100%). As shown in this Figure 3 there is a linear trade off between the quality and the purity of the equivalence classes. The quality increased by about 3% while the purity increased from 97% to 100%.

The effect of number of the clustering iterations on the performance of kNN-EC algorithm Another experiment was performed to determine how much our algorithm depends on the number of clustering iterations. In this ex-



(a) (b) (c)

Fig. 3. Results of kNN and kNN-EC for the image segmentation dataset with the different k's values.

periment we evaluate the performance of the 5NN-EC classifier on the ring norm dataset given 20% of the dataset as a training set. We run the k-means algorithm on three different ranges $k = 3..10$, $k = 3..20$ and $k = 3..30$ as shown in Table 3. As the number of the clustering runs increases, the purity of the equivalence classes increases, and the number of the equivalence classes increases dramatically. (When the clustering runs increased from 8 to 18 runs the purity increased from 93 to 99.8 and the equivalence classes increased from 3351 equivalence classes to 7171 equivalence classes). However, the performance of the algorithm preserves it stability (i.e the quality increased from 81% to 83%). This occurred because the distance function metric based on ensemble clustering is stable, and if for example an equivalence class was partitioned then the distance between the instances which were equivalent will be 1 instead of zero. Thus with hight probability they will still be classified to the same class.

Table 3. The effect of the number of the clustering iterations

k-Means	Equivalence matrix size	Purity %	Quality %
$k = 3..10$	3351×8	93	81
$k = 3..20$	7171×18	99.8	83
$k = 3..30$	7400×28	100	85

5.2 Experiments with Images

In a final set of experiments we tested our algorithm using a real color image. We use images for two reasons, first images provide large complex datasets and second that the results obtained by applying classification algorithms on images can be easily viewed and evaluated. This image contains three birds (shown in Figure 4(a)). It was manually segmented into two classes, the foreground (birds) and the background yielding the ground truth (as shown in Figure 4(b)). The reader can appreciate that segmenting these images using a color based

(a) (b) (c) (d)

Fig. 4. (a) The original image with three birds. (b) The classified image (the goal). (c),(d) The output of the EDISON system

segmentation algorithm into foreground and background images will not be an easy task.

We chose to work with the Edge Detection and Image Segmentation (EDISON) System. This program implements the mean shift image segmentation algorithm described in [6,7]. Each pixel in the image is represented by its two image coordinates and RGB color values yielding a 5D dataset. The user is asked to provide the algorithm with values for two bandwidths, one for the *spatial* domain h_s (the image coordinates) and the other for the *range* domain h_r (the RGB values). The output of this program is a clustered image. Each cluster was assigned a color, (i.e points in the same cluster have the same color). Figure 4 (c,d) shows some of these clustering results.

In our experiments we used the following values for the two bandwidths $h_s = \{5, 10, 20, 30\}$ and $h_r = \{10, 15, 20, 25, 30, 35\}$ yielding 24 clustered images. Results for which nearly the whole image belonged to a single cluster were automatically discarded. It is important to note that the original kNN classifier has to choose values for these bandwidths (or actually their ratio) in order to define the distance metric between points. As optimal values for these bandwidths are not available, it is not clear how this method can be compared to kNN-EC. In the experiments we therefore ran them using all 24 bandwidth pair values.

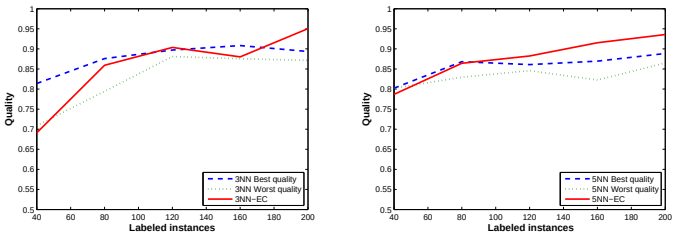
For each image the EDISON algorithm was run with the h_r and h_s values, described above, to build the clusters matrix C. We then joined clusterings with small Rand index measures and worked with the weighted C_w matrix. The equivalence relation E is employed to build the equivalence matrix C'. Table 4 summarizes the information about the three birds image.

In this table we can see that the new space is about 10 times smaller than the original space. As the complexity of the algorithm is $O(N^2)$, the running time of kNN-EC is two orders of magnitude smaller than the running time of kNN.

Table 4. Image properties

Dataset	Picture size	Cluster matrix size	Equivalence matrix size	Ratio %	Purity	Fg pixels
Three Birds	207×352	72864×24	8233×18	11%	100%	14407

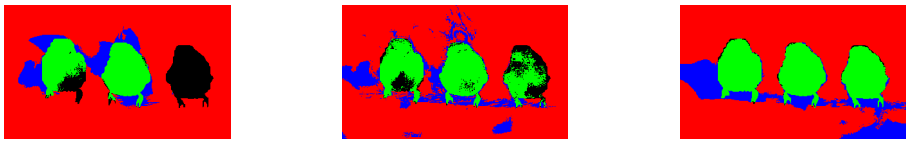
The two classifiers were evaluated on several training sets of size 40 pixels to 200 pixels (less than 0.3% of the data), with different numbers of neighbors values (i.e $k = 3, 5$). As the optimal bandwidth parameters can not be found automatically in the kNN algorithm, then in the resulting curves we compared the kNN-EC with the best case and the worst case of the kNN for each training dataset which evaluated how well the algorithms perform (as shown in Figure 5).



(a) (b)

Fig. 5. Results of kNN and kNN-EC for the three birds image dataset with the different k's values.

The experimental results shown in Figure 5 show that the kNN-EC performs better than the kNN with the Euclidean distance. Due to the learning curves presented in the figures below we see that our algorithm is superior, where its quality was around 95% while the best quality of the kNN algorithm was less than 90%. Figure 6 shows the superiority of the kNN-EC. It shows an example of running the kNN-EC and kNN (the best and the worst cases) algorithms with k=5, and with 120 labeled pixels as a training set.



(a) (b) (c)

Fig. 6. Results of 5NN and 5NN-EC for the three birds image dataset for given 120 labeled pixels as a training dataset. (a) The output for the worst case of kNN. (b) The output for the best case of kNN. (c) The output for the kNN-EC. The color of the pixels represents the results of the classifier. Red is background, green is birds, blue is wrong background and black is wrong birds pixels.

6 Conclusions and future work

In this work, we have presented a new unsupervised distance metric learning based on ensemble clustering and use it within the kNN classifier. Each data point is characterized by the identity of the clusters that it belongs to in several clustering runs. The distance metric is defined as the Hamming distance between these clustering results.

This new distance has two important contributions. The first contribution is that this distance is more meaningful than the Euclidean distance between points resulting in a better kNN classifier. The second and more general contribution results from the observation that all points which always belong to the same cluster form an equivalence relation. Thus, the algorithm only has to consider one member of each equivalence class. This reduces the complexity of the algorithm considerably (by at least two orders of magnitude in our case). Our algorithm however is only a private case of a more general concept, the concept of data reduction. This concept is orthogonal to other methods of data reduction such as feature selection or PCA which reduce the size of the representation of the data points but not their number.

References

1. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
2. L. Bias. Variance and arcing classifiers. *Tec. Report 460,Statistics department*, 1996.
3. S. Chabrier, B. Emile, C. Rosenberger, and H. Laurent. Unsupervised performance evaluation of image segmentation. *EURASIP Journal on Applied Signal Processing*, 2006:1–12, 2006.
4. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.
5. S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 26–33, 2005.
6. C. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *Proceedings of International Conference on Pattern Recognition*, pages 150–155, 2002.
7. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.
8. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
9. P. Derbeko, R. El-Yaniv, and R. Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *Journal of Artificial Intelligence Research*, 22(1):117–142, 2004.
10. C. Domeniconi, D. Gunopulos, and J. Peng. Large margin nearest neighbor classifiers. *IEEE Transactions on Neural Networks*, 16(4):899–909, 2005.

11. X.Z. Fern and C.E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, pages 36–43. ACM, 2004.
12. A. Frank and A. Asuncion. UCI machine learning repository, 2010.
13. B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Proceedings of the 9th International Conference on Computer Vision*, pages 456–463, 2003.
14. J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520, 2004.
15. Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.
16. T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, 1996.
17. M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54(2):125–152, 2004.
18. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*, pages 281–297, 1967.
19. J. Min, M. Powell, and K.W. Bowyer. Automated performance evaluation of range image segmentation algorithms. *IEEE Transactions on Systems Man and Cybernetics-Part B-Cybernetics*, 34(1):263–271, 2004.
20. W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.
21. L.K. Saul and S.T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155, 2003.
22. S. Shalev-Shwartz, Y. Singer, and A.Y. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the twenty-first international conference on Machine learning*, pages 94–102. ACM, 2004.
23. J.B. Tenenbaum, V. Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):19–23, 2000.
24. K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.
25. H. Zhang, J.E. Fritts, and S.A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.