An ensemble-clustering-based distance metric and its applications

Loai AbdAllah*

Department of Mathematics,
University of Haifa,
Haifa, 31905, Israel
and
Department of Mathematics and Computer Science,
The College of Sakhnin,
Sakhnin, B.O. 100, ZIP:20173, Israel
E-mail: Loai1984@gmail.com
*Corresponding author

Ilan Shimshoni

Department of Information Systems,
University of Haifa,
Haifa, 31905, Israel
E-mail: ishimshoni@mis.haifa.ac.il

Abstract: A distance metric learned from data reflects the actual similarity between objects better than the geometric distance. So, in this paper, we propose a new distance that is based on clustering. Because objects belonging to the same cluster usually share some common traits even though their geometric distance might be large. Thus, we perform several clustering runs to yield an ensemble of clustering results. The distance is defined by how many times the objects were not clustered together. To evaluate the ability of this new distance to reflect object similarity, we apply it to two types of data mining algorithms, classification (kNN) and selective sampling (LSS). We experimented on standard numerical datasets and on real colour images. Using our distance, the algorithms run on equivalence classes instead of single objects, yielding a considerable speedup. We compared the kNN-EC classifier and LSS-EC algorithm to the original kNN and LSS algorithms.

Keywords: clustering; classification; unsupervised distance metric learning; ensemble clustering.

Reference to this paper should be made as follows: AbdAllah, L. and Shimshoni, I. (2013) 'An ensemble-clustering-based distance metric and its applications', *Int. J. Business Intelligence and Data Mining*, Vol. 8, No. 3, pp.264–287.

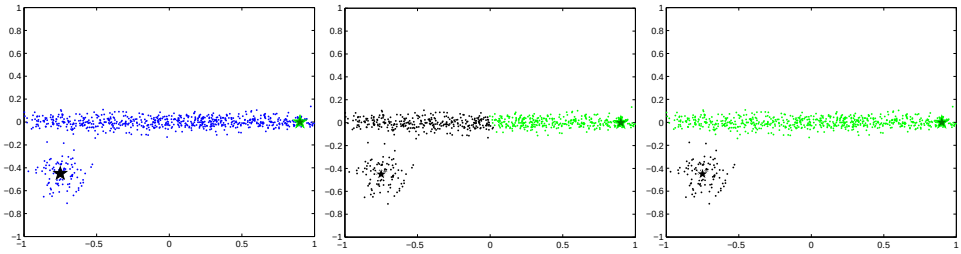
Biographical notes: Loai AbdAllah received his BSc in Mathematics and Management Information Systems from the University of Haifa, and MSc in Mathematics from the University of Haifa, where he is currently working toward the PhD in Mathematics in the University of Haifa. He was a member of the Departments of Mathematics and Computer Science at the College of Sakhnin from 2011. His current research interest is in data mining.

Ilan Shimshoni received his BSc in Mathematics from the Hebrew University in Jerusalem, MSc in Computer Science from the Weizmann Institute of Science, and PhD in Computer Science from the University of Illinois at Urbana Champaign (UIUC). He was a Post-Doctorate Fellow at the Faculty of Computer Science at the Technion from 1995–998, and was a member of the Faculty of Industrial Engineering and Management from 1998–2005. He joined the Department of Information Systems (IS) at Haifa University in 2005.

1 Introduction

The performance of many learning and data mining algorithms depends critically on their being given a good metric over the input space. Learning a 'good' metric from examples may therefore be the key to the successful application of these algorithms. Many data mining algorithms use only the geometric distance to measure object similarity or dissimilarity without using any statistical regularities in the data. But this metric does not always reflect the actual similarity between the objects. The following example illustrates this situation. Consider the dataset in Figure 1(a), with two labelled points. When the nearest neighbour (NN) classifier (Cover and Hart, 1967) uses the Euclidean distance, it works poorly and many points belonging to the green class are incorrectly classified as black Figure 1(b).

Figure 1 Euclidean distance does not reflect actual object similarity (see online version for colours)



(a) (b) (c)

To overcome this problem, we need to define a better metric. To this end, we turned to clustering. As there is no optimal clustering algorithm with optimal parameter values, we performed several clustering runs, which yielded 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 *k*NN algorithm (*k*NN-EC). Figure 1(c) shows that this method worked very well.

Using the new distance function, points that are always clustered together in the same cluster (distance= 0) are defined as members of an equivalence class. As a result, the algorithms now run on equivalence classes instead of single points. In our experiments, the number of equivalence classes is usually between 2% to 24% of the number of points. This equivalence class representation is in effect a novel data reduction technique with a wide range of possible applications. It is complementary to other data reduction methods

such as feature selection and to dimensionality reduction methods such as the well-known principal component analysis (PCA) technique.

Our metric is general and can be used with any approach which uses a distance metric. But because its performance depends strongly on the chosen distance measure, we opted for the k nearest neighbour classifier to evaluate its ability to accurately reflect object similarity. As many researchers have demonstrated, kNN (Cover and Hart, 1967) classification can be notably improved by learning a distance metric from labelled examples (Chopra et al., 2005; Domeniconi et al., 2005; Goldberger et al., 2004; Hastie and Tibshirani, 1996).

In the second part of the paper, we evaluate how this new metric function improves the selection of the most informative samples to be labelled and added to the training dataset. This approach, known as *selective sampling*, (SS) is one of the common active learning approaches. We decided to work with the *selective sampling for nearest neighbour classifier* [lookahead selective sampling (LSS)] (Lindenbaum et al., 2004), because it is based on kNN. At each iteration of the algorithm, all of the unlabelled examples are tested and the point which yields the highest expected utility is chosen. A short version of kNN-EC is given in Lindenbaum et al. (2004).

The paper is organised as follows. Related work on distance metric learning and selective sampling 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. The lookahead algorithm for selective sampling for nearest neighbour classifiers using the suggested metric is presented in Section 5.

Experimental results on numerical datasets and real colour images are presented in Sections 6 and 7, respectively. Finally, our conclusions are presented in Section 8.

2 Related work

In our work we present a new distance metric and, within the framework of kNN, apply it to selective sampling. We will therefore now review related work on distance metric learning and on selective sampling algorithms.

2.1 Distance metric learning

A large body of work has been published on the topic of distance metric learning, and we will briefly mention a few examples. Most of the work can be organised into the following two categories: supervised/semi-supervised distance metric learning and unsupervised distance metric learning.

Most supervised/semi-supervised methods attempt to learn metrics that keep data points within the same classes close, while separating data points from different classes (Chopra et al., 2005; Hastie and Tibshirani, 1996). Goldberger et al. (2004) provided a distance metric learning method to improve the classification of the kNN algorithm. They use a gradient descent function to reduce the chance of error under the stochastic neighbourhood assignments. Domeniconi et al. (2005) proposed to use a locally adaptive distance metric for kNN classification, such as SVM decision boundaries. Shalev-Shwartz et al. (2004) considered an online method for learning a Mahalanobis distance metric. The goal of their method is to minimise the distance between all

similarly labelled inputs by defining margins and inducing hinge loss functions. Recently, Weinberger and Saul (2006) presented a similar method, which uses only the similarly labelled inputs that are specified as neighbours in order to minimise the distance.

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. Gonzalez and Woods (2001) provided the well-known PCA technique, which finds the subspace that best maintains the variance of the input data. Tenenbaum et al. (2000) proposed a method called ISOMAP, which finds the subspace that best maintains the geodesic inter-point distances. Saul and Roweis (2003) provided a locally linear embedding (LLE) method to establish the mapping relationship between the observed data and the corresponding low dimensional data. Belkin and Niyogi (2003) presented an algorithm called the Laplacian eigenmap to focus on the maintenance of local neighbour structure.

Our method falls into the category of unsupervised distance metric learning. Given an unlabelled 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.

This problem of combining multiple clusterings of a set of objects without accessing the original features is called *cluster ensemble*. Combination of clusterings is a more challenging task than combination of supervised classifications. Strehl and Ghosh (2002) and Topchy et al. (2003) addressed this issue by formulating consensus functions that avoid an explicit solution to the correspondence problem. Recent studies have demonstrated that consensus clustering can be found using graph-based, statistical or information-theoretic methods without explicitly solving the label correspondence problem as mentioned in Topchy et al. (2005). Other empirical consensus functions were also considered in Dudoit and Fridlyand (2003), Fischer and Buhmann (2003a) and Fern and Brodley (2003).

A clustering-based learning method was proposed in Derbeko et al. (2004). There, several clustering algorithms are run to generate several (unsupervised) models. The learner then utilises the labelled 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 minimises the PAC-Bayesian bound is chosen and used as the classifier. The authors assume that at least one of the clustering runs produces a good classifier and that their algorithm finds it.

Our technique differs from Derbeko et al.'s in several ways, primarily with regard to the assumptions made. We assume only that the equivalence classes, which were built by running the clustering algorithm several times, are quite pure. Moreover, we do not assume that at least one of the clustering runs produces a good classifier but rather that the true classifier can be approximated quite well by a set of equivalence classes (in other words, 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).

2.2 Selective sampling

Previous work on the problem of selecting a sample of relevant instances from a set of unlabelled data falls under the paradigm of active learning and, more specifically, *selective sampling* (Cesa-Bianchi et al., 1997; Dagan and Engelson, 1995; Lewis and

Catlett, 1994; Lindenbaum et al., 2004), which is more common in practice. Here it is assumed that a set of unlabelled examples is available. In this approach the learner selects an unlabelled example from the given set and asks the teacher to label it.

Much work has been done in selective sampling of examples related mainly to training classifiers: for neural networks (Davis and Hwang, 1992; Cohn et al., 1994), for the C4.5 rule-induction algorithm (Lewis and Catlett, 1994), and for hidden Markov models (Dagan and Engelson, 1995).

The most simple sampling technique is random sampling, where we select a group of instances from an instance space. Another option is to select the point with the largest uncertainty (Lewis and Gale, 1994).

Lindenbaum et al. (2004) claimed that the problem of selective sampling is similar to the problem of cost-sensitive learning (Tan and Schlimmer, 1990; Turney, 1995). They proposed the *lookahead algorithm for selective sampling* (LSS) for the nearest neighbour classifier. The main goal of their algorithm is to develop a selective sampling methodology for nearest-neighbour (NN) classification learning algorithms. Our method is based on this algorithm. We will therefore give a short overview of it in Section 5.1.

In our algorithm we exploit clustering to guide the selective sampling. Clustering has been used in the selective sampling process in other works in different ways.

Dasgupta and Hsu use hierarchical clustering (Dasgupta and Hsu, 2008). Their method exploits the cluster structure (if there is any) in the unlabelled data. Their algorithm assumes that querying the label of only one of the data points in a cluster is sufficient to determine the label of the other data points in that cluster. We only assume that the equivalence classes (which are much smaller) are quite pure.

Nguyen and Smeulders (2004) proposed a density-based approach that first clusters instances and tries to avoid querying outliers by propagating label information to instances in the same cluster. They first select points from the large clusters and use them to build a logistic regression classifier. Additional points are selected, either with the largest uncertainty or points that are cluster centres. The algorithm is obviously very different from the algorithm we propose here. Moreover, our algorithm is based on an NN classifier whereas theirs is based on logistic regression. Thus, in our case, using the point with the largest uncertainty would not necessarily approximate the best contribution to the classifier.

Similarly, Xu et al. (2007) use clustering to construct sets of queries for batch-mode active learning with SVMs. Specifically, they query the centroids of clusters of instances that lie closest to the decision boundary.

3 Distance metric learning using ensemble clustering

We now turn to define our ensemble clustering-based distance metric. Let A be a set of instances, where each $x_i \in A$ is a vector in some space χ. Instances are assumed to be i.i.d. distributed according to some unknown fixed distribution ρ. The Euclidean distance defined on A does not always reflect the actual similarity or dissimilarity of the objects to be classified. However, it is known that 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 for choosing the best clustering. Several attempts have been made to select the optimal parameter values of the clustering algorithms in supervised and unsupervised settings,

usually in the range image and colour image domain, but a general solution to this problem has not been found (Min et al., 2004; Chabrier et al., 2006; Zhang et al., 2008). We therefore decided to run different clustering algorithms several times with different parameter values. The result of all these runs yields a cluster ensemble (Fern, and Brodley, 2004).

There are many approaches proposed that can be used in our method to generate different multiple clustering; Strehl and Ghosh (2002) apply various clustering algorithms. Fred and Jain (2003) use one algorithm with different built-in initialisation and parameters, and Fern and Brodley (2003) project data onto different subspaces, Topchy et al. (2003) choose different subsets of features. Fischer and Buhmann (2003b) and Minaei-Bidgoli et al. (2004) select different subsets of data points and run a clustering algorithm on each subset. All these are instances of these generative mechanisms.

In our approach the clustering results are stored in a matrix denoted the *cluster matrix* $C \in Mat_{N\times K}$, where $N = |A|$ and K is the number of times the clustering algorithms were run. The i^{th} row consists of the cluster identities of the i^{th} point in the different runs. This results in a new instance space, $\chi_{cl} = \mathbb{Z}^K$, which contains the rows of the *cluster matrix*. The new distance between points from this space should be defined in order to reflect our intuitive notion of proximity among the corresponding points.

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

where $dis(x_i, y_i) = \begin{cases} 1 & x_i \neq y_i \\ 0 & x_i = y_i \end{cases}$ is the metric of a single feature. This metric is known as the *Hamming distance*. The idea of measuring the similarity between objects according to their clustering labels was introduced in Strehl and Ghosh (2002). 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 now work with A/E, which yields $C^{eq} \in Mat_{M\times K}$, where $M = |A/E|$. Points which always belong to different clusters will, however, be infinitely distant (i.e., $d_{cl}(x, y) = \infty$ if and only if x and y always belong to different clusters). Thus, x is a neighbour of y if and only if $d_{cl}(x, y) < \infty$. The set of the neighbours of x will be defined as: $= \{y|d_{cl}(x, y) < \infty\}$. For each $x \in A$, $\mathsf{N}_x = \emptyset$, since by using the reflexive property of E, we get $d_{cl}(x, x) = 0 < \infty$, and thus, $x \in \mathsf{N}_x$.

The goal now is to adapt the kNN classifier to work with this new distance metric using the new instance space. Consider the following paradigm. Let X be the *unlabelled data* – a set of unlabelled instances where each x_i is a vector in χ. 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 $D = \{\langle x_i, f(x_i)\rangle: x_i \in X, i = 1, \ldots, N_D\}$ be the *training data* – a set of labelled examples already known. In our algorithm the sets X and D are

represented by the equivalence classes X_{cl} and D_{cl} respectively. In this setting all the unlabelled points in X_{cl} will be labelled according to a given training dataset D_{cl}.

As our metric is general and can be used with any approach which uses a distance metric, we decided to evaluate how this new metric function improves the selection of most informative samples to be labelled and added to the training dataset. We decided to work with the *selective sampling for nearest neighbour classifier* (LSS) (Lindenbaum et al., 2004), because it is based on *k*NN.

The main assumption made by the algorithms is that *equivalent points have the same label*, but this assumption does not always hold in practice. Several options exist for overcoming this hurdle. One is to label several points from each equivalence class $x_{cl} \in D_{cl}$; x_{cl} will then be labelled according to the majority voting. Another option is to label x_{cl} according to its centroid. 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 equivalence classes to yield ones that are smaller but hopefully purer.

4 Ensemble clustering using the mean shift and *k*-means algorithms

As mentioned above, the main problem with a clustering-based approach is that there is no known method for choosing the best clustering. It is unknown how many clusters there should be, their shapes, which clustering algorithm is best, and which parameter values should be used. We therefore decided to run two different clustering algorithms several times with different parameter values. We use the well-known *k*-means algorithm (MacQueen, 1967) and the mean shift clustering algorithm (Georgescu et al., 2003; Comaniciu and Meer, 2002) in order to build the cluster matrix. Our algorithm, however, is general and any good clustering algorithm could be used. In general there is no specific clustering algorithm that is suitable for all datasets and the choice of the clustering algorithm depends on the dataset. Thus, each clustering algorithm that maps the dataset structure and yields pure equivalence classes will work. In addition, the clustering matrix can contain clustering results from different clustering algorithms.

For completeness we will now give a short overview of the mean shift algorithm. Mean shift is a non-parametric clustering algorithm. As it requires no prior knowledge of the number of clusters nor places any constraints on their shape, it is ideal for handling clusters of arbitrary shape and number. In addition, it is 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 centres 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 with a 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; see Georgescu et al. (2003) and Comaniciu and Meer (2002) for more details. 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 neighbours k as a parameter and the bandwidth is determined for each point in the data as the distance to its k^{th} neighbour. We chose 30 different values of k with fixed intervals between 1% to 30% of N (for more details see Section 6).

Some clustering algorithms work with continuous parameters, such as the mean shift algorithm described above, or with continuous weights over the features, such as the EDISON program which will be discussed in Section 7. In these cases the differences between two consecutive iterations might be small. There are two possible ways to deal with these similar clusterings: by eliminating the clustering results or by simply taking all of them. We preferred the latter because, if a set of samples were clustered together in several clustering runs, there is a higher probability that these samples belong to one class. So if we eliminate them we stand to lose this information. However, it is not efficient to preserve similar clustering runs. Therefore, we decided to join them, as a result of which the dimensionality of the data is reduced. We use the Rand (1971) index, which is a measure of similarity between two data clusterings, to decide whether to join them.

The Rand index is defined as follows. 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 that are in a different set in $C1$ and in a different set in $C2$, s describes the number of pairs of elements that are in the same set in $C1$ and in a different set in $C2$, and δ describes the number of pairs of elements that are in a different set in $C1$ and in the same set in $C2$.

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

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

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 Lookahead algorithm for selective sampling using ensemble clustering

As mentioned above, we also evaluate the improvement achieved by the ensemble-clustering-based metric by applying it to selective sampling methods.

Selective sampling is one of the common active learning approaches. It assumes that a set of unlabelled examples is available, and the learner selects an unlabelled example from the given set and asks the teacher to label it. It is important in many cases when we wish to construct a training dataset or add examples to it in order to improve the classifier's accuracy. In real environments, it is usually difficult to obtain a large set of labelled examples because each example must be labelled by a domain expert. Reducing the number of the training examples is therefore essential.

The algorithm that will now be described is based on the *selective sampling for nearest neighbour classifier* (LSS) (Lindenbaum et al., 2004). We will therefore first review this algorithm and then describe the algorithm we developed (LSS-EC) which is based on the ensemble clustering distance.

5.1 Lookahead algorithm for selective sampling

An *active-learner* consists of a classifier learning algorithm L, and a selective sampling algorithm S_L. The selective sampling algorithm determines which unlabelled instance in X should be labelled by the teacher f, which is a mapping $f: \chi \rightarrow \mathcal{W}$.

The active learner first applies S_L to choose one unlabelled instance x from X. The label w of x is then revealed and the pair (x, w) is added to D and x is removed from X. Then the learner applies L to induce a new classifier. This sequence repeats until some stopping criterion is satisfied.

The *lookahead* algorithm for selective sampling considers all the unlabelled examples and selects the example that yields the best expected classifier. Let $U_L(x, D)$ be a *utility function* that estimates the merit of adding an unlabelled instance x to the set D as a training example for learning algorithm L. Let $P(f(x) = w|D)$ denote the conditional class probabilities of x for a given labelled set D. For each unlabelled example, its expected utility is measured using the utility function on the training set and using expected probabilities for the possible classes of the unlabelled example. Then the lookahead algorithm for selective sampling with respect to learning algorithm L selects the example that leads to the learning example with the highest expected utility. This algorithm is depicted in Algorithm 1.

Algorithm 1 Lookahead selective sampling (X, D)

1. *If D is empty, return a random point from X.*
2. *Otherwise, set* $U_{max} \leftarrow 0$.
3. *For each* $x \in X$ *do*.
   - *(a)* $D' \leftarrow D \cup \{\langle x, -1\rangle\}$.
   - *(b) Compute class probabilities for all points in X based on data D'.*
   - *(c) Compute utility by approximating the accuracy AL (D') of the classifier based on data D',* $U_1 \leftarrow A_L(D')$.
   - *(d) Repeat the above steps for* $D' \leftarrow D \cup \{\langle x, -1\rangle\}$ *and get* $U_2(x)$.
   - *(e) Compute class probabilities for x based on data D.*
   - *(f)* $U(x) \leftarrow P(f(x) = -1|D) \cdot U_1(x) + P(f(x) = 1|D) \cdot U_2(x)$.
   - *(g) If* $U_{max} < U(x)$ *then* $U_{max} \leftarrow U(x)$, $x_{best} \leftarrow x$.
4. *Return* x_{best}.

In order to be able to use this general algorithm for a specific learner, $A_L(D)$ and $P(f(x) = w|D)$ have to be defined. $A_L(D)$ denotes the expected accuracy of classifier $h = L(D)$ as:

$$A_L(D) = \frac{1}{|X|}\sum_{x \in X} P(f(x) = h(x) \mid D), \tag{5}$$

is produced by a learning algorithm L on labelled data D.

As $f(x)$ is unknown for the unlabelled dataset X, we are not able to calculate these probabilities. A possible solution for this problem is to use the maximum likelihood estimation, which assumes that if $P(h(x) = 1 \mid D) > \frac{1}{2}$, then $f(x) = 1$ else $f(x) = -1$. Therefore, we define:

$$P_{\max}(x \mid D) = \max\left(P(h(x) = 1 \mid D),\ P(h(x) = -1 \mid D)\right)$$

and use it to estimate $A_L(D)$ as follows:

$$A_L(D) = \frac{1}{|X|}\sum_{x \in X} P_{\max}(x \mid D). \tag{6}$$

The last piece of the puzzle is to assign conditional class probabilities to the nearest neighbour classifier. To this end the random field model is used to estimate the probabilities for the 2-NN classifier.

In order to calculate $P(f(x) = 1|D)$, first the two nearest neighbours y, z from the labelled data D must be found. Then the probability will be

$$P(f(x) = 1 \mid y, z) = \frac{1}{2} + \frac{l(y) \cdot \gamma\left(d(x,y) + l(z)\right) \cdot \gamma\left(d(x,z)\right)}{\frac{1}{2} + 2 \cdot l(y) \cdot l(z) \cdot \gamma\left(d(y,z)\right)},$$

where $l(x)$ is the label of x and $d(x, y)$ is the distance between x and y

$$\gamma(d) = 0.25e^{-d/\sigma}, \sigma = \frac{1}{\mathfrak{D}} \frac{1}{|N|^2} \sum_{p \in X} \sum_{q \in X} d(p,q), \tag{7}$$

where $\mathfrak{D}$ is a scaling parameter.

5.2 Selective sampling using ensemble clustering

We will now describe the changes we made to the original LSS formulas to cause the algorithm to work with the new distance metric and the new instance space. In each iteration of the LSS-EC algorithm, one of the equivalence classes is chosen to be added to the training dataset. Its representative point x_0 is labelled by the expert and added to D_{cl}. This is in contrast to the original LSS where a single point is chosen. In order to choose the equivalence class, the expected accuracy from (6) will be modified to be:

$$A_L\left(D'_{cl}\right) = \frac{1}{|X|}\left[\sum_{x \in D_{cl}} P_{\max}\left(x \middle| D'_{cl}\right) \cdot card(x) + P_{\max}\left(x_0 \middle| D'_{cl}\right) \cdot card\left(x_0\right) + \sum_{x \in X_{cl}^{eq} \setminus D'_{cl}} P_{\max}\left(x \middle| D'_{cl}\right) \cdot card(x)\right]$$

where X_{cl}^{eq} is a set of equivalence points, $D'_{cl} = D_{cl} \cup \{x_0\}$, and $card(x)$ is the cardinality of x, which is $|[x]_E|$. The probability $P_{\max}(x \mid D'_{cl})$ of each point is multiplied by its cardinality because in the original space X each point represents $|[x]_E|$ points which we assume have the same label. $P_{\max}(x \in D_{cl} \mid D'_{cl}) = 1$ because D_{cl} is labelled. As a result, the expected accuracy will be:

$$A_L\left(D'_{cl}\right) = \frac{1}{|X|}\left[\sum_{x \in D_{cl}} card(x) + card\left(x_0\right) + \sum_{x \in X_{cl}^{eq} \setminus D'_{cl}} P_{\max}\left(x \middle| D'_{cl}\right) \cdot card(x)\right].$$

The contribution of the point $x0$ to the utility function is:

$$\frac{1}{|X|}\left[\left(1 - P_{\max}\left(x_0 \middle| D_{cl}\right)\right) \cdot card\left(x_0\right) + \sum_{x \in X_{cl}^{eq} \setminus D'_{cl}} \left(P_{\max}\left(x \middle| D'_{cl}\right) - P_{\max}\left(x \middle| D_{cl}\right)\right) \cdot card(x)\right]. \tag{8}$$

Studying the two components of (8), we can see that the first, which is termed the uncertainty component, is the product of the cardinality of the set of points equivalent to x_0 with the reduction in uncertainty obtained since x_0 has been chosen. $P_{max}(x_0|D_{cl})$ is the probability of x_0 to be classified correctly in the past, and now since x_0 is chosen its

probability will be 1. The second component, which is termed the classifier component, estimates the added contribution to the utility function, which will be obtained by reducing the uncertainty of the neighbours of x_0 to less than what it was for set D_{cl}. This term measures how the classifier improves when x_0 is added to the labelled set.

As in the original LSS algorithm, in our algorithm the size of $\mathfrak{D}$ also affects performance. As $\mathfrak{D}$ increases, $P_{\max}(x \mid D'_{cl})$ decreases. As a result, for large values of $\mathfrak{D}$ the points are selected only according to their cardinalities and not by the probabilities of their neighbours. As the size of $\mathfrak{D}$ decreases, the importance of the neighbourhood increases.

We ran the lookahead selective sampling algorithm using the d_{cl} metric and the new accuracy to choose the most informative points to be given to the expert. The rest of the points were labelled by the classifier h using the d_{cl} metric. The LSS ensemble clustering method (LSS-EC) is expected to be much more efficient than LSS since $|X/E| \ll |X|$.

6 Experiments on numerical datasets

In order to measure the ability of the new distance function to reflect the actual similarity or dissimilarity between objects, we ran two sets of experiments: on numerical and on image datasets. For the numerical datasets, we measured the improvement of our distance function on the performance of the *k*NN and LSS algorithms. The results will be presented in this section, and the results for the image dataset in Section 7.We first compare the performance of *k*NN-EC to that of *k*NN. We then compare the performance of our LSS-EC algorithm to the random sampling, uncertainty sampling, and LSS methods. All the algorithms were implemented in MATLAB.

We ran our experiments on three standard numerical datasets: the image segmentation dataset [UCI machine learning repository (Frank and Asuncion, 2010)], the breast cancer dataset [LIBSVM library (Chang and Lin, 2011)], and Leo Breiman's ring norm (Bias, 1996). The image segmentation dataset contains 2310 instances, which are divided into seven classes. Since we chose towork with a binary *k*NN, the classes were joined to create two class labels (as was done in Lindenbaum et al., 2004), one corresponding to BRICKFACE, SKY and FOLIAGE and the other corresponding to CEMENT, WINDOW, PATH and GRASS. The breast cancer dataset contains 683 instances, divided into two class labels, with 444 points from the first class and the rest from the second. Leo Breiman's ring norm dataset contains 7,400 instances of a two-class classification problem. Each class is drawn from a multivariate normal distribution. All these datasets were labelled, but this knowledge was used only to evaluate the accuracy of the resulting classifier. In all experiments these datasets are assumed to be unlabelled.

To build the cluster matrix, we used the mean shift or the *k*-means algorithms, as discussed in Section 4. As our method is general we arbitrarily chose to start with the mean shift algorithm for all the numerical datasets with the *k* or *h* values described there. But, because the mean shift algorithm did not yield pure equivalence classes for breast cancer and ring norm datasets, we used the *k*-means algorithm for these two datasets. For the breast cancer dataset, *k*-means was run with $k = 3.15$ and for the ring norm dataset it was run with $k = 3.30$. Later (in the experiment in Section 6.1.3) we can see that the values of *k* are not critical for the algorithms performance. The results are stored in the

cluster matrix C. The equivalence relation E was employed to build the equivalence matrix C^{eq}. 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 Dataset properties

Dataset type	*Dataset*	*Dataset size*	*Cluster matrix size*	*Equivalence matrix size*	*Ratio %*	*Purity %*
Numerical datasets	I. Segmentation	2,310 × 19	2310 × 38	548 × 38	24	97
	Breast Cancer	683 × 8	683 × 13	160 × 13	23	98
	Ring Norms	7,400 × 20	7,400 × 28	7,400 × 28	100	100
Image datasets	One Bird	131 × 100	13,100 × 18	1,312 × 18	10	99
	Three Birds	207 × 352	72,864 × 11	3,073 × 11	4	98
	Wolf	321 × 481	154,401 × 24	3,348 × 24	2	96.7

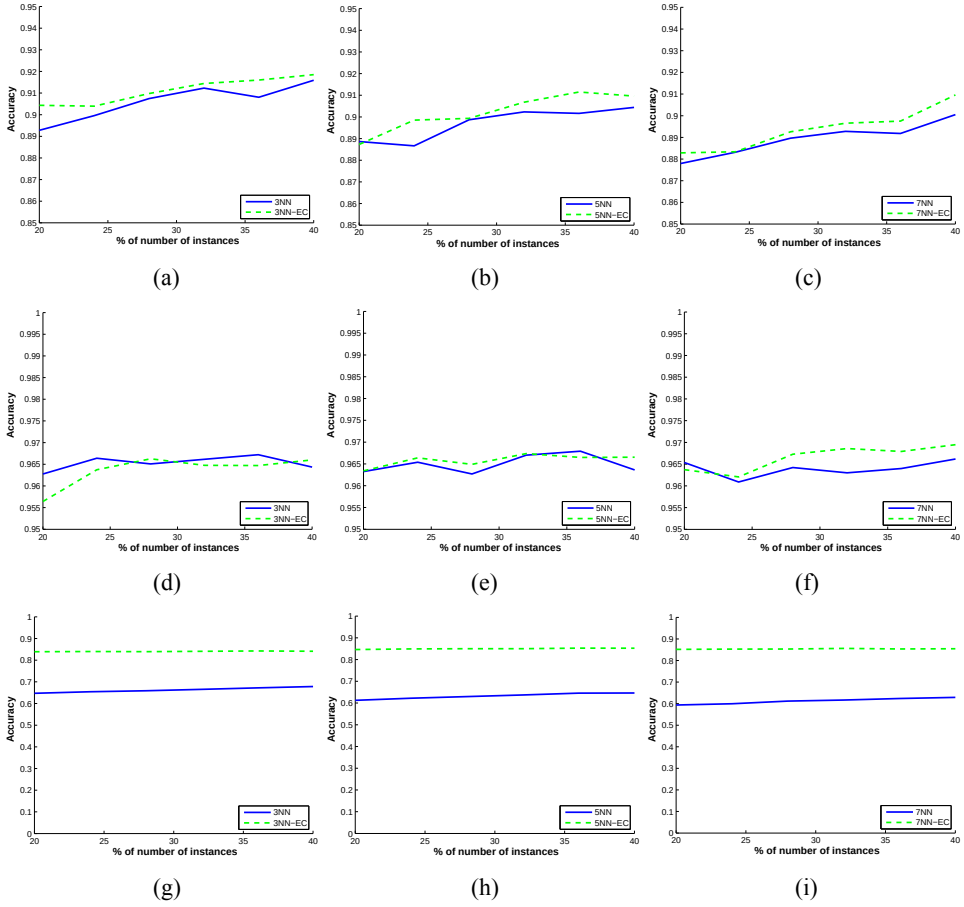
6.1 kNN-EC experiments

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

6.1.1 Results

As can be seen from Figure 2, *k*NN-EC performs better than or comparable to *k*NN with the Euclidean distance. The learning curves are constructed by computing the ratio of correctly classified instances to the whole unlabelled data. For the image segmentation and breast cancer datasets, the curves show that *k*NN-EC performs comparably to *k*NN, while for ring norm they show that *k*NN-EC exhibits superior performance. As Figure 2 shows, *k*NN-EC achieves accuracy of 85% while *k*NN achieves accuracy of 65%. This improvement in *k*NN-EC accuracy is due to the ability of the EC metric to better measure the actual similarity between the objects. We also computed the runtime of the two algorithms when the training dataset includes 30% of the points, and $k = 5$. Table 2 shows that using the EC metric usually results in a speedup of the algorithm. The exception is the ring norm dataset, for which our algorithm works slowly but that is because in this dataset we do not have equivalence classes, as can be seen in Table 1. We thus performed another experiment to determine how much the runtime of our algorithm depends on the number of equivalence classes. In this experiment we ran *k*-means on a different range (i.e., $k = 3.10$ instead of 3.30) and we got 3351 equivalence classes. As a result, the runtime of the *k*NN-EC algorithm is only 1.04 seconds.

Figure 2 Results of *k*NN and *k*NN-EC for the three datasets (see online version for colours)



Notes: The first row shows the learning curves of the image segmentation dataset, the second shows the breast cancer dataset, and the third shows the ring norm dataset. The columns show the learning curves for the different *k* values.

Table 2 Runtime of the algorithms in seconds

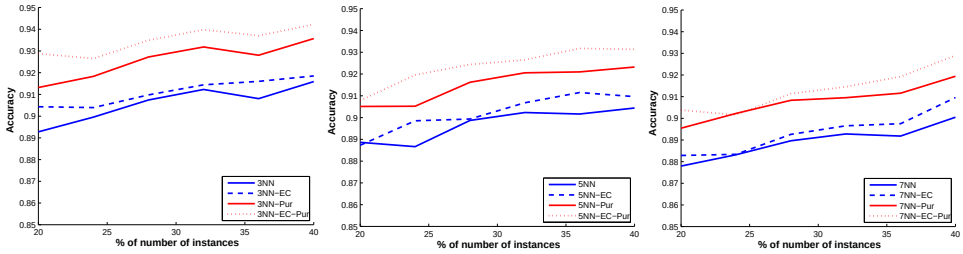
Dataset	*kNN*	*kNN-EC*
Image segmentation	0.45	0.15
Breast cancer	0.15	0.01
Ring norms	2.4	3.9

6.1.2 The effect of the purity of the equivalence classes

As shown in the previous subsection, the performance of *k*NN-EC is not always superior. We conducted an experiment to determine the extent to which our algorithm's performance depends on the purity of the equivalence classes. Unlike the previous experiment where the equivalence classes were not completely pure (e.g., 97% for the image segmentation dataset), in this experiment the classes were changed until a purity of

100% was obtained. As can be seen in Figure 3, there is a linear relationship between the accuracy and the purity of the equivalence classes. The accuracy increased by about 3% while the purity increased from 97% to 100%.

Figure 3 The effect of purity: Results of *k*NN and *k*NN-EC for the image segmentation dataset with the different *k* values (see online version for colours)



(a) (b) (c)

6.1.3 The effect of number of clustering iterations on kNN-EC performance

We performed yet another experiment to determine the extent to which our algorithm depends on the number of clustering iterations. In this experiment we evaluate the performance of the 5NN-EC classifier on the ring norm dataset, with 20% of the dataset used 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 clustering runs increases, the purity of the equivalence classes also increases, and the number of equivalence classes increases dramatically. (When the clustering runs increased from 8 to 18, the purity increased from 93 to 99.8% and the number of equivalence classes increased from 3351 equivalence classes to 7171). However, the performance of the algorithm remain stable (i.e., the accuracy increased only 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 equivalent instances would be 1 instead of zero. Thus with high probability, these instances would still be classified as belonging to the same class.

Table 3 The effect of the number of clustering iterations

k-means	*Equivalence matrix size*	*Purity %*	*Accuracy%*
$k = 3.10$	3,351 × 8	93	81
$k = 3.20$	7,171 × 18	99.8	83
$k = 3.30$	7,400 × 28	100	85

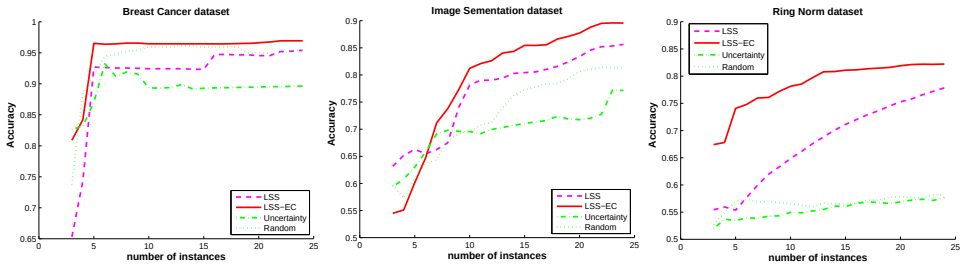
6.2 LSS-EC experiments

We now turn to the selective sampling algorithms. In the first stage of the algorithms, a training set of size 4 was randomly drawn and labelled. The algorithms were then asked to select 20 additional points. During each iteration the active learner selects a sample point to be labelled and added to the training set. After each iteration the accuracy was evaluated by the ability of the classifier to label the rest of the unlabelled points. The

results were averaged over 10 different runs of the algorithms on each dataset. For each dataset we constructed a curve to evaluate how well the algorithms select the points.

Figure 4 shows the superior performance of LSS-EC, which outperforms its competitors over all the datasets. It also converges faster to its maximum after only one iteration.

Figure 4 Results of LSS and the LSS-EC for the numerical datasets (see online version for colours)



(a) (b) (c)

7 Experiments with images

In a second set of experiments we tested our algorithms using three real colour images. We used images for two reasons: first, they provide large, complex datasets, and second, the results obtained by applying learning algorithms on them are easy to view and evaluate. The first image shows one bird [Figure 5(a)], the second shows three birds [Figure 6(a)], and the third [from the Berkeley database (Martin et al., 2001)] shows a wolf [Figure 6(c)]. Each image was manually segmented into two classes, the foreground (birds and wolf) and the background, yielding the ground truth as shown in Figures 5(b), 6(b) and 6(d). The reader can appreciate that segmenting these images using a colour-based segmentation algorithm will not be an easy task.

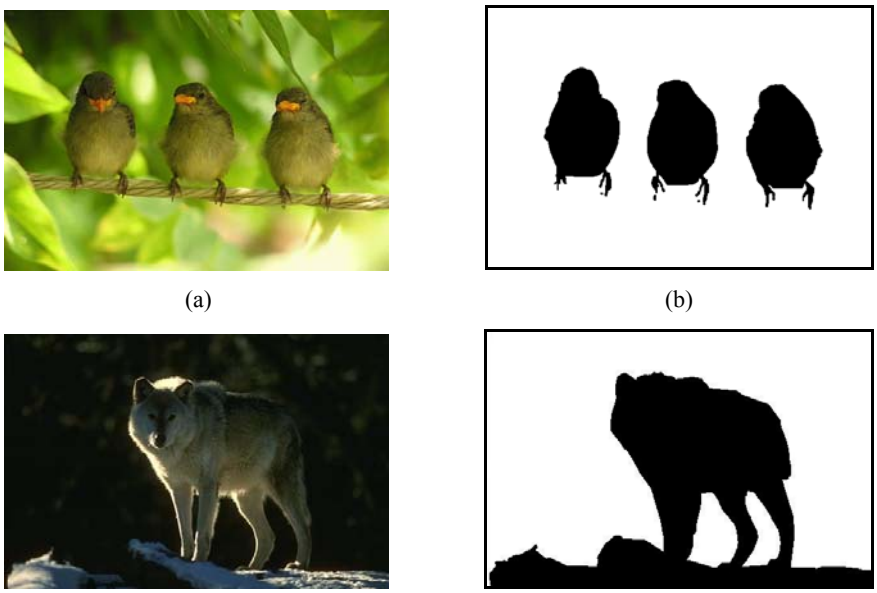
As can be seen in the 6th column in Table 1, the number of equivalence classes is usually between 2% and 10% of the number of pixels. As the complexity of the algorithm for each iteration is $O(N^2)$, the running time of LSS-EC is two orders of magnitude smaller than the running time of LSS. Therefore we are able to run the LSS-EC algorithm on the large number of pixels in these images, which we are not able to do with the much slower LSS algorithm. We therefore compared the LSS-EC algorithm only with the uncertainty and the random sampling algorithms.

We worked with the edge detection and image segmentation (EDISON) system (Christoudias et al., 2002). This program implements the mean shift image segmentation algorithm described in Section 4. Each pixel in the image is represented by its two image coordinates and RGB colour 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 *hr* (the RGB values). The output of this program is a clustered image.

Figure 5 (a) The original image with one bird (b) The classified image (goal) (c) The equivalence class image (see online version for colours)

(a) (b) (c)

Figure 6 (a) and (c) The original images (b) and (d) The classified images (goal) (see online version for colours)



(a) (b)

(c) (d)

Each cluster was assigned a colour, (i.e., points in the same cluster have the same colour). Figure 7 shows some of these clustering results for the single bird image. 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 uncertainty and random sampling methods have 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 these methods can be compared to LSS-EC. In the experiments we therefore ran them using all 24 bandwidth pair values.

Figure 7 Output of the EDISON system (see online version for colours)

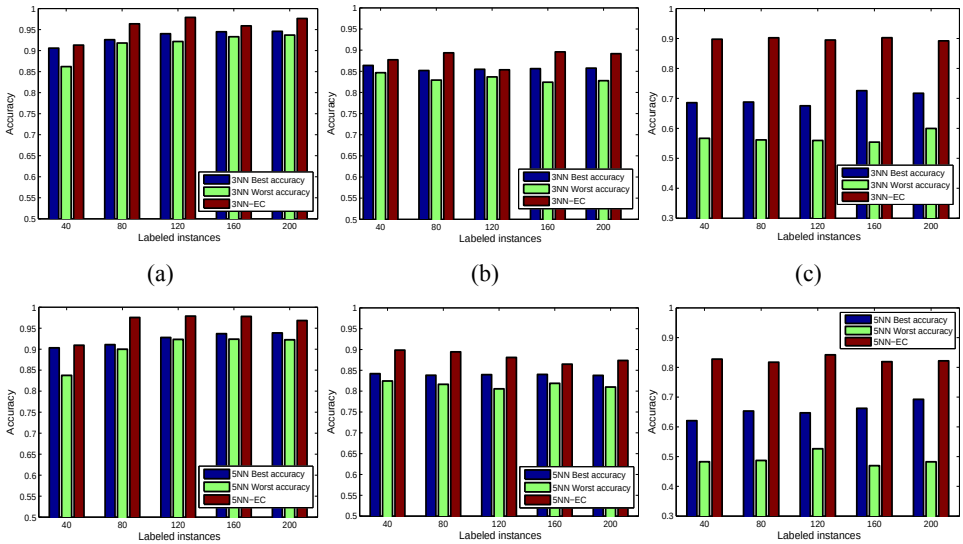
(a) (b) (c)

The clustering images were mapped to the cluster matrix C, in the same way as in the first three datasets. From this matrix the equivalence point matrix C^{eq} was generated. Figure 5(c) shows the equivalence image for the one bird image.

7.1 kNN-EC experiments on images

The *k*NN-EC and *k*NN algorithms were tested on the three images. The two classifiers were evaluated on several training sets of 40 to 200 pixels, with different numbers of neighbour values (i.e., k = 3, 5). As the optimal bandwidth parameters cannot be found automatically in the *k*NN algorithm, in Figure 8 we show only the best case and the worst case of the *k*NN algorithm for each training set, and compare them to the *k*NN-EC accuracy.

Figure 8 Results of *k*NN and *k*NN-EC for the three images (see online version for colours)



(a) (b) (c)

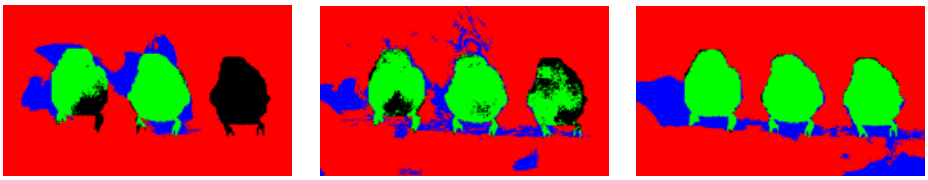
(d) (e) (f)

As can be seen from Figure 8, *k*NN-EC performs better than *k*NN with the Euclidean distance with different values of k. The learning bars, which describe the accuracy for each classifier of the three images, show that *k*NN-EC is superior over all the image datasets. For example, when it was run with k=5 and 120 labelled pixels on the one bird image, it achieved 96.4% accuracy while the best accuracy of *k*NN is 95.4%. For the three birds image the *k*NN-EC algorithm's performance was about 87% while the *k*NN algorithm achieved 84% in the best case. For the wolf image the *k*NN-EC algorithm was superior, with performance of about 84% while the *k*NN algorithm achieved 64% in the best case.

Figure 9 shows an example of running the *k*NN-EC and *k*NN (the best and the worst cases) algorithms with $k = 5$ with 120 labelled pixels as a training set on the three birds image.

From these experiments it is clear that the EC metric is not only more efficient and more meaningful than the Euclidean distance but it also gives us uniform results. This is because the EC metric combines several possible sets of weights, while the weighted Euclidean distance uses only one set of weights which have to be determine somehow.

Figure 9 Results of 5NN and 5NN-EC for the three birds image dataset using 120 labelled pixels as a training dataset, (a) the output for the worst case of *k*NN (b) the output for the best case of *k*NN (c) the output for *k*NN-EC (see online version for colours)



(a) (b) (c)

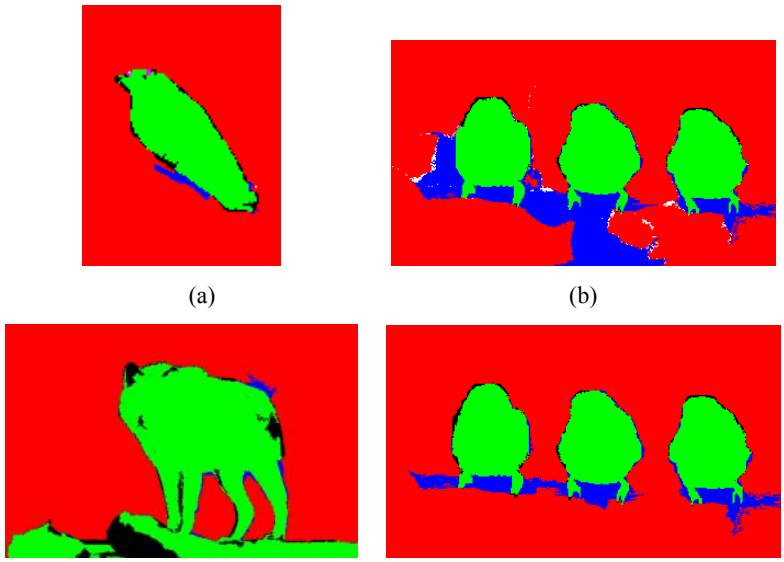
Note: The colour of the pixels represents the results of the classifier. Red is background, green is birds, blue is misclassified background and black is misclassified birds pixels.

7.2 LSS-EC experiments on images

For each dataset the LSS-EC algorithm was given as input the matrix C^{eq}, and four randomly selected labelled points. It was then asked to sample 26 more points from each image. The results of the algorithm for the three images are shown in Figure 11. The classified images are shown in Figure 10.

Examining Figure 10(b), we see many misclassified pixels. This occurred because the equivalence classes were not pure (as shown in Table 1). In several cases a point belonging to the minority was chosen, causing the most of the equivalence class to be wrongly labelled. When the centroid pixel of each equivalence class is chosen to be labelled by the expert, it worked much better, as can be seen in Figure 10(d).

Figure 10 Output of LSS-EC, (a) the output for choosing 30 points from the single bird image (b) the result for choosing 30 points from the three birds image (c) the output for choosing 30 points from the wolf image (d) the LSS-EC results using labelled centroid pixels (see online version for colours)

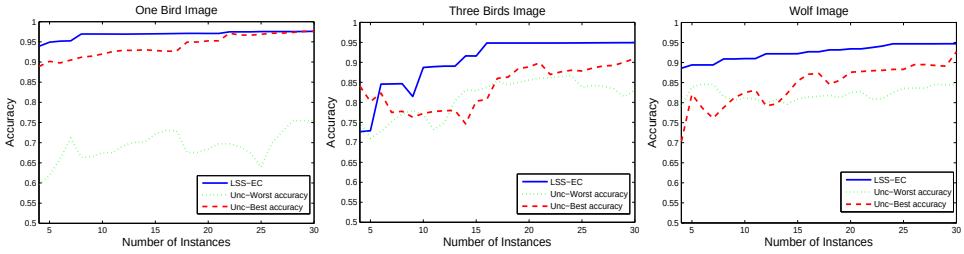


(a) (b)

(c) (d)

Notes: The colour of the pixels represents the results of the classifier. Red is background, green is foreground, white is unknown background, magenta is unknown foreground, blue is incorrect background and black is incorrect foreground pixels.

Figure 11 Results of the LSS-EC and uncertainty algorithms over the images according to the chosen pixels (see online version for colours)



(a) (b) (c)

Note: The blue curve describes the LSS-EC results, the red curve describes the best uncertainty results, and the green curve describes the worst uncertainty results.

In a final experiment we compare LSS-EC with uncertainty and random sampling. As mentioned above, we had to run the algorithm 24 times with all the different values of the two parameters, yielding 24 different distance metrics. In Figure 11 we plot for each image the best and worst obtained results. We can see in these plots that LSS-EC performs better than uncertainty sampling. Furthermore the accuracy of the latter varies greatly, as can be seen by studying the curves. For the single bird image, our algorithm achieved 97.7% accuracy with 30 points. Similar results were previously obtained using

three points. In comparison, the best uncertainty sampling achieved only 90.0% accuracy. The random sampling algorithm achieved 83.3%–95% accuracy but further analysis showed worse results for the foreground pixels: between 4.5% to 93.4%. In some cases the foreground object was not detected at all. For the three birds image our algorithm achieved 93.25% accuracy with 30 points. The maximal results were already achieved after 13 points were sampled. This is in contrast to uncertainty sampling, which achieved only 80.0% accuracy for this number of samples.

All in all, the accuracy of the random sampling algorithm ranges between 74.6%–86.6%. For the wolf image our algorithm achieved 94.7% accuracy with 30 points while the best case of the uncertainty algorithm achieved accuracy of 92.7%. Random sampling achieved accuracy values between 70.5% and 78.5%.

8 Conclusions

In this work, we have presented a new unsupervised distance learning metric based on ensemble clustering and used it within the *k*NN classifier and the LSS framework. Each data point is characterised by the identity of the clusters it belongs to in several clustering runs. The distance metric is defined as the Hamming distance between these clustering results. Our experimental results show that the new ensemble-based distance function reflects the actual similarity between the objects better than the Euclidean one.

This results in a better *k*NN classifier and a better LSS algorithm. Moreover, our observation that all points which always belong to the same cluster form an equivalence class means that the algorithm only has to consider one member of each such class. This reduces the complexity of the algorithms considerably (by at least two orders of magnitude in our case). This equivalence class representation is, in effect, a private case of the more general concept of data reduction. As such, it 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.

Because our distance metric does not depend on a specific clustering algorithm, it can be used with any good clustering algorithm whose resulting clusters are strongly correlated to the classes. In addition, according to the experimental results we conclude that the number of the clusterings runs is not critical for the algorithm's performance.

Moreover, we saw that with some datasets, the Euclidean distance does not gives us uniform results because it sometimes requires different parameter values whereas, the new clustering-based distance metric takes into account the different parameter values and yields uniform and better results.

Acknowledgements

This research was supported by the IMG4 consortiums of the Ministry of Industry and Commerce as well as a graduate student fellowship from the Ministry of Science and Technology.

References

AbedAllah, L. and Shimshoni, I. (2012) 'k nearest neighbor using ensemble clustering', *Proceedings of the 14th Data Warehousing and Knowledge Discovery*, pp.265–278

Belkin, M. and Niyogi, P. (2003) 'Laplacian eigenmaps for dimensionality reduction and data representation', *Neural Computation*, Vol. 15, No. 6, pp.1373–1396.

Bias, L. (1996) *Variance and Arcing Classifiers*, Tec. Report 460,Statistics Department.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D.P., Schapire, R.E. and Warmuth, M.K. (1997) 'How to use expert advice', *Journal of the ACM (JACM)*, Vol. 44, No. 3, pp.427–485.

Chabrier, S., Emile, B., Rosenberger, C. and Laurent, H. (2006) 'Unsupervised performance evaluation of image segmentation', *EURASIP Journal on Applied Signal Processing*, No. 1, pp.1–12.

Chang, C-C. and Lin, C-J. (2011) 'LIBSVM: a library for support vector machines', *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 27, pp.1–27 [online] http://www.csie.ntu.edu.tw/-cjlin/libsvm.

Chopra, S., Hadsell, R. and LeCun, Y. (2005) 'Learning a similarity metric discriminatively, with application to face verification', *IEEE Conf. on Computer Vision and Pattern Recognition*, pp.26–33.

Christoudias, C., Georgescu, B. and Meer, P. (2002) 'Synergism in low level vision', in *Proceedings of International Conference on Pattern Recognition*, pp.150–155.

Cohn, D., Atlas, L. and Ladner, R. (1994) 'Improving generalization with active learning', *Machine Learning*, Vol. 15, No. 2, pp.201–221.

Comaniciu, D. and Meer, P. (2002) 'Mean shift: A robust approach toward feature space analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp603–619.

Cover, T. and Hart, P. (1967) 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp.21–27.

Dagan, I. and Engelson, S.P. (1995) 'Committee-based sampling for training probabilistic classifiers', in *Proceedings of the 12th International Conference on Machine Learning*, pp.150–157.

Dasgupta, S. and Hsu, D. (2008) 'Hierarchical sampling for active learning', in *Proceedings of the 25th International Conference on Machine Learning*, pp.208–215.

Davis, D.T. and Hwang, J.N. (1992) 'Attentional focus training by boundary region data selection', in *Proceedings of International Joint Conference on Neural Networks*, pp.676–681.

Derbeko, P., El-Yaniv, R. and Meir, R. (2004) 'Explicit learning curves for transduction and application to clustering and compression algorithms', *Journal of Artificial Intelligence Research*, Vol. 22, No. 1, pp.117–142.

Domeniconi, C., Gunopulos, D. and Peng, J. (2005) 'Large margin nearest neighbor classifiers', *IEEE Transactions on Neural Networks*, Vol. 16, No. 4, pp.899–909.

Dudoit, S. and Fridlyand, J. (2003) 'Bagging to improve the accuracy of a clustering procedure', *Bioinformatics*, Vol. 19, No. 9, pp.1090–1099.

Fern, X.Z. and Brodley, C.E. (2003) 'Random projection for high dimensional data clustering: a cluster ensemble approach', in *Proceedings of 20th International Conference on Machine Learning*, pp.186–193.

Fern, X.Z. and Brodley, C.E. (2004) 'Solving cluster ensemble problems by bipartite graph partitioning', in *Proceedings of the 21st International Conference on Machine Learning*, pp.36–43, ACM.

Fischer, B. and Buhmann, J.M. (2003a) 'Bagging for path-based clustering', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 11, pp.1411–1415.

Fischer, B. and Buhmann, J.M. (2003b) 'Path-based clustering for grouping of smooth curves and texture segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 4, pp.513–518.

Frank, A. and Asuncion, A. (2010) 'UCI machine learning repository' [online] http://archive.ics.uci.edu/ml (accessed 2013).

Fred, A.L.N. and Jain, A.K. (2003) 'Data clustering using evidence accumulation', in *Proceedings of 16th International Conference on Pattern Recognition*, pp.276–280.

Georgescu, B., Shimshoni, I. and Meer, P. (2003) 'Mean shift based clustering in high dimensions: a texture classification example', in *Proceedings of the 9th International Conference on Computer Vision*, pp.456–463.

Goldberger, J., Roweis, S., Hinton, G. and Salakhutdinov, R. (2004) 'Neighbourhood components analysis', in *Proceedings of the 21st International Conference of Advances in Neural Information Processing System*, pp.513–520.

Gonzalez, R.C. and Woods, R.E. (2001) *Digital Image Processing*, 2nd ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Hastie, T. and Tibshirani, R. (1996) 'Discriminant adaptive nearest neighbor classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 6, pp.607–616.

Lewis, D.D. and Catlett, J. (1994) 'Heterogeneous uncertainty sampling for supervised learning', in *Proceedings of the 11th International Conference on Machine Learning*, pp.148–156.

Lewis, D.D. and Gale, W.A. (1994) 'A sequential algorithm for training text classifiers', in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.3–12, Springer-Verlag, New York, Inc.

Lindenbaum, M., Markovitch, S. and Rusakov, D. (2004) 'Selective sampling for nearest neighbor classifiers', *Machine Learning*, Vol. 54, No. 2, pp.125–152.

MacQueen, J.B. (1967) 'Some methods for classification and analysis of multivariate observations', *Proceedings of the 5th Symposium on Math*, *Statistics*, *and Probability*, pp.281–297.

Martin, D., Fowlkes, C., Tal, D. and Malik, J. (2001) 'A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics', in *Proceeding of the 8th International Conference on Computer Vision*, Vol. 2, pp.416–423, July.

Min, J., Powell, M. and Bowyer, K.W. (2004) 'Automated performance evaluation of range image segmentation algorithms', *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part B*. *Cybernetics*, Vol. 34, No. 1, pp.263–271.

Minaei-Bidgoli, B., Topchy, A. and Punch, W.F. (2004) 'Ensembles of partitions via data resampling', in *Proceedings of International Conference on Information Technology*. *Coding and Computing*, pp.188–192.

Nguyen, H.T. and Smeulders, A. (2004) 'Active learning using pre-clustering', in *Proceedings of the 21st International Conference on Machine Learning*, pp.623–630.

Rand, W.M. (1971) 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical Association*, Vol. 66, No. 336, pp.846–850.

Saul, L.K. and Roweis, S.T. (2003) 'Think globally, fit locally: unsupervised learning of low dimensional manifolds', *The Journal of Machine Learning Research*, Vol. 4, pp.119–155.

Shalev-Shwartz, S., Singer, Y. and Ng, A.Y. (2004) 'Online and batch learning of pseudo-metrics', in *Proceedings of the 21st International Conference on Machine Learning*, pp.94–102, ACM.

Strehl, A. and Ghosh, J. (2002) 'Cluster ensembles – a knowledge reuse framework for combining multiple partitions', *The Journal of Machine Learning Research*, Vol. 3, pp.583–617.

Tan, M. and Schlimmer, J.C. (1990) 'Two case studies in cost-sensitive concept acquisition', in *Proceedings of the 8th National Conference on Artificial Intelligence*, pp.854–860.

Tenenbaum, J.B., Silva, V. and Langford (2000) 'A global geometric framework for nonlinear dimensionality reduction', *Science*, Vol. 290, No. 5500, pp.19–23.

Topchy, A., Jain, A.K. and Punch, W. (2003) 'Combining multiple weak clusterings', in *Third IEEE International Conference on Data Mining*, pp.331–338, IEEE.

Topchy, A., Jain, A.K. and Punch, W. (2005) 'Clustering ensembles: Models of consensus and weak partitions', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 12, pp.1866–1881.

Turney, P.D. (1995) 'Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm', *Journal of Artificial Intelligence Research*, Vol. 2, No. 1, pp.369–409.

Weinberger, K.Q. and Saul, L.K. (2006) 'Distance metric learning for large margin nearest neighbor classification', *Proceedings of the 18th International Conference of Advances in Neural Information Processing Systems*, pp.1473–1480.

Xu, Z., Akella, R. and Zhang, Y. (2007) 'Incorporating diversity and density in active learning for relevance feedback', *Advances in Information Retrieval*, Vol. 4425, pp.246–257.

Zhang, H., Fritts, J.E. and Goldman, S.A. (2008) 'Image segmentation evaluation: a survey of unsupervised methods', *Computer Vision and Image Understanding*, Vol. 110, No. 2, pp.260–280.