Pattern Anal Applic (2016) 19:311–323
DOI 10.1007/s10044-014-0401-y

THEORETICAL ADVANCES

An efficient algorithm for large-scale quasi-supervised learning

Bilge Karaçalı

Received: 8 May 2013 / Accepted: 22 July 2014 / Published online: 6 August 2014

Abstract We present a novel formulation for quasi-supervised learning that extends the learning paradigm to large datasets. Quasi-supervised learning computes the posterior probabilities of overlapping datasets at each sample and labels those that are highly specific to their respective datasets. The proposed formulation partitions the data into sample groups to compute the dataset posterior probabilities in a smaller computational complexity. In experiments on synthetic as well as real datasets, the proposed algorithm attained significant reduction in the computation time for similar recognition performances compared to the original algorithm, effectively generalizing the quasi-supervised learning paradigm to applications characterized by very large datasets.

Keywords Quasi-supervised learning · Posterior probability estimation · Nearest neighbor rule · Large-scale pattern recognition · Transductive inference

1 Introduction

Across the spectrum of statistical learning algorithms requiring various degrees of guidance from available data, supervised pattern classification algorithms such as the nearest neighbor rule [6, 9], support vector machines [5, 25], artificial neural networks [12], discriminant functions [21], and fuzzy classifiers [1, 18, 20] have enjoyed a particularly wide audience ranging from object recognition to biomedical data analysis. Such a far-reaching pertinence can be attributed to the ability of these algorithms to construct decision rules based on a given set of training samples for which the desired decisions are already available. The decision rules effectively infer the conditional dependence of the true decisions on the patterns using the training data and generalize it to form predictions on future data.

Recently, the quasi-supervised learning method was proposed to address the issue of learning on overlapping datasets that arise in applications where obtaining manually curated ground truth training datasets is problematic and the available labelings are unreliable [14]. Note that in classical pattern classification problems with clear class definitions, the overlap of the datasets associated with different classes reflects an inadequacy of the collected features to present clearly separable regions in the observation space for the respective classes. The dataset overlap under consideration in this case, however, is caused by the lack of adequate labeling of the data points due to a variety of possible reasons, such as the sheer volume of data points to manually label, errors in existing labels, or a complete lack of labels for one of the classes of interest. Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be two datasets of samples drawn from the distributions

$$p(x|\mathcal{C}_0) = \lambda_0 p_r(x) + (1-\lambda_0) p_{\mathcal{C}_0}(x) \quad (1)$$

and

$$p(x|\mathcal{C}_1) = \lambda_1 p_r(x) + (1-\lambda_1) p_{\mathcal{C}_1}(x), \quad (2)$$

respectively; quasi-supervised learning aims to identify the samples in $\mathcal{C}_0$ and $\mathcal{C}_1$ drawn, respectively, from $p_{\mathcal{C}_0}(x)$ and $p_{\mathcal{C}_1}(x)$ in the absence of any representative samples of these distributions. In the expressions above, $p_r(x)$ represents the

Electronic supplementary material The online version of this article (doi:10.1007/s10044-014-0401-y) contains supplementary material, which is available to authorized users.

B. Karaçalı (✉)
Department of Electrical and Electronics Engineering, İzmir Institute of Technology, Urla, İzmir 35430, Turkey
e-mail: bilge@iyte.edu.tr

overlap between $\mathcal{C}_0$ and $\mathcal{C}_1$, $p_{\mathcal{C}_0}(x)$ and $p_{\mathcal{C}_1}(x)$ govern the samples specific to the corresponding dataset and absent in the other, and $\lambda_0, \lambda_1 \in [0, 1]$ control the extent of the overlap. After learning, the samples are classified into three categories, one for samples in $\mathcal{C}_0$ highly specific to $\mathcal{C}_0$, another for those in $\mathcal{C}_1$ highly specific to $\mathcal{C}_1$, and a third for samples that are not specific to either and can figure equally in $\mathcal{C}_0$ as in $\mathcal{C}_1$.

Note that this problem description deviates from the usual binary classification setting where $\lambda_0 = \lambda_1 = 0$ due to the overlap between $\mathcal{C}_0$ and $\mathcal{C}_1$. This overlap would task the classical supervised learning algorithms treating the datasets as representing distinct classes with the separation of a set of $p_r(x)$ samples in $\mathcal{C}_0$ from another set of $p_r(x)$ samples in $\mathcal{C}_1$ in an exercise in futility. Especially in cases characterized by a large overlap with $1 - \lambda_0 << 1$ and/or $1 - \lambda_1 << 1$, these algorithms place a great emphasis on separating the overlapping samples and, in the process, risk losing track of the few differentiating samples [14]. The distinction between the quasi-supervised learning problem and semi-supervised learning also rests on this overlap, precluding training sets of $p_{\mathcal{C}_0}(x)$ and $p_{\mathcal{C}_1}(x)$ samples from exacting a separation boundary guided by the dispersion patterns of the unlabeled samples in the observation space [4].

Likewise, when one of λ_0 or λ_1 is equal to zero, the recognition task coincides with abnormality detection, as well as a restricted case of multiple instance learning with the whole of $\mathcal{C}_0$ representing one single sample (or bag in the corresponding terminology) characterized by the instances therein and $\mathcal{C}_1$ representing the other bag to be differentiated from the first [2, 3, 7]. For general λ_0 and λ_1, the quasi-supervised learning algorithm derived in [14] allows identifying samples in $\mathcal{C}_0$ and $\mathcal{C}_1$ that are exclusively specific to their respective datasets without any identifying samples for $p_{\mathcal{C}_0}(x)$, $p_{\mathcal{C}_1}(x)$ or $p_r(x)$, or any knowledge of λ_0 and λ_1, a task not undertaken by any other learning paradigm.

From a Bayesian perspective, the quasi-supervised learning problem can be addressed by representing the probability densities $p_r(x)$, $p_{\mathcal{C}_0}(x)$ and $p_{\mathcal{C}_1}(x)$ in terms of parametric families and deriving the conditions under which the unknown parameters can be determined uniquely from available data. Once the estimates for the distribution parameters are formulated, optimal recognition rules can be derived based on a probability model of choice. The solution offered by the quasi-supervised learning algorithm described in [14], on the other hand, involves non-parametric and model-free estimates of the dataset posterior probabilities at each sample. This estimation is carried out using the pairwise distances between the samples in $\mathcal{C}_0$ and $\mathcal{C}_1$ by a low computational complexity scheme that can be shown to converge to the unknown true posterior probabilities when the number of samples in the two datasets grows large.

Note that among the conventional learning strategies, fuzzy classification appears to be the only one suitable to sort through the samples observed in overlapping datasets, since it allows associating samples to the classes via fuzzy memberships. Indeed, the quasi-supervised learning algorithm can be viewed as a special form of fuzzy classification, with fuzzy class memberships expressed explicitly by class posteriors in a situation where the available datasets represent distinct classes (see [18], Definition 1.1.2). Posterior probabilities also capture the reliability concept of conflict proposed in [19], since it would produce roughly equal posteriors for the datasets for samples that are situated between them, identifying such samples as nonspecific to either dataset. A fuzzy classification-based solution to the quasi-supervised learning problem described above, however, has not been proposed to date.

The main computationally intensive component of the quasi-supervised learning algorithm is the computation of the pairwise distances. This can amount to a substantial computational load for large sample sets, though learning on a dataset containing over 55,000 samples was carried out successfully in a previous application [15]. Nonetheless, the computational load associated with calculating, storing and further processing all pairwise distances limits the use of the algorithm for learning over larger datasets composed of hundreds of thousands of samples or more.

In this paper, we derive a novel algorithm that computes the dataset posterior probabilities using a conditional probability decomposition over sample groups or clusters. This group formulation avoids the computation of pairwise sample distances and, instead, computes the posterior probability estimates using the sample to cluster distances. Given that the number of clusters that summarize the data can be orders of magnitude smaller than the number of samples, this amounts to a dramatic reduction in the data storage requirements as well as the overall computational complexity. In experiments on synthetic and real datasets, the proposed algorithm achieved comparable recognition performances to the original algorithm in significantly reduced computation times. Improvement in recognition accuracy was also observed in some cases, which can be attributed to a secondary effect of the group formulation on the learning framework regularizing the resulting posterior probability estimates.

The details of the proposed algorithm based on a novel group formulation for quasi-supervised learning over large datasets are provided in the next section. Section 3 presents the results of comparative performance evaluation experiments on synthetic and real datasets against the original quasi-supervised learning algorithm, followed by concluding remarks in Sect. 4.

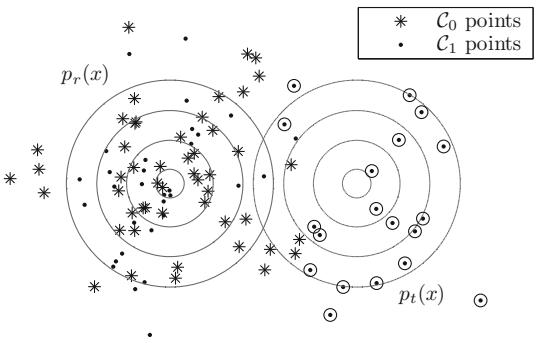

Fig. 1 Illustration of the simplified quasi-supervised learning problem. The points in $\mathcal{C}_0$ and $\mathcal{C}_1$ are represented by the *asterisk* and *dot* symbols. The learning problem is to recognize the points in $\mathcal{C}_1$ drawn from the target distribution marked by *circle dot*

2 Methodology

In this section, we first provide a technical derivation for the asymptotic property of the nearest neighbor classification rule that allows estimating the posterior probabilities $p(\mathcal{C}_0|x)$ and $p(\mathcal{C}_1|x)$ at a given sample x, and describe the original quasi-supervised learning algorithm that computes these estimates for each sample in a collection. Next, we derive the novel group formulation that decomposes the dataset posterior probabilities conditionally over sample groups. Finally, we frame the proposed algorithm for large-scale quasi-supervised learning.

In the following, we use a simpler quasi-supervised problem setting by letting $\lambda_0 = 1$, $\lambda_1 = \lambda$, and $p_{\mathcal{C}_1}(x) = p_t(x)$, allowing $\mathcal{C}_0$ to represent a homogeneous dataset of samples drawn from a reference distribution $p_r(x)$ and $\mathcal{C}_1$ to represent a mixed dataset of unlabeled samples, with the objective of recognizing the samples in $\mathcal{C}_1$ drawn from $p_t(x)$ (Fig. 1). Note, however, that while this allows an easier interpretation of the posterior probability estimation scheme, it does not limit its application to the general class of quasi-supervised learning problems described earlier.

2.1 Estimation of posterior probabilities using the nearest neighbor rule

Let $x_o \in \mathbf{X}$ and $p_X(x)$ be the probability density function of a random variable X defined over $\mathbf{X}$ equipped with a metric d. The cumulative distribution function of the random variable $D = d(X, x_o)$ is then defined by

$$P_D(\delta) = \int\limits_{x \in B_\delta(x_o)} p_X(x)\mathrm{d}x \tag{3}$$

for all $\delta \geq 0$, where $B_\delta(x_o)$ denotes the ball of radius δ around x_o. The corresponding probability density function $p_D(\delta)$ is also defined in the usual way as the derivative of $P_D(\delta)$ with respect to δ. Note that when δ is small, $p_X(x) \simeq p_X(x_o)$ for $x \in B_\delta(x_o)$ and these distributions can be approximated by

$$P_D(\delta) = V(\delta)p_X(x_o) \tag{4}$$

and

$$p_D(\delta) = V'(\delta)p_X(x_o) \tag{5}$$

where $V(\delta)$ is the volume of a hypersphere of radius δ in $\mathbf{X}$, and $V'(\delta)$ its derivative with respect to δ.

Next, let the collection $X_1, X_2, \ldots, X_n$ be independent and identically distributed with $p_X(x)$. The cumulative distribution function $P_{D^m}(\delta)$ governing the minimum distance $D^m = \min_i d(X_i, x_o)$ is given by

$$P_{D^m}(\delta) = 1 - (1 - P_D(\delta))^n \tag{6}$$

with the associated density function

$$p_{D^m}(\delta) = n(1 - P_D(\delta))^{n-1} p_D(\delta) \tag{7}$$

following the formulations for the distributions of extreme values [8].

Now, consider the minimum distances D_0^m and D_1^m observed over n_0 points drawn from $p(x|\mathcal{C}_0)$ and n_1 points from $p(x|\mathcal{C}_1)$, respectively, populating a random reference set $R = \{(X_j, y_j)|\ X_j \in \mathbf{X}, y_j \in \{0, 1\}, j = 1, 2, \ldots, n_0 + n_1\}$ for nearest neighbor classification, represented by the labeling rule

$$f(x_o|R) = y_{j^\star}, \quad j^\star = \arg\min_j d(X_j, x_o)\,. \tag{8}$$

Clearly, the rates at which the point x_o is assigned to $\mathcal{C}_0$ or $\mathcal{C}_1$ are given by the probabilities $\Pr\{D_0^m < D_1^m\}$ and $\Pr\{D_1^m < D_0^m\}$ with

$$\Pr\{D_0^m < D_1^m\} + \Pr\{D_1^m < D_0^m\} = 1.$$

Expanding $\Pr\{D_0^m < D_1^m\}$ over the joint probability distribution $p_{D_0^m, D_1^m}(\delta_0, \delta_1)$ and using the independence of D_0^m and D_1^m provide

$$\begin{aligned}
\Pr\{D_0^m < D_1^m\} &= \int\limits_{\delta_0, \delta_1} \mathbf{1}(\delta_0 < \delta_1) p_{D_0^m, D_1^m}(\delta_0, \delta_1)\mathrm{d}\delta_0\mathrm{d}\delta_1 \\
&= \int\limits_{\delta_0=0}^{\infty} \int\limits_{\delta_1=\delta_0}^{\infty} p_{D_0^m}(\delta_0) p_{D_1^m}(\delta_1)\mathrm{d}\delta_1\mathrm{d}\delta_0 \\
&= \int\limits_{\delta_0=0}^{\infty} \left(\int\limits_{\delta_1=\delta_0}^{\infty} p_{D_1^m}(\delta_1)\mathrm{d}\delta_1 \right) p_{D_0^m}(\delta_0)\mathrm{d}\delta_0 \\
&= \int\limits_{\delta_0=0}^{\infty} (1 - P_{D_1^m}(\delta_0)) p_{D_0^m}(\delta_0)\mathrm{d}\delta_0
\end{aligned}$$

where $\mathbf{1}(\cdot)$ returns 1 when its argument is true and 0 otherwise. Due to the asymptotic properties of the non-

negative extreme value distributions for sufficiently large n_0 and n_1, the probability masses in $P_{D_0^m}(\delta_0)$ and $P_{D_1^m}(\delta_0)$ become concentrated in an interval $[0, \Delta]$ with $\Delta << 1$. This implies that

$$\Pr\{D_0^m < D_1^m\} \simeq \int_{\delta_0=0}^{\Delta} (1 - P_{D_1^m}(\delta_0)) p_{D_0^m}(\delta_0) \mathrm{d}\delta_0.$$

Replacing the extreme value distributions with the respective expressions derived earlier followed by further algebraic manipulations indicated above expresses the probability $\Pr\{D_0^m < D_1^m\}$ as the sum of two terms as described later.

Further simplifications can be obtained by noting that since Δ is small, $P_{D_1}(\delta_0) \simeq 0$ and $p_{D_0}(\delta_0) \simeq V'(\delta_0) p(x_o|\mathcal{C}_0)$ for $\delta_0 \in [0, \Delta]$. This eliminates the second term on the right-hand side and allows expressing $\Pr\{D_0^m < D_1^m\}$ as in Eq. (9). Repeating the same derivation for $\Pr\{D_1^m < D_0^m\}$ provides the expression in Eq. (10). Taking the ratio of both sides, we obtain

$$\begin{aligned}\frac{\Pr\{D_0^m < D_1^m\}}{\Pr\{D_1^m < D_0^m\}} &\simeq \frac{n_0 p(x_o|\mathcal{C}_0)}{n_1 p(x_o|\mathcal{C}_1)} \\ &\simeq \frac{p(\mathcal{C}_0|x_o)}{p(\mathcal{C}_1|x_o)} \frac{n_0 p(\mathcal{C}_1)}{n_1 p(\mathcal{C}_0)}.\end{aligned}$$

$$\begin{aligned}\Pr\{D_0^m < D_1^m\} &\simeq \int_{\delta_0=0}^{\Delta} \Big(1 - \big(1 - (1 - P_{D_1}(\delta_0))^{n_1}\big)\Big) n_0 \\ &\quad \times (1 - P_{D_0}(\delta_0))^{n_0 - 1} p_{D_0}(\delta_0) \mathrm{d}\delta_0 \\ &\simeq n_0 \int_{\delta_0=0}^{\Delta} (1 - P_{D_1}(\delta_0))^{n_1} (1 - P_{D_0}(\delta_0))^{n_0 - 1} \\ &\quad \times p_{D_0}(\delta_0) \mathrm{d}\delta_0 \\ &\simeq n_0 \int_{\delta_0=0}^{\Delta} (1 - P_{D_1}(\delta_0))(1 - P_{D_1}(\delta_0))^{n_1 - 1} \\ &\quad \times (1 - P_{D_0}(\delta_0))^{n_0 - 1} p_{D_0}(\delta_0) \mathrm{d}\delta_0 \\ &\simeq n_0 \int_{\delta_0=0}^{\Delta} (1 - P_{D_1}(\delta_0))^{n_1 - 1} (1 - P_{D_0}(\delta_0))^{n_0 - 1} \\ &\quad \times p_{D_0}(\delta_0) \mathrm{d}\delta_0 \\ &\quad - n_0 \int_{\delta_0=0}^{\Delta} P_{D_1}(\delta_0)(1 - P_{D_1}(\delta_0))^{n_1 - 1} \\ &\quad \times (1 - P_{D_0}(\delta_0))^{n_0 - 1} p_{D_0}(\delta_0) \mathrm{d}\delta_0\end{aligned}$$

$$\begin{aligned}\Pr\{D_0^m < D_1^m\} &\simeq n_0 p(x_o|\mathcal{C}_0) \\ &\times \int_{\delta_0=0}^{\Delta} (1 - P_{D_1}(\delta_0))^{n_1 - 1} (1 - P_{D_0}(\delta_0))^{n_0 - 1} V'(\delta_0) \mathrm{d}\delta_0 \end{aligned} \quad (9)$$

$$\begin{aligned}\Pr\{D_1^m < D_0^m\} &\simeq n_1 p(x_o|\mathcal{C}_1) \\ &\times \int_{\delta_1=0}^{\Delta} (1 - P_{D_0}(\delta_1))^{n_1 - 1} (1 - P_{D_1}(\delta_1))^{n_0 - 1} V'(\delta_1) \mathrm{d}\delta_1 \end{aligned} \quad (10)$$

where the last step follows from the Bayes rule. Note that the second term in the expression above disappears when $n_0/n_1 = p(\mathcal{C}_0)/p(\mathcal{C}_1)$. By the same token, assuming equal prior probabilities for both $\mathcal{C}_0$ and $\mathcal{C}_1$ and subsequently letting $n_0 = n_1 = n$ provide

$$\frac{\Pr\{D_0^m < D_1^m\}}{\Pr\{D_1^m < D_0^m\}} \simeq \frac{p(\mathcal{C}_0|x_o)}{p(\mathcal{C}_1|x_o)}. \quad (11)$$

Finally, since

$$\Pr\{f(x_o|R_n) = 0\} = \Pr\{D_0^m < D_1^m\}$$

and

$$\Pr\{f(x_o|R_n) = 1\} = \Pr\{D_1^m < D_0^m\}$$

by definition, this shows that the posterior probabilities $p(\mathcal{C}_0|x_o)$ and $p(\mathcal{C}_1|x_o)$ can be estimated by the average fraction of times x_o is assigned to $\mathcal{C}_0$ and $\mathcal{C}_1$ via a nearest neighbor classification rule operated using random reference sets R_n with sufficiently large n.

2.2 The original quasi-supervised learning algorithm

The quasi-supervised learning algorithm estimates the probabilities $\Pr\{f(x|R_n) = 0\}$ and $\Pr\{f(x|R_n) = 1\}$ by computing the fraction of times a sample x is assigned to classes $\mathcal{C}_0$ and $\mathcal{C}_1$ for all possible reference sets R_n constructed using n points from the datasets $\mathcal{C}_0$ and $\mathcal{C}_1$, corresponding with a minor abuse of notation to the respective classes [14]. To this end, the probability $\Pr\{y = 0\}$ with $y = f(x|R_n)$ and R_n, restricted to the distinct reference sets in $\mathcal{C}_0 \bigcup \mathcal{C}_1$, is decomposed as

$$\begin{aligned}&\Pr\{y = 0\} \\ &= \Pr\{y = 0|(x_{(1)}, y_{(1)}) \in R_n\} \Pr\{(x_{(1)}, y_{(1)}) \in R_n\} \\ &\quad + \Pr\{y = 0|x_{(1)} \notin R_n\} \Pr\{x_{(1)} \notin R_n\} \\ &= \mathbf{1}(y_{(1)} = 0) \Pr\{(x_{(1)}, y_{(1)}) \in R_n\} \\ &\quad + \Pr\{y = 0|(x_{(1)}, y_{(1)}) \notin R_n\} \Pr\{(x_{(1)}, y_{(1)}) \notin R_n\}.\end{aligned}$$

In the expression above, $\{x_{(i)}\}$ indicates a ranking of the points in x_i in increasing distance to x with $x_{(1)}$ the closest, and $\{y_{(i)}\}$ indicates their labels. This decomposition links $\Pr\{y = 0\}$ to the conditioning event $(x_{(1)}, y_{(1)}) \in R_n$, since the presence of $x_{(1)}$ in R_n sets the label produced by the corresponding nearest neighbor classifier to $y_{(1)}$ regardless of the other points in R_n. Clearly, this decomposition can be

carried out further to compute $\Pr\{y=0|(x_{(1)},y_{(1)})\notin R_n\}$ using the conditioning event $(x_{(2)},y_{(2)})\notin R_n$ and so on for $\Pr\{y=0|(x_{(i)},y_{(i)})\notin R_n, i=1,2,\ldots,k\}$ until such k for which

$$\min\left\{\sum_{i=k}^{\ell}\mathbf{1}(y_{(i)}=0),\sum_{i=k}^{\ell}\mathbf{1}(y_{(i)}=1)\right\}=n,$$

making the probability $\Pr\{(x_{(k)},y_{(k)})\in R_n|(x_{(i)},y_{(i)})\notin R_n, i=1,2,\ldots,k-1\}=1$. Collecting back the probabilities starting from this limiting value of k computes $\Pr\{y=0\}$ as well as $\Pr\{y=1\}=1-\Pr\{y=0\}$, and estimates the posterior probabilities of $\mathcal{C}_0$ and $\mathcal{C}_1$ at the sample x via

$$p_0(x)\triangleq\Pr\{f(x|R_n)=0\} \tag{12}$$

and

$$p_1(x)\triangleq\Pr\{f(x|R_n)=1\}, \tag{13}$$

respectively. For a sample x_i in one of $\mathcal{C}_0$ or $\mathcal{C}_1$, $p_0(x_i)$ and $p_1(x_i)$ are computed by carrying out this procedure using the reduced collection $\{x_j\}$, $j=1,2,\ldots,\ell$, $j\neq i$, so that whether $x_i\in\mathcal{C}_0$ or $x_i\in\mathcal{C}_1$ does not affect the calculations in accordance with a leave-one-out framework. Finally, the parameter n is selected to minimize the functional

$$E(n)=4\sum_{i=1}^{\ell}p_0(x_i)p_1(x_i)+2n \tag{14}$$

at an optimal trade-off between the separation of $\mathcal{C}_0$ and $\mathcal{C}_1$ calculated over the resulting posterior probabilities via the first term and the VC dimension of the corresponding nearest neighbor classification rule expressed by the second term [14, 16, 17].

Note that the computational complexity of the algorithm described above consists mainly of the computation and sorting of all pairwise distances at $O(\ell^2\log\ell)$. Assuming an exhaustive approach to optimize $E(n)$ that repeats the posterior probability calculations for each $n=1,2,\ldots,\ell$ at the worst case provides an overall complexity of $O(\ell^3\log\ell)$.

Note also that the quasi-supervised learning algorithm described above corresponds to a transductive learning strategy where the statistical learning occurs in the form of posterior probabilities estimated individually at each sample instead of an approximating function defined globally [25–27]. Furthermore, the estimated posterior probabilities are invariant to all nonlinear transformations of the sample space $\mathbf{X}$ such as via kernel functions replacing the original inner product as long as the induced re-structuring of the local neighborhoods procures a monotonic transformation of the distances, thereby preserving the order of $d(x,x_{(i)})$ for any $x\in\mathbf{X}$. Finally, since the posterior probability estimates $p_0(x_i)$ and $p_1(x_i)$ are computed exclusively based on the ordering of the pairwise distances $d(x_i,x_j)$, it suggests that the pairwise distances provide a complete characterization of the collection $\{x_i\}$ for statistical learning purposes.

2.3 Decomposition of nearest neighbor classification rates over sample groups

The quasi-supervised learning algorithm computes the probability of assigning a newly observed sample x into $\mathcal{C}_0$ or $\mathcal{C}_1$ based on the labels of the nearest points in a collection $\{x_i\}$ to x and their likelihoods of figuring in a random reference set R_n drawn from $\{x_i\}=\mathcal{C}_0\bigcup\mathcal{C}_1$, for $i=1,2,\ldots,\ell$. The posterior probabilities are then computed by accumulation over the resulting conditional probability decomposition that can extend up to $\ell-2n$ steps. Consequently, reducing the number of steps required for the computation of the posterior probabilities is critical for lowering the computational expense of the learning algorithm.

This issue can be addressed by considering the group $G=\{x_{(1)},x_{(2)},\ldots,x_{(\ell^G)}\}$ of ℓ^G samples in $\{x_i\}$ nearest to x, of which ℓ_0^G are from $\mathcal{C}_0$ and ℓ_1^G are from $\mathcal{C}_1$. Now, the probability $\Pr\{y=0\}$ of assigning the sample x to $\mathcal{C}_0$ by a nearest neighbor classifier using a reference set R_n chosen randomly from $\{x_i\}$ can be decomposed as

$$\begin{aligned}&\Pr\{y=0\}\\&\quad=\Pr\{y=0|R_n\cap G\neq\emptyset\}\Pr\{R_n\cap G\neq\emptyset\}\\&\qquad+\Pr\{y=0|R_n\cap G=\emptyset\}\Pr\{R_n\cap G=\emptyset\}\end{aligned} \tag{15}$$

with respect to the conditioning event $R_n\cap G=\emptyset$.

Now, if the samples in G cover a relatively small region in the observation space so that the probabilities $p(x|\mathcal{C}_0)$ and $p(x|\mathcal{C}_1)$ are approximately constant for $x_i\in G$, then the ordering between them becomes insignificant and inconsequential to the recognition problem. In that case, when one or more of them appear in the reference set R_n, the average rate at which x would be assigned to $\mathcal{C}_0$ across all possible orderings becomes ℓ_0^G/ℓ^G. Note that this is tantamount to replacing the samples $\{x_{(1)},x_{(2)},\ldots,x_{(\ell^G)}\}$ in G with another set of latent samples $\{\tilde{x}_{(1)},\tilde{x}_{(2)},\ldots,\tilde{x}_{(\ell^G)}\}$, each belonging ℓ_0^G/ℓ^G parts to $\mathcal{C}_0$ and ℓ_1^G/ℓ^G parts to $\mathcal{C}_1$. The expression for $\Pr\{y=0\}$ then becomes

$$\begin{aligned}&\Pr\{y=0\}\\&\quad=\frac{\ell_0^G}{\ell^G}\Pr\{R_n\cap G\neq\emptyset\}\\&\qquad+\Pr\{y=0|R_n\cap G=\emptyset\}\Pr\{R_n\cap G=\emptyset\}\end{aligned} \tag{16}$$

with

$$\Pr\{R_n \cap G = \emptyset\} = \frac{\binom{\ell_0 - \ell_0^G}{n}\binom{\ell_1 - \ell_1^G}{n}}{\binom{\ell_0}{n}\binom{\ell_1}{n}} = \prod_{i=0}^{n-1} \frac{(\ell_0 - \ell_0^G - i)(\ell_1 - \ell_1^G - i)}{(\ell_0 - i)(\ell_1 - i)}.$$

and $\Pr\{R_n \cap G \neq \emptyset\} = 1 - \Pr\{R_n \cap G = \emptyset\}$. Now, the probability $\Pr\{y = 0 | R_n \cap G = \emptyset\}$ can be decomposed further with respect to the conditioning event $R_n \cap G' = \emptyset$ with $G' = \{x_{(n^G+1)}, x_{(n^G+2)}, \ldots, x_{(n^G+n^{G'})}\}$ and so on over a succession of sample groups, until the probability of R_n and a subsequent group being mutually exclusive becomes zero due to the exhaustion of samples.

2.4 The large-scale quasi-supervised learning algorithm

The decomposition of the probability $\Pr\{y = 0\}$ over sample groups as described above offers substantial savings in computation time as it reduces the number of steps required in the calculation, but these groups still remain to be determined separately for each x_i. While such groupings can be carried out in a number of different ways such as treating large jumps in the sequence of ordered distances $d(x_i, x_{(j)})$ for $j = 1, 2, \ldots, \ell - 1$, as transition zones between successive groups, the approach is still challenged by two issues.

First, collecting x_i and x_j into the same group in computing $p_0(x)$ for some x just because $d(x, x_i) \simeq d(x, x_j)$ risks grouping together distant points that can be separated from each other by as much as $d(x, x_i) + d(x, x_j)$. While this is still legitimate within the context of estimating posterior distributions of $\mathcal{C}_0$ and $\mathcal{C}_1$ at x as shown in the previous section, it falls at odds with a more general notion of summarizing the dispersion of $\{x_i\}$ across $\mathbf{X}$ via clusters formed by samples at close proximity to each other. Second, carrying out the probability decomposition over groups determined independently for each sample does not address the greatest computational expense of the learning algorithm: the computation of the pairwise distances $d(x_i, x_j)$ for all $i, j = 1, 2, \ldots, \ell$, $i \neq j$.

We address both issues by considering a clustering of the samples $\{x_i\}$ into C_k containing $\ell_0^{C_k}$ and $\ell_1^{C_k}$ points from $\mathcal{C}_0$ and $\mathcal{C}_1$, $\ell_0^{C_k} + \ell_1^{C_k} = \ell^{C_k}$, for $k = 1, 2, \ldots, K$, with $K << \ell$. Given a sample-to-cluster distance measure ρ, computation of $p_0(x)$ for a sample $x \notin \{x_i\}$ then requires calculating the distances $\rho(x; C_k)$ from x to each cluster C_k, ranking the clusters in the ascending order of distances into $\{C_{(k)}\}$ containing $\ell_0^{C_{(k)}}$ and $\ell_1^{C_{(k)}}$ samples from $\mathcal{C}_0$ and $\mathcal{C}_1$, respectively.

The computation of $p_0(x_i)$ for a sample x_i in the collection involves a minor complication of revising the numbers $\ell_0^{C_{k^\star}}$ or $\ell_1^{C_{k^\star}}$ along with $\ell^{C_{k^\star}}$ for the cluster $C_{k^\star}$ with $x_i \in C_{k^\star}$. This revision prevents the knowledge of x_i figuring in $\mathcal{C}_0$ or $\mathcal{C}_1$ from affecting the results and preserves the leave-one-out formalism.

Given this cluster-oriented formulation for the computation of the posterior probabilities, what remains to be resolved is the constitution of the clusters C_k and the selection of a suitable sample-to-cluster distance measure ρ. The well-known cluster distances $\rho^{\min}$ and $\rho^{\max}$ defined by

$$\rho^{\min}(x_i; C_k) = \min_{x_j \in C_k} d(x_i, x_j) \tag{17}$$

and

$$\rho^{\max}(x_i; C_k) = \max_{x_j \in C_k} d(x_i, x_j) \tag{18}$$

are both inadequate as they entail computing all pairwise distances $d(x_i, x_j)$. On the other hand, the mean distance ρ^{mean} defined by

$$\rho^{\text{mean}}(x_i; C_k) = d(x_i, \mu_k) \tag{19}$$

where

$$\mu_k = \frac{1}{\ell^{C_k}} \sum_{x_j \in C_k} x_j$$

avoids the computation of pairwise distances and offers a viable option to assess the distance from the sample x_i to the cluster C_k.

As for the clustering of $\{x_i\}$, any method from the unsupervised learning literature can be used provided that it can be operated on large datasets. For instance, methods based on hierarchical clustering [13, 22, 23], or vector quantization [10, 11], can be incorporated to organize the samples in the collection $\{x_i\}$ in a way that avoids computing the pairwise distances $d(x_i, x_j)$ at least in a large part. Clearly, the simplest strategy is to randomly select K samples from the collection and carry out a nearest neighbor classification of all samples into the clusters represented by the selected ones in a random-k clustering. While this scheme does not guarantee optimality of the representation of the collection in any sense, it produces locally contiguous clusters with little computational expense. A more sophisticated strategy is the k-means clustering, revising the cluster centers with the arithmetic means of the samples assigned to the respective clusters followed by nearest neighbor classification anew until convergence.

At last, we formulate the proposed large-scale quasi-supervised learning algorithm that computes the posterior probabilities $p_0(x_i)$ and $p_1(x_i)$ for the datasets $\mathcal{C}_0$ and $\mathcal{C}_1$ at each sample x_i, $i = 1, 2, \ldots, \ell$, as follows:

- Partition $\{x_i\}$ into K clusters and compute $\ell_0^{C_k}$ and $\ell_1^{C_k}$.
- Initialize the $\ell \times K$ matrices L^0 and L^1 of sample counts.
- For $i = 1, 2, \ldots, \ell$,
  - Compute and sort $\rho^{\text{mean}}(x_i; C_k)$ in the ascending order for $k = 1, 2, \ldots, K$.
  - Populate the i'th rows of L^0 and L^1 via $L^0_{i,k} = \ell_0^{C_{(k)}}$ and $L^1_{i,k} = \ell_1^{C_{(k)}}$ for all k.
  - Find the index $k^\star$ with $x_i \in C_{(k^\star)}$, and reduce $L^0_{i,k^\star}$ by 1 if $x_i \in \mathcal{C}_0$ and $L^1_{i,k^\star}$ by 1.
- Otherwise, optimize $E(n)$ using the sample counts in L^0 and L^1 to compute $\{p_0(x_i)\}$ and $\{p_1(x_i)\}$.
- Return the probabilities $\{p_0(x_i)\}$ and $\{p_1(x_i)\}$ for the optimal n.

Note that while the algorithm above is implemented in a way to compute $p_0(x_i)$, it can easily be modified to compute $p_1(x_i)$ instead with no difference in the final outcome. The optimization for $E(n)$ is to be carried out numerically such as using a line search or by searching for the optimal n inside shrinking intervals. The number of clusters K and the choice of the clustering method remain to be specified as the operational parameters of the algorithm.

For a given choice of n, the computational complexity associated with the procedure above is determined essentially by the computation of the posterior probability $p_0(x_i)$ through K sample-to-cluster distances that are subsequently sorted at a complexity of $O(K \log K)$ for every x_i. Repeating this for ℓ samples, the overall complexity reaches $O(\ell K \log K)$. Since an exhaustive optimization of $E(n)$ recomputes the posterior probabilities for $n = 1, 2, \ldots, (\ell - 1)$, the worst-case complexity becomes $O(\ell^2 K \log K)$. Note that this entails a reduction upon the computational complexity of the original quasi-supervised learning algorithm by a factor of $K \log K / \ell \log \ell$.

3 Results

In this section, we present the experimental results obtained from a comprehensive performance evaluation of the proposed quasi-supervised learning algorithm. Following an illustrative comparison, the method is contrasted to the original quasi-supervised learning algorithm, in the absence of any other alternative method in the literature that addresses the quasi-supervised learning problem, in terms of both the recognition accuracy and the computation time on synthetic datasets representing controlled recognition tasks. The comparison is then extended to real datasets used in the prediction of N-linked glycosylation sites in amino acid sequences of human proteins and in high-energy particle identification on collections representing signal and background events.

In the experiments, a C language implementation of the proposed algorithm was executed within a Matlab environment (The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA). Briefly, the computation of the posterior probability for a given sample was carried out in C, while the clustering as well as the data management used Matlab routines. This implementation mirrored that of the original algorithm distributed at the Internet address http://web.iyte.edu.tr/~bilgekaracali/Projects/QSL/ to establish the comparability of the two algorithms in terms of computation times.

All experiments were carried out using a single core of an IBM ×3650 M2 rack server (IBM Corporation, 1 New Orchard Road, Armonk, New York 10504-1722, USA) equipped with two quad-core Intel Xeon processors (Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95054-1549, USA) and 52GB of RAM, operated by Debian Linux 6.0.4 (http://www.debian.org/).

3.1 Illustration of the group formulation for quasi-supervised learning

To elucidate the proposed algorithm for large-scale quasi-supervised learning, we have generated a reference dataset $\mathcal{C}_0$ and a mixed dataset $\mathcal{C}_1$ within the simplified learning framework illustrated in Fig. 1. Each dataset consisted of 200 samples, and the mixed dataset contained samples from the reference probability distribution $p_r(x)$ at a rate $\lambda = 0.75$, with the remaining samples drawn from the target distribution $p_t(x)$. We have then carried out the original quasi-supervised learning algorithm as well as the proposed method using random-k and k-means clustering schemes with $K = 20$ clusters separately to compute the posterior probability of the mixed dataset $\mathcal{C}_1$ at each sample. Finally, from each set of probabilities $\{p_1(x_i)\}$, we have computed the detection and false alarm rates

$$P_D(P_c) = \frac{1}{\sum\limits_{\substack{x_i \in \mathcal{C}_1 \\ x_i \sim p_t(x)}} 1} \sum_{\substack{x_i \in \mathcal{C}_1 \\ x_i \sim p_t(x)}} \mathbf{1}(p_1(x_i) > P_c) \tag{20}$$

and

$$P_{\text{FA}}(P_c) = \frac{1}{\sum\limits_{\substack{x_i \in \mathcal{C}_1 \\ x_i \sim p_r(x)}} 1} \sum_{\substack{x_i \in \mathcal{C}_1 \\ x_i \sim p_r(x)}} \mathbf{1}(p_1(x_i) > P_c) \tag{21}$$

for varying detection thresholds $P_c \in [0, 1]$ and calculated the receiver operating characteristic curves from the P_D–P_{FA} graphs.

The original data distribution and the $\mathcal{C}_1$ samples detected to have been drawn from the target distribution at

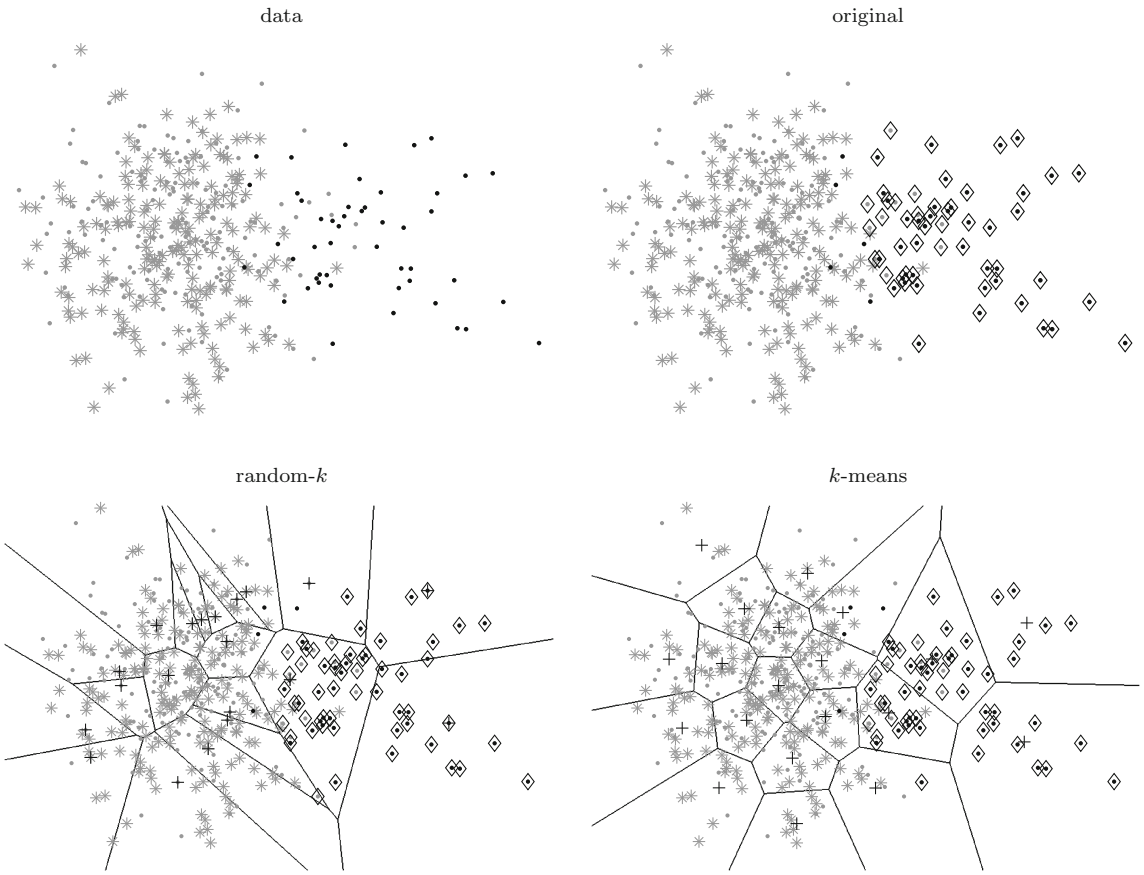

Fig. 2 Illustration of the proposed quasi-supervised learning algorithm. Samples in the reference control and the mixed datasets are shown with *asterisk* and *dot* symbols, respectively, with the mixed dataset samples drawn from the target distribution shown with *dark dot* symbols. The mixed dataset samples recognized to have been drawn from the target distribution are shown with *diamond* symbols. The cluster centers in the group formulation methods are shown with *plus* symbols leading to the partitions shown with *solid lines*

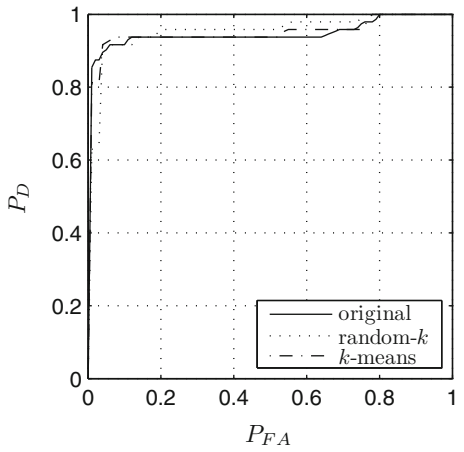

Fig. 3 The receiver operating characteristic curves for the illustration data obtained by the proposed group formulation of the quasi-supervised learning algorithm using random-k and k-means clustering methods as well as the original formulation. The recognition accuracies provided by the group formulation methods are very similar to the one achieved using the original method, slightly surpassing it for larger false alarm rates

a false alarm rate of 5% using the original as well as the proposed algorithm based on random-k and k-means clustering schemes are shown in Fig. 2. The distribution of the samples between the clusters was considerably more uniform in the k-means clustering, while the partitioning by the random-k clustering produced markedly irregular partitions. The effects of these differences on the eventual recognition accuracy, however, appears to be minimal as evidenced by the corresponding receiver operating characteristic curves that follow each other very closely as well as the one achieved by the original method shown in Fig. 3.

3.2 Comparative performance evaluation results

The recognition task in a quasi-supervised learning setting described in Section 2.2 is characterized primarily by the overlap between the distributions $p(x|\mathcal{C}_0)$ and $p(x|\mathcal{C}_1)$ associated with the datasets $\mathcal{C}_0$ and $\mathcal{C}_1$, and the inherent overlap between the underlying reference and target probability distributions $p_r(x)$ and $p_t(x)$. The usual factors

including the dimensionality of the observation space $\mathbf{X}$ and the size of the collection $\{x_i\}$, $x_i \in \mathbf{X}$, $i = 1, 2, \ldots, \ell$ contribute to the overall difficulty in a secondary capacity.

To evaluate the proposed algorithm in terms of the computational expense as well as the recognition performance, we carried out a series of experiments on synthetic datasets of differing difficulty. In all instances, the underlying reference and target distributions were represented by multivariate Gaussian functions with identity covariance matrices $I_{D\times D}$, differing only in their means such that while $p_r(x) \sim \mathcal{N}([0\ 0\ldots0]^T, I_{D\times D})$, $\quad p_t(x) \sim \mathcal{N}([3\ 0\ldots0]^T, I_{D\times D})$ in $\mathbf{X} = \mathbb{R}^D$. The experiments entailed collecting a set of $\ell/2$ samples drawn from $p_r(x)$ into $\mathcal{C}_0$ and another set of $\ell/2$ samples drawn from $\lambda p_r(x) + (1-\lambda)p_t(x)$ into $\mathcal{C}_1$. Following the original quasi-supervised learning algorithm, the proposed algorithm was carried out for varying number of clusters K established using the random-k and the k-means clustering schemes. The computational expense recorded the calculation time of the posterior probability estimates $p_0(x_i)$ and $p_1(x_i)$ for all samples x_i including the minimization of $E(n)$ for the optimal reference set size parameter n. The recognition accuracy was determined in terms of the detection and false alarm rates defined in Eqs. (20) and (21), and the area under the receiver operating characteristic curves generated by the P_D–P_{FA} graphs was computed.

The average computation times of the original and the proposed quasi-supervised learning algorithms for ℓ = 5,000, 7,500, 10,000, 15,000 and 20,000 with $D = 5$, $K = 50$ and $\lambda = 0.50$ are shown in Fig. 4. The savings in computation time achieved by the proposed algorithm are substantial and statistically significant as evidenced by the respective 95 % confidence intervals, especially for larger ℓ. This improvement is certainly due to the smaller number of iterations required to compute the posterior probabilities in the group formulation running no higher than the number of clusters K, as opposed to as high as $\ell - 2n$ in the original formulation. The difference in the computation times observed for the random-k and k-means clustering methods amounts to the extra iterations involved in the latter until convergence to a stationary configuration between the cluster centers and the assignments of the samples into the respective clusters.

The joint plots of the areas under the P_D–P_{FA} graphs and the corresponding computation times by the proposed algorithm using random-k and k-means clustering for varying with K along with those obtained by the original method observed for each combination of $\lambda = 0.50, 0.75, 0.90$ and ℓ = 5,000, 10,000, 20,000 are shown in Fig. 5. Clearly, the proposed algorithm achieves comparable recognition performance in significantly smaller computation times using both clustering strategies. The recognition accuracy improves for larger K at the expense of the computation times.

It is also interesting to note that the proposed algorithm occasionally achieves better recognition performance than the original method. This improvement in recognition performance can be attributed to the averaging effect achieved by the group formulation. In the original formulation, small disturbances on x can change the order of the distances $d(x, x_i)$ and the corresponding binary sequence from which the posterior probabilities are computed, introducing a jitter effect similar to noise in the resulting calculations. The group formulation, however, reduces this jitter by collecting the label data from nearby points into clusters with average labels that achieve a more stable ordering of the sample-to-cluster distances $\rho(x; C_k)$.

3.3 Application to the *N*-glycosylation prediction dataset

In a second comparison experiment, we have applied the proposed quasi-supervised learning algorithm to the *N*-glycosylation prediction dataset studied previously by [15]. Prediction of functional or structural attributes of amino acid sequences is problematic for algorithms requiring absolute examples to train on due to the incomplete and error-prone nature of the accumulated body of structural and functional annotations. First and foremost, existing annotations document only the sites with positively identified attributes, but do not provide a complementary list of sites that are experimentally verified to lack the attribute in question. To make matters worse, the positive identifications themselves can be faulty due to a

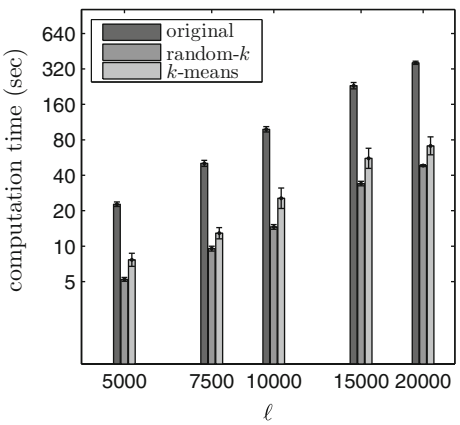

Fig. 4 The average computation times obtained for increasing dataset size ℓ for the original quasi-supervised learning algorithm as well as the group formulations operated with random-k and k-means clustering schemes for $K = 50$ along with the 95 % confidence intervals. Both axes are drawn in a logarithmic scaling. All *plots* represent average computation times over 20 independent repeats

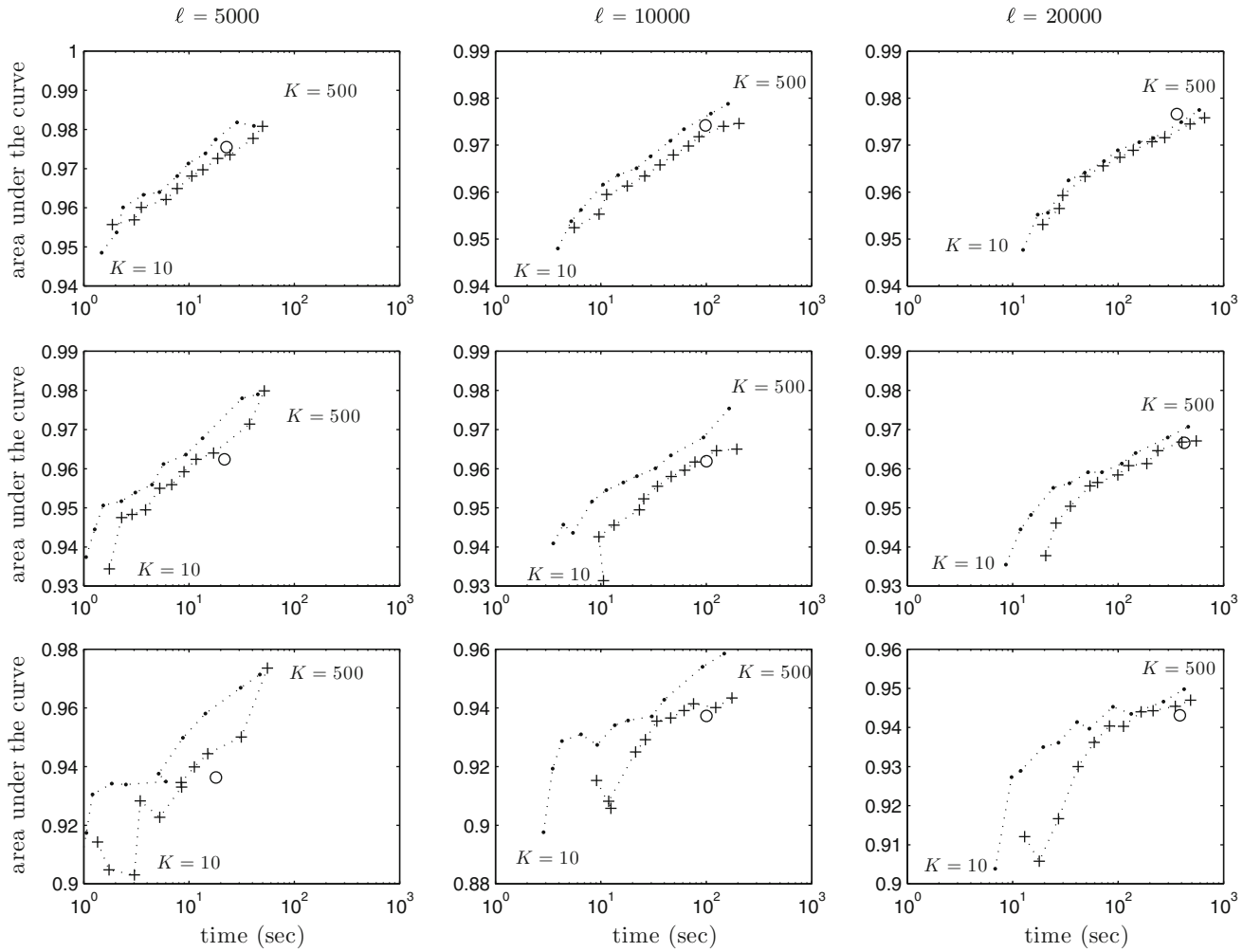

Fig. 5 Recognition performance plotted against the corresponding computation times by the proposed algorithm using random-k (marked by *dot*) and k-means clustering (marked by *plus*) for K = 10, 15, 20, 35, 50, 75, 100, 150, 200, 350, 500, along with those by the original method (marked by *circle*), and for $\lambda = 0.50$ (*upper row*), $\lambda = 0.75$ (*middle row*) and $\lambda = 0.90$ (*lower row*). All values represent averages over 20 independent repeats

variety of factors associated with wet laboratory experimentation, prompting a constant need for revision of the existing annotations in the sequence databases. In that respect, the quasi-supervised learning strategy appears particularly well suited to functional or structural attribute prediction problems in computational biology.

The *N*-glycosylation prediction dataset consists of 55184 motif vectors composed of 150 features that best characterize the physico-chemical composition in the vicinity of the sites possessing the consensus sequon for *N*-glycosylation along the among amino acid sequences of human proteins. The consensus sequon N-X-S/T consists of an asparagine residue followed by any amino acid X other than proline and either a serine or a threonine residue [28, 29]. Among these sites, only 1939 were experimentally verified *N*-glycosylation sites documented in the UniProt Knowledgebase (http://www.uniprot.org/help/uniprotkb) excluding the potential and probable glycosylation annotations. The recognition task, then, is to predict which of the remaining 53,245 consensus sites are most likely to be glycosylated based on their motif vectors, given that an unknown albeit small fraction of the 1,939 true-positive sites are potentially due to erroneous experimental validation.

We applied the proposed quasi-supervised learning algorithm on this dataset using random-k clustering for K = 50, 100, 200, 500, 1,000, 2,000, and recorded the computation time in seconds as well as the separation between the motif vectors of sites annotated for glycosylation and the remaining ones. To evaluate the separation between the vector groups, we have derived the receiver operation characteristics curves in the usual way, by plotting the fraction of annotated sites predicted to be glycosylated against the fraction of non-annotated sites also predicted to be glycosylated for varying prediction threshold P_c, and computing the area under the curve.

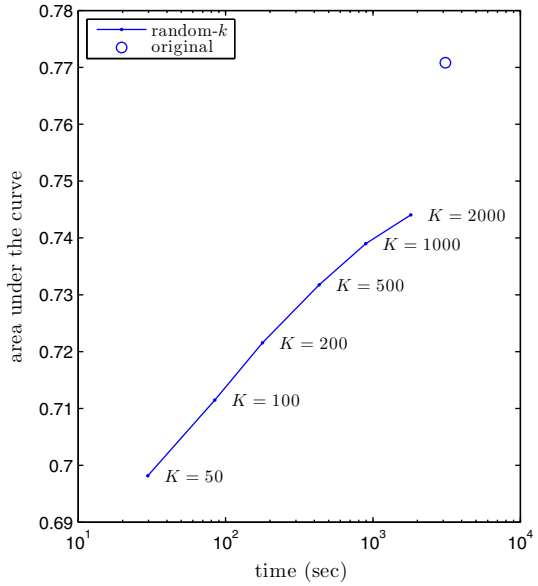

Fig. 6 Comparison of the proposed algorithm for large-scale quasi-supervised learning using random-k clustering to the original algorithm in terms of computation times and separation between the motif vectors of the amino acid sites annotated and non-annotated for N-linked glycosylation. The proposed method achieved learning at smaller computation times, though it trailed the original algorithm in the separation performance

The results are shown in comparison to the original algorithm in Fig. 6 where the areas under the curves are plotted against the computation times in a logarithmic scale as the average values obtained from ten independent runs for each K. The proposed algorithm is clearly superior to the original formulation in terms of the computation times, though this is achieved at the expense of the separation performance. While the separation between the groups improves for larger K, it falls short of the separation achieved by the original algorithm, due potentially to the inherent complexity in the recognition problem. The alternative approach using k-means clustering was omitted from this analysis as the computation times exceeded that of the original algorithm due to the poor convergence in clustering of the motif vector data.

3.4 Application to the MiniBooNE neutrino dataset

The MiniBooNE experiment forms the first stage of the Booster Neutrino Experiment (BooNE) conducted at the Fermi National Accelerator Laboratory (Fermilab, P.O. Box 500, Batavia, IL 60510-5011), with the objective of conclusively confirming or refuting the existence of neutrino oscillations of muon neutrinos into electron neutrinos (http://www-boone.fnal.gov/index.html). The neutrino dataset consists of a total of 130065 instances, with 36499 signal events of electron neutrinos and 93,565 background

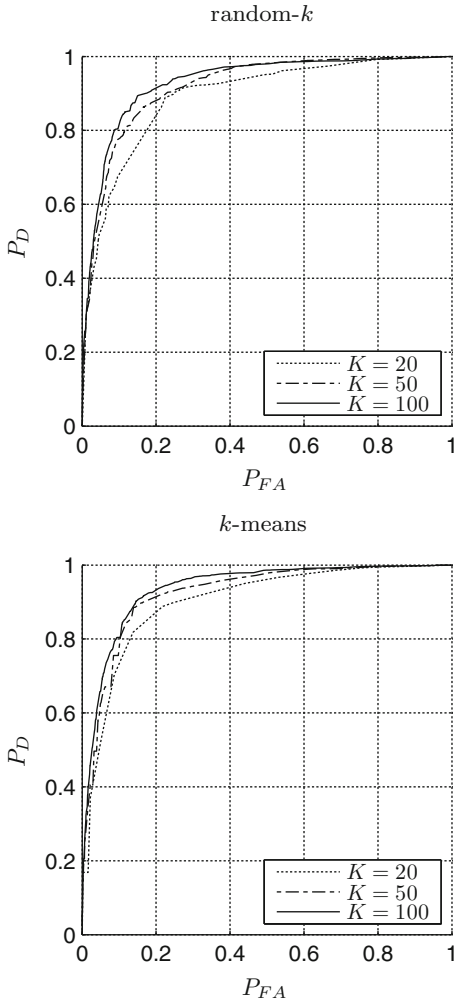

Fig. 7 The receiver operating characteristic curves of the proposed quasi-supervised learning algorithm using random-k and k-means clustering methods on the MiniBooNE data. The recognition performance achieved using the k-means clustering was superior, though the difference grew smaller for larger K. The curves are shown for $K = 20, 50, 100$ only for readability purposes

events of muon neutrinos [24]. Each instance is characterized by 50 attributes such as the event hit multiplicity, energy, and the reconstructed radial position.

To evaluate the labeling performance of the proposed quasi-supervised learning algorithm over this dataset, we have set up a learning experiment with the aim of identifying the signal events in a mixed dataset in contrast to a homogeneous dataset of background events. First, we have removed the outliers characterized by attributes beyond the $[-500, 10{,}000]$ interval, and linearly normalized all attributes across the remaining 129,592 instances with 36,488 signal and 93,104 background events to have unit standard deviation. Next, we have randomly selected 64,796 background events to form the reference dataset $\mathcal{C}_0$ and pooled the remaining background events with all of the signal events into a mixed dataset $\mathcal{C}_1$ of equal size. The

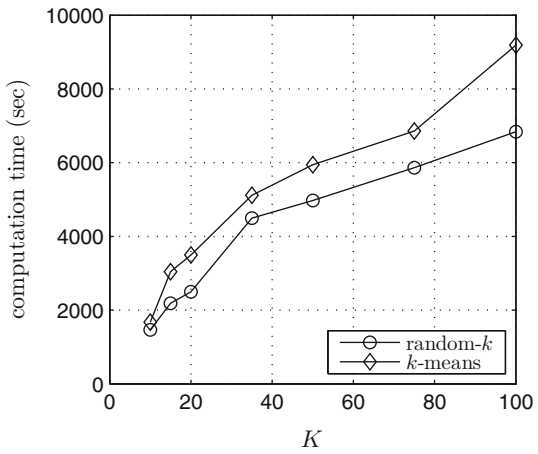

Fig. 8 The computation times of the proposed large-scale quasi-supervised learning algorithm using random-k and k-means clustering methods on the MiniBooNE data. The extra iterations involved in the k-means clustering are responsible for higher computation times

recognition problem then consisted of identifying the signal events in C_1.

We have applied the proposed algorithm to this data using both random-k and k-means clustering methods for K = 10, 15, 20, 35, 50, 75, 100. The receiver operating characteristic curves in terms of the P_D–P_{FA} graphs and the respective computation times are shown in Figs. 7 and 8. The higher recognition performance achieved using the k-means clustering method can be attributed to a more adequate organization of the data among the clusters, though the difference between the two alternatives grows smaller with increasing K. The price for better recognition performance is paid, however, in higher computation times due to the extra iterations involved in the k-means clustering.

4 Conclusion

In this paper, we have introduced a new method to address the quasi-supervised learning problem over large datasets. The proposed method decomposes the expression for the posterior probabilities of the contrasting datasets over sample groups instead of individual samples. This reduces the computational expense incurred in posterior probability calculation and allows large-scale quasi-supervised learning. In the experimental results on synthetic and real datasets, the proposed algorithm operated with random-k and k-means clustering alternatives achieved substantial reduction in computation times for comparable recognition performances. In particular, the results on the MiniBooNE neutrino dataset confirmed the viability of quasi-supervised learning on datasets containing over 100000 samples using the proposed algorithm. It should also be pointed out that the nature of the proposed algorithm is very suitable for parallel computation, and the computational load incurred during the calculation of the posterior probabilities can be readily distributed among multiple cores to obtain further reductions in the computation time.

Experimental results also revealed that the group formulation can also improve the recognition performance in addition to reducing the computational expense. This can be attributed to a regularizing effect of the group formulation that arbitrates the antagonistic effects of nearby points of opposing datasets to the estimated probability. While a change in the proximity order of samples alters the resulting estimate in the original formulation, the group formulation is, to a certain extent, immune to such small perturbations in the data, as nearby points tend to be clustered together.

Among the two clustering schemes evaluated here, the random-k clustering is the simplest and the quickest one, assigning samples to clusters via nearest neighbor classification to a randomly selected collection of k samples. In contrast, the k-means clustering refines this initial grouping by recomputing the cluster centers and reassigning the samples until convergence to a stationary partitioning of the whole dataset. In experiments, a positive effect of this refinement was observed on the recognition performance. This improvement, however, came at the expense of greater computation times due to the extra iterations. Naturally, the group formulation can also be adapted readily to operate on groups established using any other clustering algorithm of choice, as long as the distances between the clusters and the individual samples can be computed within limits of computational feasibility.

On a final note, further improvement in the computational expense can be achieved by expediting the optimization procedure carried out to determine the best reference set size for the ultimate posterior probability estimation. To this end, a variety of numerical optimization methods that require the fewest evaluations of the cost functional can be considered, since each evaluation of the cost functional involves recomputing the posterior probability estimates for the whole dataset. Another strategy would be to limit the posterior probability computations to a smaller, but representative subset of samples. A complete analysis in this direction must also address the specific relationship between the optimal reference set size and the size of the learning dataset. These avenues of research remain to be explored in future studies.

Acknowledgments This work was supported by the European Union Seventh Framework Programme Marie Curie Action grant PIRG03-GA-2008-230903. The MiniBooNE neutrino dataset was provided by Dr. Byron Roe, Emeritus Professor at the Department of Physics, University of Michigan at Ann Arbor.

References

1. Angelov P, Lughofer E, Zhou X (2008) Evolving fuzzy classifiers using different model architectures. Fuzzy Sets Syst 159(23): 3160–3182
2. Auer P (1997) On learning from multi-instance examples: empirical evaluation of a theoretical approach. In: Proceedings of the fourteenth international conference on machine learning, pp 21–29
3. Blum A, Kalai A (1998) A note on learning from multiple-instance examples. Mach Learn 30:23–29
4. Chapelle O, Schölkopf B, Zien A (2006) Introduction to semi-supervised learning. MIT Press, USA
5. Cortes C, Vapnik VN (1995) Support vector networks. Mach Learn 20(1–2):273–297
6. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. Inform Theory 13(1):21–27
7. Dietterich TG, Lanthrop RH, Lozano-Perez T (1997) Solving the multiple-instance problem with axis-parallel rectangles. Artif Intell 89(1–2):31–71
8. Embrechts P, Kluppelberg C, Mikosch T (2000) Modelling extremal events for insurance and finance, applications of mathematics, vol 33. Springer, Berlin
9. Fukunaga K, Hostetler LD (1975) k-nearest-neighbor bayes risk estimation. IEEE Trans Inform Theory 21(3):285–293
10. Gersho A, Gray RM (1991) Vector quantization and signal compression. The springer international series in engineering and computer science. Springer, Berlin
11. Gray RM (1984) Vector quantization. IEEE ASSP Mag 1(2):4–29
12. Haykin S (2008) Neural networks and learning machines, 3rd edn. Prentice Hall, USA
13. Johnson SC (1967) Hierarchical clustering schemes. Psychometrika 32(3):241–254
14. Karaçalı B (2010) Quasi-supervised learning for biomedical data analysis. Pattern Recognit 43(10):3674–3682
15. Karaçalı B (2012) Hierarchical motif vectors for prediction of functional sites in amino acid sequences using quasi-supervised learning. IEEE/ACM Trans Comput Biol Bioinform 9(5): 1432–1441
16. Karaçalı B, Krim H (2003) Fast minimization of structural risk by nearest neighbor method. IEEE Trans Neural Netw 14(1): 127–137
17. Karaçalı B, Ramanath R, Snyder W (2004) Structural risk minimization-based nearest neighbor classifier. Pattern Recognit Lett 25(1):63–71
18. Kuncheva LI (2000) Fuzzy classifier design. Springer, Berlin
19. Lughofer E (2012) Single-pass active learning with conflict and ignorance. Evol Syst 3(4):251–271
20. Lughofer E, Buchtala O (2013) Reliable all-pairs evolving fuzzy classifiers. IEEE Trans Fuzzy Syst 21(4):625–641
21. McLachlan GJ (2004) Discriminant analysis and statistical pattern recognition., Wiley series in probability and statisticsWiley-Interscience, USA
22. Murtagh F (1983) A survey of recent advances in hierarchical clustering algorithms. Comput J 26(4):354–359
23. Olson CF (1995) Parallel algorithms for hierarchical clustering. Parallel Comput 21(8):1313–1325
24. Roe BP, Yang HJ, Zhu J, Liu Y, Stancu I, McGregor G (2005) Boosted decision trees as an alternative to artificial neural networks for particle identification. Nucl Instrum Methods Phys Res Sect A Accel Spectrom Detect Assoc Equip 543(2–3):577–584
25. Vapnik V (1998) Statistical learning theory. Wiley, USA
26. Vapnik V (2006a) Estimation of dependences based on empirical data. Information science and statistics. Springer, Berlin
27. Vapnik V (2006b) Transductive inference and semi-supervised learning. In: Chapelle O, Schölkopf B, Zien A (eds) Semi-supervised learning, chap 24. MIT Press, USA, pp 453–472
28. Varki A, Cummings RD, Esko JD, Freeze HH, Hart GW, Etzler ME (2008) Essentials of glycobiology. Cold Spring Harbor Laboratory Press, USA
29. Weerapana E, Imperiali B (2006) Asparagine-linked protein glycosylation: from eukaryotic to prokaryotic systems. Glycobiology 16(6):91R–101R