*Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 4, 455–464*
DOI: 10.2478/v10006-008-0040-9

amcs

# RANDOM PROJECTION RBF NETS FOR MULTIDIMENSIONAL DENSITY ESTIMATION

EWA SKUBALSKA-RAFAJŁOWICZ

Institute of Computer Engineering, Automation and Robotics
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50–370 Wrocław, Poland
e-mail: ewa.rafajlowicz@pwr.wroc.pl

The dimensionality and the amount of data that need to be processed when intensive data streams are observed grow rapidly together with the development of sensors arrays, CCD and CMOS cameras and other devices. The aim of this paper is to propose an approach to dimensionality reduction as a first stage of training RBF nets. As a vehicle for presenting the ideas, the problem of estimating multivariate probability densities is chosen. The linear projection method is briefly surveyed. Using random projections as the first (additional) layer, we are able to reduce the dimensionality of input data. Bounds on the accuracy of RBF nets equipped with a random projection layer in comparison to RBF nets without dimensionality reduction are established. Finally, the results of simulations concerning multidimensional density estimation are briefly reported.

**Keywords:** radial basis functions, multivariate density estimation, dimension reduction, normal random projection, novelty detection.

## 1. Introduction

Radial Basis Function (RBF) nets have proved their flexibility in a large number of tasks. By trying to apply them for processing multivariate data, which form an intensive stream, we are faced with the well-known curse of dimensionality, which is magnified by the fact that multivariate data must be processed on-line. The aim of this paper is to propose an extension of the RBF net architecture which is intended to be more efficient than the classical RBF net architectures. The idea is to add a new dimensionality reduction layer. In opposition to multi-layer RBF nets, in which all layers are nonlinear, the proposed layer is linear in input variables. Furthermore, weights of this layer are chosen in a random way, instead of using a kind of training process. At first glance, this can be surprising, but recent results on random projections (see the bibliography cited in Section 3) provide tools for dimensionality reduction, which retain (with prescribed accuracy and probability) the Euclidean distances between pairs of projected points. Using random projections as the first layer, we are able to reduce the dimensionality of the input data, deteriorating their metric relationships only slightly. As a consequence, the RBF layer has also a reduced dimensionality and it requires fewer observations in the trainig phase, while in the operating stage the net works much faster. The above ideas are shown in greater detail for the task of estimating multivariate probability densities. This task was selected for several reasons, which are explained below:

1. RBF nets are well suited for density estimation problems and their applications in this area have a relatively long history. The dimensionality reduction allows extending possible applications.
2. Multivariate density estimation is part of many important tasks, including pattern recognition, nonparametric regression estimation and novelty detection (see the fundamental monographs (Devroye and Györfi, 1985; Devroye *et al.*, 1996).
3. We put emphasis on novelty detection tasks, since—jointly with RBF nets—random projections are expected to allow on-line monitoring of intensive data streams, arising, e.g., in quality control by industrial cameras.

**1.1. Introductory remarks on novelty detection.** In many industrial applications, it has become more and more important to monitor the behaviour of complex systems using multivariate measurements. The dimensionality and the amount of data that need to be processed when

intensive data streams are observed grow rapidly together with the development of sensors arrays, CCD and CMOS cameras and other devices.

Many approaches which have been proposed for fault detection in complex systems (Willsky, 1976; Patton, 1994; Gertler, 1998; Patton *et al.*, 2000; Korbicz *et al.*, 2004) require the availability of a precise model of the system under diagnosis. Unfortunately, it is commonly recognized that the model-based approach is often very sensitive to modelling errors and disturbances acting on the system under consideration. Moreover, the time needed to validate a reliable plant model is often too long for practical applications.

Feature-based or pattern recognition approaches need no physical process model. System knowledge is assumed to be contained in a training set composed of measurement vectors and associated operating conditions. This approach can be regarded as a data modelling approach. Large data sets (multidimensional time-series) are obtained during process monitoring and they are used for non-parametric probability data density estimation.

In a model-based approach the overall data obtained are used to build the precise predictive model of the process under consideration. Non-parametric and semi-parametric approaches, as well as neural network and fuzzy modelling methods, can be used for reducing our need for a physical process model. Nevertheless, these approaches need very careful model tuning and some kind of regularization is necessary to avoid the overfitting phenomenon.

We concentrate on using an RBF neural network for non-parametric density estimation directed to data analysis. In this approach the overall data co-occurrence in a chosen time interval is modelled. Unusual data (with low probability density) indicate that some changes in the process occurred. The process, which is often called novelty detection (Bishop, 1994; Roberts, 2000), can be exploited in two different situations. One of them appears when it is required not only to classify known 'normal' and 'fault' input vectors, but also to recognize that a particular input is neither 'normal' nor a member of one of the existing fault categories (Li *et al.*, 2002). This approach leads to pattern recognition methods (Leonard and Kramer, 1991; Leonard and Kramer, 1990).

The second one is based only on positive (normal) data examples. In such a case, "novelty", i.e., abnormal behaviour, indicates that a process is under the influence of special causes, and possibly a faulty situation occurs. Thus, a model of normality is learnt by including only normal examples in the training data; abnormalities are then identified by testing for a novelty against this description. The number of input variables, as well as a long horizon of the observation, which are taken into account during the diagnosis process result in large and very large data dimensionalities.

**1.2. Novelty detection in process diagnosis and statistical process control.** Novelty detection is a data-based approach that can achieve an anomalous detection while only requiring nominal (no-fault) conditions for learning.

The interpretation of novelty detection, understood as the recognition of abnormal patterns, is well established in Statistical Process Control (SPC). Novelty detection is the task of observing changes in the state in a process. For novelty detection, a description of normality is learnt which fits a model to the set of normal examples. Previously unseen patterns are then tested by comparing their novelty scores (as defined by the model) against some threshold. Statistical control charts are designed in order to detect abnormalities (out-of-control states) in the process under consideration. The most common abnormalities are mean shifts, variance changes and trends.

Suppose that $X_1, X_2, \ldots$ are independent random vectors observed sequentially and $X_1$ to $X_{q-1}$ have a distribution function with a probability density $f_0$ while $X_q, X_{q+1}, \ldots$ have a distribution function with a probability density $f_1 \neq f_0$. Here $q$ is unknown and some action should be taken after an undesirable change in the process.

Given observations $X_t = (x_{t1}, \ldots, x_{td})$, one has to decide whether $X_t$ is a random variable $q$ with pdf $f_0$ , i.e., the process is normal, i.e. "in-control", or if $X_t$ is another random variable, the process is "out-of-control", i.e., changes in the process occurred. We assume that probability densities $f_0$ and $f_1$ exist but are unknown.

In other words, the unconditional probability density of an input vector decides whether $X_t$ is novel. All data recognized as "out-of-control" must indicate that their probability density values are below a novelty threshold.

It should be emphasized that most of the known neural network models designed for detecting changes in (mostly univariate) statistical processes work in pattern recognition settings, i.e., they rely on the assumption that also abnormal observations (out-of control states) are available and their class-memberships (in-control and out-of-control labels) are known (Guh, 2005).

A neural network-based approach used when only in-control data are available has been considered in a few papers only (Skubalska-Rafajłowicz, 2006a; Skubalska-Rafajłowicz, 2000; Zorriassatine *et al.*, 2003). This approach consists of two stages. Namely, the density estimation from a training sequence and the selection of a novelty threshold. If a current observation is below this threshold, then it is highly probable that the observation comes from a probability distribution, which is different from that describing a typical (in-control) state. In this paper we concentrate on the density estimation, which is the main stage.

The paper is organized as follows: RBF nets for density estimation are briefly introduced in the next section. Then, we provide basic facts concerning random projec-

tions, which are used for stating a result on the accuracy of RBF nets equipped with a random projection layer in comparison with RBF nets without dimensionality reduction. Finally, the results of simulations are briefly reported.

## 2. RBF neural network models

Radial basis function networks originated from the multidimensional interpolation model (Broomhead, 1988; Buhman, 2003; Powell, 1987). A radial basis function network can be described as a parameterized model used to approximate an arbitrary function by means of a linear combination of basis functions. RBF networks belong to the class of kernel function networks where the inputs to the model are passed through kernel functions which limit the response of the network to a local region in the input space for each kernel or basis function.

Radial basis function neural networks (Bishop, 1995; Haykin, 1999; Krzyżak and Niemann, 2001; Yee and Haykin, 2001) have been widely used in classification problems such as speech recognition, medical diagnosis, handwriting recognition, image processing, and fault diagnosis.

The basic radial basis function network consists of three layers having entirely different roles: an input layer, a hidden layer, which applies a nonlinear transformation from the input space to the hidden space, and a linear output layer. Hence

$$f_N(x) = \sum_{i=1}^{N} w_i G(\|x - c_i\|) \,, \tag{1}$$

where $x \in \mathbb{R}^d$ and $c_i \in \mathbb{R}^d$ are tunable vectors, $w_i$ are tunable weights, and $N$ is the number of neurons.

Usually $\|x\|$ is the Euclidean norm. However, also a generalized weighted norm $\|x|\_Q_i$, defined by the quadratic form $\|x\|^2_{Q_i} = x^T Q_i^T Q_i x$, can be used, where the $Q_i$s are (usually tunable) $d \times d$ matrices.

The most popular choice for the non-linearity $G$ is the Gaussian one. Typically, RBF networks use memory-based learning for their design. Specifically, learning is viewed as a curve-fitting problem in a high-dimensional space (Broomhead, 1988; Poggio and Girosi, 1990). RBF networks can be used to provide an effective and computationally efficient solution to the interpolation and to the approximation problems.

RBF networks are related to Parzen window (Parzen, 1962) estimators of a probability density (Specht, 1990; Schlorer and Hartman, 1992; Rafajłowicz, 2006; Skubalska-Rafajłowicz, 2006a) or to Nadaraya-Watson regression estimators (Bishop, 1994; Xu *et al.*, 1994; Krzyżak, 1996; Krzyżak and Skubalska-Rafajłowicz, 2004; Yee and Haykin, 2001). Similarities between the RBF network structure and kernel regression estimators lead to RBF networks with the centres chosen to be a subset of the training input vectors and associated weights which directly correspond to the responses at the centres (Krzyżak, 2001).

Generally speaking, training an RBF network consists in determining the number of basis functions (hidden units), centres and widths of each basis function, and output layer weights. For some algorithms, these steps are carried out separately, while in others, all parameters are found simultaneously. Furthermore, different techniques can be mixed and matched for training different parameters.

The existing training strategies for RBF neural networks include the following: strategies selecting radial basis function centres randomly from the training data (Broomhead, 1988), strategies employing unsupervised procedures for selecting radial basis function centres (Chen *et al.*, 1991; Holmström and Hamalainen, 1993; Moody and Darken, 1989), strategies employing supervised selection of centres for selecting radial basis function centres (Karayiannis, 1999; Poggio and Girosi, 1990; Wettschereck and Dietterich, 1992) and regularized interpolation exploiting the connection between an RBF network and the Watson-Nadaraya regression kernel (Yee and Haykin, 2001).

Chen, Cowan, and Grant (1991) derived a systematic method of training radial basis functions in a one-stage approach. They proposed that choosing the RBF centres can be likened to subset model selection where the aim is to choose a subset of centres from a larger set of candidates. More specifically, they suggested that an orthogonal least squares method can be employed as a forward regression procedure by treating the centres as the regressors. The initial set may be the total set of data points or some larger set of centres obtained by some means.

One of the simplest procedures for selecting the centres for radial basis functions is based on the notion of using one centre for each data point to be approximated. For small data sets, this method is reasonable, but clearly it is not suitable for larger data sets.

A relatively simple method for choosing the centres is to randomly sample the data and use the sampled data as centres. By sufficiently over-sampling the input space, good performance may be obtained.

To determine the centres, Moody and Darken proposed the $k$-means clustering algorithm (Moody and Darken, 1989). In this case, the data is clustered into $k$ regions and the centers are determined as the Euclidean centers of each cluster of data. The widths of each basis function can be determined by using a $k$-nearest-neighbor algorithm. Poggio and Girosi (1990) also proposed that Kohonen's self-organizing feature map (Holmström and Hamalainen, 1993) can be used for initializing the radial basis function centres before gradient descent is used to adjust all of the free parameters of the network. Rafa-

jłowicz and Skubalska-Rafajłowicz (2003) suggest that the centres should be selected in a data driven way from equidistributed (or quasi-random) points, which are deterministic sequences having properties of uniformly distributed random variables.

**2.1. RBF neural network models for density estimation and novelty detection.** The Parzen density estimator (2) cannot be directly applied to intensive data streams, since the number of summands would be prohibitively large. Therefore, as a vehicle for presenting RBF nets with random projections we shall use nets with reduced numbers of centers, which are trained in a relatively simple way, well suited for intensive data streams (Skubalska-Rafajłowicz, 2006a). In Section 4 we investigate the estimation accuracy for this class of RBF nets. Below, we summarize the way of their learning (Skubalska-Rafajłowicz, 2006a; Skubalska-Rafajłowicz, 2006b).

The architecture of RBF networks related to Parzen window estimators of the probability density is simple and consists of one hidden layer with kernel units and an output layer. The kernel functions of the hidden units are usually taken as Gaussian functions:

$$G(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \text{ for some } \sigma > 0 \text{ and } r \in \mathbb{R}\,.$$

These neural networks estimate the multidimensional probability density function as a sum of kernel functions. The number of units in the hidden layer is equal to the sample size. The parameters of the network (1) are obtained from an $n$-sample observation data set (training sequence) $L_n = ((X_1, Y_1), \ldots, (X_n, Y_n)$. As regards the probability density estimation, labels $Y_i, i = 1, \ldots, n$ are all set as 1 and do not carry any additional information. In the context of the pattern recognition problems, we can treat the whole learning sequence as belonging to the same class. When density estimation is applied for novelty detection, then the case that all $Y_i = 1$ corresponds to the lack of novel observations in the training sequence.

Thus, the network with Gaussian kernel functions takes the following form:

$$\frac{1}{n(2\pi\sigma^2)^{d/2}} \sum_{i=1}^{n} \exp\left(-\frac{||X - X_i||^2}{2\sigma^2}\right)\,, \qquad (2)$$

where $(n(2\pi\sigma^2)^{d/2})^{-1}$ is the normalizing factor. For the sake of simplicity, this factor will be neglected in the rest of the paper.

Let $N$ be the number of centres. Assuming that the centres should be distributed according to the same probability distribution as the training data, the centres are simply a subset of the training input vectors. One can take, e.g., first $N$ elements from the training sequence $(X_1, \ldots, X_n)$.

Note that if $X_i$ is close to a centre $C$, then

$$G(||X - X_i||) - G(||X - C||) \approx 0.$$

So, we can replace each $X_j$ in the sum (2) by its nearest neighbour among a set of centres $\{C_1, C_2, \ldots, C_N\}$ breaking ties at random. Note that the same $C_i$ can be the nearest neighbour for several $X_j$s and that each $C_i$ has at least one point from the training sequence (namely, itself) as a neighbour, since every centre is taken from the training set.

Let $n_j$ stand for the number of points closest to the centre $C_j$, i.e.,

$$n_j = \text{card}[\{X_i : ||X_i - C_j|| < ||X_i - C_k||\}].$$

Thus, we obtain the approximate version of (2):

$$y(X) = \frac{1}{n(2\pi\sigma^2)^{d/2}} \sum_{j=1}^{N} n_j \exp\left(-\frac{||X - C_j||^2}{2\sigma^2}\right). \qquad (3)$$

One can choose any known method, e.g., the cross-validation, for selecting $\sigma$. Having selected centres and using (3), one can considerably reduce the computational burden needed for selecting $\sigma$ in a data-driven way using the least squares cross-validation (Bowman, 1984; Jones *et al.*, 1996). The method is based on representing the Integrated Squared Error (ISE) as

$$\begin{aligned} &ISE(\sigma) \\ &\quad = \int (y(X,\sigma) - f(X))^2 \\ &\quad = \int (y(X,\sigma))^2 - 2\int y(X,\sigma) f(X) + \int (f(X))^2, \end{aligned} \qquad (4)$$

where $f(X)$ is the unknown probability density function. The minimizer of the ISE is the same as the minimizer of the first two terms of the final form. The second term is estimated by $-2/n \sum_{i=1}^{n} y_i(X_i, \sigma)$, where $y_i(X_i, \sigma)$ is the "leave-one-out" kernel density estimator defined using the data with $X_i$ removed. The minimizer is taken as a width parameter $\sigma$ of the RBF network. The simplified form of the cross-validation procedure can be performed on selected centres only, leaving the weights unchanged.

Clearly, one can use other classes of neural networks, (Chen *et al.*, 2004; Magdon-Ismail and Atiya, 2002; Yin and Allinson, 2001) or other training algorithms of RBF nets for the problem considered in this paper, but—to the best of the author's knowledge—they would require much greater computational efforts.

## 3. Linear random projections

Here we focus our attention on the high dimensionality of the probability density estimation problem. Reducing the

dimension of the feature vectors using linear random projection to enhance the performance of the proposed density estimation method is examined as a remedy to the large data dimensionality.

In normal random projections (Vempala, 2004), we can estimate the original pairwise Euclidean distances directly using the corresponding Euclidean distances in a smaller dimension. Furthermore, the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984; Dasgupta and Gupta, 2003) provides the performance guarantee.

We give a review of normal linear random projections (Achlioptas, 2001; Ailon and Chazelle, 2006; Arriaga and Vempala, 1999; Dasgupta and Gupta, 2003; Frankl and Maehara, 1987; Indyk and Motwani, 1998; Johnson and Lindenstrauss, 1984).

Let $u_i \in \mathbb{R}^d,\ i = 1, \ldots, n$ be the original data. Let $S \in \mathbb{R}^{k \times d}$ be a random matrix whose entries are i.i.d. samples of some random variable. The projected data are

$$v_i = Su_i \in \mathbb{R}^k, \quad i = 1, \ldots, n.$$

Note that for $m, l \in \{1, \ldots, n\}$ we have

$$v_{mj} - v_{lj} = \sum_{i=1}^{d} s_{ji}(u_{mi} - u_{li}), \quad j = 1, 2, \ldots, k.$$

When $s_{ij} \sim N(0, 1)$ are independent, identically distributed (i.i.d.), then

$$v_{mj} - v_{lj} = \sum_{i=1}^{d} s_{ji}(u_{mi} - u_{li}) \sim N(0, \sum_{i=1}^{d}(u_{mi} - u_{li})^2),$$

and

$$X_j = \frac{v_{mj} - v_{lj}}{\left(\sum\limits_{i=1}^{d}(u_{mi} - u_{li})^2\right)^{1/2}} \sim N(0, 1). \tag{5}$$

Denote by $\| \cdot \|_E$ the Euclidean distance. Then, we can estimate

$$d_E^2(u_m, u_l) = ||u_m - u_l||_E^2 = \sum_{i=1}^{d}(u_{mi} - u_{li})^2$$

from the sample squared distances (obtained after projections onto $k$ random directions, defined by rows of $S$) as follows:

$$\hat{d}_E^2 = \frac{1}{k}\sum_{j=1}^{k}(v_{mj} - v_{lj})^2. \tag{6}$$

We skip displaying the dependence of $\hat{d}_E^2$ on $m$, $l$ for the sake of brevity. Note that $\hat{d}_E^2$ is estimated using the elements of projected vectors. Below, we provide results, which show to what extent the distances between projected points are close to distances of their counterparts in the original space.

It is easy to show that (Dasgupta and Gupta, 2003; Li *et al.*, 2007; Vempala, 2004):

$$E\{\hat{d}_E^2\} = d_E^2(u_m, u_l). \tag{7}$$

Thus, $\hat{d}_E^2$ is an unbiased estimator of the "true" distance between points in the higher dimensional space, while its variance

$$\mathrm{var}(\hat{d}_E^2) = \frac{2}{k} d_E^4(u_m, u_l) \tag{8}$$

decreases to zero as $k \to \infty$. Furthermore, according to (5), we have

$$\frac{k\hat{d}_E^2}{d_E^2(u_m, u_l)} = \sum_{i=j}^{k} X_j^2 \sim \chi_k^2,$$

where $\sum_{i=j}^{k} X_j^2 \sim \chi_k^2$ means that the sum has the chi-squared distribution with $k$ degrees of freedom. Thus, using chi-squared tail Chernoff bounds (see (Dasgupta and Gupta, 2003) for details), we can obtain the bound on the probability that the relative error exceeds $\varepsilon$ $(1 > \varepsilon > 0)$:

$$\Pr\left\{\frac{|\hat{d}_E^2 - d_E^2(u_m, u_l)|}{d_E^2(u_m, u_l)} \geq \varepsilon\right\} \leq 2\,\exp\Big(-\frac{k}{4}\varepsilon^2 + \frac{k}{6}\varepsilon^3\Big). \tag{9}$$

In order to provide more explicit bounds, select the admissible probability of error $\delta > 0$. Since there are in total $n(n-1)/2 < n^2/2$ pairs among $n$ data points, by the union bound inequality, it suffices that

$$\frac{n^2}{2}\Pr\left\{|\hat{d}_E^2 - d_E^2(u_m, u_l)| \geq \varepsilon d_E^2(u_m, u_l)\right\} \leq \delta.$$

Using (9), we obtain

$$\frac{n^2}{2} 2\exp(-\frac{k}{4}\varepsilon^2 + \frac{k}{6}\varepsilon^3) \leq \delta$$

and, consequently,

$$k \geq \frac{2\log n - \log\delta}{\varepsilon^2/4 - \varepsilon^3/6} = c(n, \varepsilon, \delta)$$

provides the required upper bound for the probability of error.

It should be mentioned that one can also sample $s_{ij}$ from other distributions with zero mean and unit variance (Achlioptas, 2001; Indyk and Naor, 2006; Li *et al.*, 2006).

The above inequalities are bounds for the probabilities of deviations between distances of pairs of points in the original space and in the projection space, which has a reduced dimension.

**3.1. Some experiments on distance preservation after random projections.** It is of interest to grasp an experience on the behaviour of particular distances. To this end, the following simulations were performed:
(i) Firstly, $n = 1001$ points $u_i \in \mathbb{R}^{100}$ were generated with independent elements, which were drawn from the Gaussian distribution. It would be very time consuming to calculate the distances between all 1000 points. Therefore, an additional point, numbered as 1001, was drawn and the distances reported below were calculated between this point and all remaining 1000 points.
(ii) Then a $49 \times 100$ random matrix $S$ were generated with its entries drawn independently from the Gaussian distribution with zero mean and dispersion $1/7$.
(iii) The dimensionality reduction was done as $v_i = S\,u_i$.
(iv) The distances between the first points and the rest of them were calculated, i.e., $\Delta_u(i) \stackrel{\text{def}}{=} ||u_i - u_1||$ and $\Delta_v(i) \stackrel{\text{def}}{=} ||v_i - v_1||$, $i = 2, 3, \ldots, 1001$.

Pairs $(\Delta_u(i)\,\Delta_v(i))$, $i = 2, 3, \ldots, 1001$ are shown as dots in Fig. 1 (left panel). In an ideal situation, when projections maintain exactly the distances of their parents, all dots should be located along the line bisecting the first and the third quadrant. Departures from that line illustrate departures in preserving distances. Taking into account that we simulated a 100-dimensional space, the observed departures are of a small or moderate size. Note that departures should be considered in the context of the dimensions of the corresponding spaces. This can be achieved by scaling the axes by the square roots of the dimensions, i.e., 10 and 7, respectively. Then, the departure for one component of a vector is of the order of at most $1 - 2$.

The right panel of Fig. 1 illustrates the impact of selecting matrix $S$. Dots in this panel were obtained by repeating Steps (ii)–(iv) of the above simulations, i.e., the same 1001 points were multiplied by another matrix $S$, which was independently generated form the same distribution as above. As one can notice, there are no qualitative changes between positions of points in the left and right panels. Many other simulations, which are not reported here, provide qualitatively the same patterns.

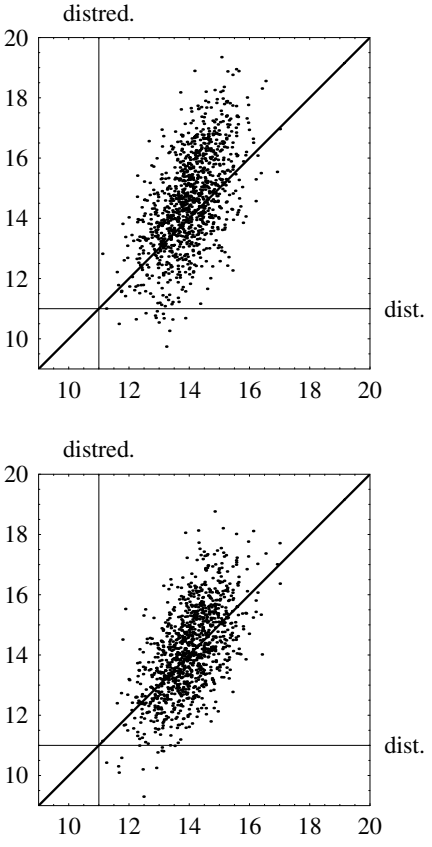


Fig. 1. Departures in distances of pairs of points before and after dimensionality reduction (see the text for explanations).

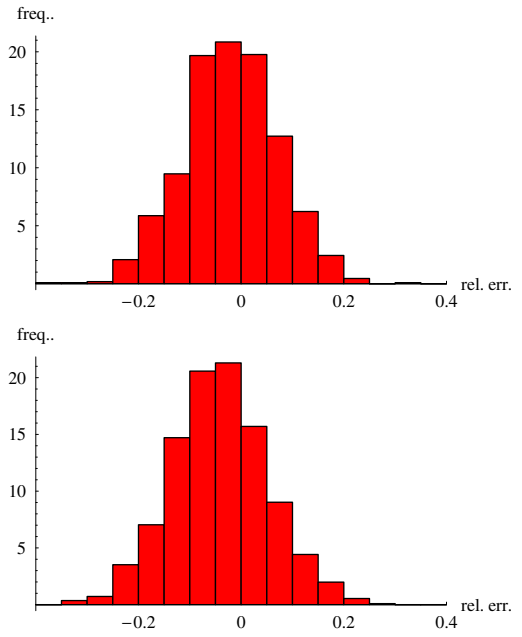


Fig. 2. Frequency histograms (in %) of relative errors in preserving the distances of pairs of points before and after dimensionality reduction (see the text for explanations).

In addition to the above qualitative deliberations, it is expedient to outline relative errors, which are understood as

$$\delta_i \stackrel{\text{def}}{=} \frac{\Delta_u(i) - \Delta_v(i)}{\Delta_u(i)}, \quad i = 2, 3, \ldots, 1001. \tag{10}$$

The basic statistics for the $\delta_i$s are the following:
(a) mean = $-0.0416$,
(b) median = $-0.0426$,
(c) dispersion = $0.094$,
and these values keep their orders when different random

matrices are used. More detailed information about the distribution of the relative errors is presented in Fig. 2, in which the histograms of $\delta_i$s are shown for two random matrices $S$ and the same set of multivariate points. The analysis of these histograms shows that the overall shape and the range of errors changes only sightly when different random matrices are used for dimensionality reduction. The distribution of $\delta_i$s looks very similar to the density of the Gaussian distribution, but this aspect is outside the scope of this paper.

## 4. RBF with input dimension reduced by random normal projection

Random projection could be implemented by an additional network layer with weights establishing some random projection $S$. Thus, we obtain a new Gaussian RBF neural network:

$$\sum_{i=1}^{N} w_i \exp\left(-\frac{||(x-C_i)S||^2}{2\sigma_i^2}\right), \tag{11}$$

where the network parameters ($w_i$ and $C_i$, $i = 1,\ldots,N$, $\sigma$) have been obtained according to any chosen training algorithm, see, e.g., (Haykin, 1999; Krzyżak and Skubalska-Rafajłowicz, 2004; Skubalska-Rafajłowicz, 2006a).

According to (Arriaga and Vempala, 1999; Li *et al.*, 2007), if

$$k \geq \frac{\ln(2)-\ln(\delta)}{\varepsilon^2/4-\varepsilon^3/6},$$

then with the probability of at least $1-\delta$ the squared Euclidean distance between a pair of data points can be approximated with the relative accuracy of at least $1\pm\varepsilon$, using the squared Euclidean distance of the projected data after normal random projections $S$. These facts allow us to formulate the following result.

**Theorem 1.** *For an arbitrary but fixed $x \in \mathbb{R}^d$, define*

$$A_{\max} = \max_{1\leq i\leq n} \|(x-C_i)\|^2/(2\sigma^2).$$

*($A_{\max}$ depends on $x$, but we skip this for brevity.)*

1. Select $\varepsilon > 0$, which is such that $\varepsilon\, A_{\max} < 1$. If

$$k \geq \frac{\ln(2\,N)-\ln(\delta)}{\varepsilon^2/4-\varepsilon^3/6},$$

then, with the probability of at least $1-\delta$,

$$(1-\varepsilon A_{\max})\, Y_{\mathrm{RBF}}(x) \leq Y_{\mathrm{RBF}}^{\mathrm{proj}}(x)$$

and simultaneously

$$Y_{\mathrm{RBF}}^{\mathrm{proj}}(x) \leq \left(1+\frac{\varepsilon A_{\max}}{1-\varepsilon A_{\max}}\right) Y_{\mathrm{RBF}}(x).$$

2. Select $0<\alpha<1$. If $k$ is chosen such that

$$k \geq \frac{(\ln(2N)-\ln(\delta))\, A_{\max}^2}{\left(\frac{\alpha}{1+\alpha}\right)^2},$$

then, with the probability of at least $1-\delta$,

$$(1-\alpha)\; Y_{\mathrm{RBF}}(x) \leq Y_{\mathrm{RBF}}^{\mathrm{proj}}(x) \leq (1+\alpha)\; Y_{\mathrm{RBF}}(x).$$

*Outline of the proof.* If $k \geq (\ln(2N)-\ln(\delta))(\varepsilon^2/4-\varepsilon^3/6)^{-1}$, then, with the probability of at least $1-\delta$, we have

$$(1-\varepsilon)||x-C_i||^2 \leq ||(x-C_i)S||^2 \leq (1+\varepsilon)||x-C_i||^2\},$$

for $i=1,\,2,\ldots,\,N$, where $x\in\mathbb{R}^d$ is a chosen point. Let

$$Y_{\mathrm{RBF}}(x) = \sum_{i=1}^{N} w_i \exp\left(-\frac{||(x-C_i)||^2}{2\sigma^2}\right).$$

Thus,

$$\begin{aligned}
&Y_{\mathrm{RBF}}^{\mathrm{proj}}(x)\\
&\quad=\sum_{i=1}^{N} w_i \exp\left(-\frac{\|(x-C_i)S\|^2}{2\sigma^2}\right)\\
&\quad\leq\sum_{i=1}^{N} w_i \exp\left((\varepsilon-1)\frac{\|(x-C_i)\|^2}{2\sigma^2}\right)\\
&\quad=\sum_{i=1}^{N} w_i \exp\left(-\frac{\|(x-C_i)\|^2}{2\sigma^2}+\frac{\varepsilon\|(x-C_i)\|^2}{2\sigma^2}\right)\\
&\quad\leq\exp(\varepsilon A_{\max})\sum_{i=1}^{N} w_i \exp\left(-\frac{\|(x-C_i)\|^2}{2\sigma^2}\right)\\
&\quad\leq\left(1+\frac{\varepsilon A_{\max}}{1-\varepsilon A_{\max}}\right) Y_{\mathrm{RBF}}(x).
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&Y_{\mathrm{RBF}}^{\mathrm{proj}}(x)\\
&\quad\geq\sum_{i=1}^{N} w_i \exp\left(-(1+\varepsilon)\frac{\|(x-C_i)\|^2}{2\sigma^2}\right)\\
&\quad=\sum_{i=1}^{N} w_i \exp\left(-\frac{\|(x-C_i)\|^2}{2\sigma^2}\right)\exp\left(-\frac{\varepsilon\|(x-C_i)\|^2}{2\sigma^2}\right)\\
&\quad\geq(1-\varepsilon A_{\max})Y_{\mathrm{RBF}}(x).
\end{aligned}$$

## 5. Simulation studies on density estimation using RBF nets with random projections

The proposed method was tested using 50-D and 100-D normal distributions with mean $(0,0)$ and covariance matrix $I$. We compared the performance of the probability density estimation method based on the RBF network

Table 1. Mean-square error of the probability density estimation by (1) (non-reduced dimension) averaged over 10000 test samples.

| MSE | $d = 50$, $N = 5$ | $d = 50$, $N = 20$ | $d = 100$, $N = 5$ | $d = 100$, $N = 20$ |
|---|---|---|---|---|
| for true dimension | 7.91E−14 | 4.45E−14 | 8.48E−30 | 4.86E−29 |

Table 2. Mean-squared error of the probability density estimation by (11) (reduced dimension) averaged over 10000 test samples.

| MSE for $k$ | $d = 50$, $N = 5$ | $d = 50$, $N = 20$ | $d = 100$, $N = 5$ | $d = 100$, $N = 20$ |
|---|---|---|---|---|
| 2 | 2.75E−3 | 1.10E−3 | 2.44E−3 | 2.52E−3 |
| 9 | 3.07E−7 | 3.07E−7 | 1.36E−10 | 2.39E−10 |
| 16 | 7.37E−10 | 1.07E−10 | 1.59E−14 | 6.11E−14 |
| 36 | 8.24E−14 | 4.51E−14 | 8.48E−30 | 4.86E−29 |
| 49 | | | 8.48E−30 | 4.86E−29 |

(1) and the learning algorithm proposed in (Skubalska-Rafajłowicz, 2006a) with results obtained using the RBF proposed here with the input dimension reduced by random the normal projection (11).

The mean-square error of the probability density estimation by (1) with two different numbers of centres ($N = 5$ and $N = 20$) averaged over 10000 test samples are summarized in Table 1 for further comparisons.

One can ask why the estimation error in a 100-dimensional space is smaller than the one obtained in a 50-dimensional space. An explanation is based on the well-known fact that observations drawn from a multivariate Gaussian distribution have a tendency to locate in a ring surrounding the hill, but somewhat below it. As a result, the vicinity of the hill is almost empty. This phenomenon causes trouble in a precise density estimation near the hill. However, if a global estimation error, such as the mean square one, is used, then areas far from the hill are much larger. In these areas, which contain also the above-mentioned ring, the variability of the estimated density is much smaller and the estimation is more accurate. Additionally, the estimation error was averaged over 10, 000 observations, which (for the reasons explained above) also had a tendency to locate in flatter areas, near the ring.

The mean-square error of the probability density estimation by (11) for two different dimensions, i.e., $d = 50$ and $d = 100$ with two different numbers of centres ($N = 5$ and $N = 20$) and a reduced dimension $k = 2, 9, 16, 36, 49$ averaged over 10000 test samples are presented in Table 2.

The proposed method of probability density estimation based on random projections works amazingly well. It is clear that error bounds ($\alpha$) formulated in Theorem 1 are, at least on average, not violated. Furthermore, it should be mentioned that in each examined case only one random projection was generated and accepted without any prior performance examinations.

Additional simulations, not reported here in detail, revealed that 9 from 10 random projections do not essentially change the estimation accuracy, which was evaluated as above, by averaging 10,000 samples. In these simulation experiments dimensionality reduction was from $d = 100$ to 49.

## 6. Conclusions

The proposed structure of RBF nets equipped with a dimensionality reduction layer provides outputs, which are close to an RBF net without dimensionality reduction with a high probability (see Theorem 1).

The proposed method of probability density estimation is very easy to implement and promising results are obtained using simulated data. Nevertheless, it is obvious that further extensive experiments are needed for the validation of the RBF with input dimension reduced by random normal projection as a tool for novelty detection in multidimensional large data streams.

## Acknowledgement

This work was supported by a grant of the Polish Ministry of Science and Higher Education for the years 2006–2009.

## References

Achlioptas D. (2001). Database friendly random projections, *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Santa Barbara, CA, USA, pp. 274–281.

Ailon N. and Chazelle B. (2006). Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, Seattle, WA, USA, pp. 557–563.

Arriaga R. and Vempala S.(1999). An algorithmic theory of learning: Robust concepts and random projection, *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, New York, NY, USA, pp. 616–623.

Bishop C. M. (1994). Novelty detection and neural-network validation, *IEE Proceedings – Vision Image and Signal Processing,* **141**:217–222.

Bishop C.M. (1995). *Neural Networks for Pattern Recognition,* Oxford University Press, Oxford.

Bowman A.W. (1995). An alternative method of cross-validation for the smoothing of density estimates, *Biometrika* **71**(2): 353–360.

Broomhead D. and Lowe D. (1988). Multivariable functional interpolation and adaptive networks, *Complex Systems* **2**(11): 321–323.

Buhmann M. D. (1988). *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, Cambridge.

Chen S., Cowan C.F.N. and Grant P.M. (1991). Orthogonal least squares learning algorithm for radial basisfunction networks, *IEEE Transactions on Neural Networks* **2**(2): 302–307.

Chen S., Hong X. and Harris C.J. (2004). Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization, *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* **34**(4): 1708–1717.

Dasgupta S. and Gupta A. (2003). An elementary proof of a theorem of Johnson and Lindenstrauss, *Random Structures and Algorithms* **22**(1): 60–65.

Devroye L. and Györfi L. (1985). *Nonparametric Density Estimation. The $L_1$ View.* Wiley, New York, NY.

Devroye L., Györfi L. and Lugosi G. (1996). *Probabilistic Theory of Pattern Recognition,* Springer-Verlag, New York, NY.

Frankl P. and Maehara H. (1987). The Johnson-Lindenstrauss lemma and the sphericity of some graphs, *Journal of Combinatorial Theory A* **44**(3): 355–362.

Gertler J.J. (1998). *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, New York, NY.

Guh R.(2005). A hybrid learning based model for on-line detection and analysis of control chart patterns, *Computers and Industrial Engineering* **49**(1): 35–62.

Holmström L. and Hämäläinen A. (1993). The self-organizing reduced kernel density estimator, *Proceedings of the 1993 IEEE International Conference on Neural Networks,* San Francisco, CA, USA, **1**: 417–421.

Haykin S. (1999). *Neural Networks. A Comprehensive Foundation, 2nd Ed.*, Prentice-Hall, Upper Saddle River, NJ.

Indyk P. and Motwani R. (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality, *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, Dallas, TX, USA, pp. 604–613.

Indyk P. and Naor A.(2006). Nearest neighbor preserving embeddings, *ACM Transactions on Algorithms* (to appear).

Johnson W. B. and Lindenstrauss J. (1984). Extensions of Lipshitz mapping into Hilbert space, *Contemporary Mathematics* **26**: 189–206.

Jones M.C., Marron J.S. and Sheather S.J. (1996). A brief survey of bandwidth selection for density estimation, *Journal of the American Statistical Association* **91**(433): 401–407.

Karayiannis N.B. (1999). Reformulated radial basis neural networks trained by gradient descent, *IEEE Transactions on Neural Networks* **10**(3): 657–671.

Krzyżak A. (2001). Nonlinear function learning using optimal radial basis function networks, *Journal on Nonlinear Analysis* **47**(1): 293–302.

Krzyżak A., Linder T.and Lugosi G. (2001). Nonparametric estimation and classification using radial basis function nets and empirical risk minimization, *IEEE Transactions on Neural Networks* **7**(2): 475–487.

Krzyżak A. and Niemann H. (2001). Convergence and rates of convergence of radial basis functions networks in function learning, *Journal on Nonlinear Analysis* **47**(1): 281–292.

Krzyżak A.and Skubalska-Rafajłowicz E. (2004). Combining space-filling curves and radial basis function networks, *Artificial Intelligence and Soft Computing ICAISC 2004. 7th International conference. Zakopane, Poland, Lecture Notes in Artificial Intelligence,* **3070**: 229–234, Springer-Verlag, Berlin.

Korbicz J., Kościelny J. M., Kowalczuk Z. and Cholewa W. (Eds) (2004). *Fault diagnosis. Models, Artificial Intelligence, Applications,* Springer-Verlag, Berlin.

Leonard J. A., and Kramer M. A. (1990). Classifying process behaviour with neural networks: Strategies for improved training and generalization, *Proceedings of the American Control Conference*, San Diego, CA, USA, pp. 2478–2483.

Leonard J.A., and Kramer M.A. (1991). Radial basis networks for classifying process faults, *IEEE Control Systems Magazine* **11**(3): 31–38.

Li Y., Pont M. J. and Jones N.B. (2002). Improving the performance of radial basis function classifiers in condition monitoring and fault diagnosis applications where 'unknown' faults may occur, *Pattern Recognition Letters* **23**(5): 569–577.

Li P., Hastie T.J. and Church K.W. (2007). Nonlinear Estimators and tail bounds for dimension reduction in l1 using Cauchy random projections, *The Journal of Machine Learning Research* **8**(10): 2497–2532.

Li P., Hastie T.J. and Church K.W. (2006). *Sub-Gaussian random projections*, Technical report, Stanford University.

Magdon-Ismail M. and Atiya A. (2002). Density estimation and random variate generation using multilayer networks, *IEEE Transactions on Neural Networks* **13**(3): 497–520.

Moody J. and Darken C.J. (1989). Fast learning in networks of locally tuned processing units, *Neural Computation* **1**(2): 281–294.

Patton R.J.(1994). Robust model-based fault diagnosis: The state of the art, *Proceedings of the IFAC Symposium on Fault Detection Supervision and Safety of Technical Processes*, Espoo, Finland, pp. 1–24.

Patton R. J., Chen J. and Benkhedda H.(2000). A study on neurofuzzy systems for fault diagnosis, *International Journal of Systems Science* **31**(11): 1441–1448.

Parzen E. (1962 ). On estimation of a probability density function and mode, *Annals of Mathematical Statistics* **33**(3): 1065–1076.

Poggio T. and Girosi F. (1990). Networks for approximation and learning, *Proceedings of the IEEE* **78**(9): 484–1487.

Powell M.J.D. (1987). Radial basis functions for multivariable interpolation: A review, *in* (J.C. Mason, M.G. Cox, Eds.) *Algorithms for Approximation*, Clarendon Press, Oxford, pp. 143—167.

Rafajłowicz E. (2006). RBF nets in fault localization, *8th International Conference on Artificial Intelligence and Soft Computing – ICAISC 2006. Zakopane, Poland*, LNCS, Springer-Verlag, Berlin/Heidelberg, **4029/2006**: 113–122.

Rafajłowicz E., Skubalska- Rafajłowicz E. (2003). RBF nets based on equidistributed points, *Proceedings of 9th IEEE International Conference: Methods and Models in Automation and Robotics MMAR 2003,* Szczecin, Poland, **2**: 921–926.

Roberts S. (2000). Extreme value statistics for novelty detection in biomedical data processing, *IEE Proceedings: Science, Measurement and Technology* **147** (6): 363—367.

Schlorer H. and Hartman U. (1992). Mapping neural networks derived from the Parzen window estimator, *Neural Networks* **5**(6): 903–909.

Skubalska-Rafajłowicz E. (2000). On using space-filling curves and vector quantization for constructing multidimensional control charts, *Proceedings of the 5th on Conference Neural Network and Soft Computing*, Zakopane, Poland, pp. 162–167.

Skubalska-Rafajłowicz E. (2006a). RBF neural network for probability density function estimation and detecting changes in multivariate processes, *8th International Conference: Artificial Intelligence and Soft Computing – ICAISC 2006. Zakopane, Poland,* LNCS, Springer-Verlag, Berlin/Heidelberg **4029/2006**: 133–141.

Skubalska-Rafajłowicz E. (2006b). Self-organizing RBF neural network for probability density function estimation, *Proceedings of the 12th IEEE International Conference on Methods and Models in Automation and Robotics,* Międzyzdroje, Poland, pp. 985–988.

Specht D.F. (1990). Probabilistic neural networks, *Neural Networks* **3**(1): 109–118.

Vempala S. (2004). *The Random Projection Method*, American Mathematical Society, Providence, RI.

Wettschereck D. and Dietterich T. (1992). Improving the performance of radial basis function networks by learning center locations, *in* (B. Spatz, Ed.) *Advances in Neural Information Processing Systems,* Morgan Kaufmann, San Mateo, CA, Vol. 4, pp. 1133–1140.

Willsky A. S. (1976). A survey of design methods for failure detection in dynamic systems, *Automatica* **12**(6): 601–611.

Xu L., Krzyżak A. and Yuille A. (1994). On radial basis function nets and kernel regression: Statistical consistency, convergence rates, and receptive field size, *Neural Networks* **7**(4): 609–628.

Yee P. V. and Haykin S. (2001). *Regularized Radial Basis Function Networks: Theory and Applications,* John Wiley, New York, NY.

Yin H. and Allinson N.M.(2001). Self-organising mixture networks for probability density estimation, *IEEE Transactions on Neural Networks* **12**(2): 405–411.

Zorriassatine F., Tannock J.D.T.and O'Brien C.(2003). Using novelty detection to identify abnormalities caused by mean shifts in bivariate processes, *Computers and Industrial Engineering* **44**(3): 385–408.

Received: 4 December 2007
Revised: 15 May 2008