Egyptian Informatics Journal (2010) **11**, 81–92

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

Support vector machine for diagnosis cancer disease: A comparative study

Nasser H. Sweilam [a,*], **A.A. Tharwat** [b], **N.K. Abdel Moniem** [c]

[a] *Department of Mathematics, Cairo University, Faculty of Science, Giza, Egypt*
[b] *Department of Operation Research and Decision Support, Cairo University, Faculty of Computer & Information, Giza, Egypt*
[c] *Department of Statistics, Cairo University, National Cancer Institute, Giza, Egypt*

Received 5 January 2010; accepted 29 September 2010
Available online 4 November 2010

KEYWORDS

Breast cancer diagnosis mathematical model;
Support vector machine (SVM);
Particle swarm optimization (PSO);
Quantum particle swarm optimization (QPSO);
Quadratic programming (QP);
Least square (LS) method

Abstract Support vector machine has become an increasingly popular tool for machine learning tasks involving classification, regression or novelty detection. Training a support vector machine requires the solution of a very large quadratic programming problem. Traditional optimization methods cannot be directly applied due to memory restrictions. Up to now, several approaches exist for circumventing the above shortcomings and work well. Another learning algorithm, particle swarm optimization, Quantum-behave Particle Swarm for training SVM is introduced. Another approach named least square support vector machine (LSSVM) and active set strategy are introduced. The obtained results by these methods are tested on a breast cancer dataset and compared with the exact solution model problem.

1. Introduction

Cancer is a group of diseases in which cells in the body grow, change, and multiply out of control [1]. Usually, cancer is named after the body part in which it originated; thus, breast cancer refers to the erratic growth of cells that originate in the breast tissue. A group of rapidly dividing cells may form a lump or mass of extra tissue. These masses are called cancer [2].

Cancer can either be cancerous (malignant) or non-cancerous (benign). Malignant tumours penetrate and destroy healthy body tissues, for more details see [3]. Cancer detection has become a significant area of research in pattern recognition community.

This paper intends to exhibit an integrated view of implementing automated diagnostic systems for breast cancer detection, and to classify cancer patients by constructing a

* Corresponding author.
E-mail addresses: n_sweilam@yahoo.com (N.H. Sweilam), assemtharwat@hotmail.com (A.A. Tharwat), nermeen2000@hotmail.com, nermeenka2000@yahoo.com (N.K. Abdel Moniem).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.
doi:10.1016/j.eij.2010.10.005

Production and hosting by Elsevier

non-linear optimal classifier using support vector machine. Because of the importance of making a right decision, The classification accuracies of different training method for SVM, namely particle swarm optimization (PSO) method, quantum particle swarm optimization (QPSO) method, quadratic programming (QP) method, and the modifying learning problem of SVM namely least square SVM (LSSVM) are calculated.

The use of the classifier systems in the medical diagnosis' area is increasing gradually, and there is no doubt that evaluation of data taken from patients and decisions of the experts are the most important factors in diagnosis. However, expert systems and deferent artificial intelligence techniques for classification minimizing possible errors that could be occured because of inexperienced experts, and also provide medical data to be examined in shorter time and more detailed.

Fig. 1 shows the various stages followed for the design of a classification system. As it is apparent from the feedback arrows, these stages are dependent. On the contrary, they are interrelated and, depending on the results, one may go back to redesign earlier stages in order to improve the overall performance.

2. CAD system

The main purpose of pattern recognition in the field of cancer diagnosis is to solve the pattern classification dilemma where a pre-described set of input features is used to determine if a patient has a particular disorder or not. This can help in the process of diagnosis, namely Computer Aided Diagnostic (CAD) systems that used in the classification task where certain features (clinical findings) are used to assign a case to a particular pattern (malignant or benign) which represents a diagnosis.

Therefore, the CAD systems can improve the performance of the physicians in terms of (1) reducing the number of misdiagnosis and (2) reducing the time taken to reach a diagnosis, these are the most two important criteria in the developing process of a CAD system. Other performance measures, such as computational complexity and operational load can be overlooked, if kept in an acceptable level.

Breast cancer may be detected via a careful study of clinical history, physical examination, and imaging with either mammography or ultrasound. However, definitive diagnosis of a breast mass can only be established through fine needle aspiration (FNA) biopsy, core needle biopsy, or excisional biopsy. Among these methods, FNA is the easiest and fastest method of obtaining a breast biopsy, and is effective for women who have fluid-filled cysts. Research works on the Wisconsin Diagnosis Breast Cancer (WDBC) data grew out of the desire to diagnose breast masses accurately based solely on FNA. To improve the accuracy and efficiency of the detection of breast cancer, a number of research projects are focusing on developing methods for computer aided diagnosis (CAD) of breast cancer from FNA, including works on image analysis and computational intelligence [4,5]. In this study we focus on computer aided diagnosis (CAD) of breast cancer from FNA depending on computational intelligence.

3. Support vector machine (SVM): an overview

The SVM proposed by Vapnik [6] has been studied extensively for classification, regression and density estimation.

SVMs attempt to find a hyperplane $w \cdot x + b = 0, x_i \in \mathbb{R}^n$ that separates the data points x_i (meaning that all x_i in a given class are on the same side of the plane), that corresponding to a given decision rule: $g(x) = \text{sign}(w \cdot x + b)$.

The question here is how this plane is determined? SVMs choose the separating hyperplane $w \cdot x + b = 0$ that is furthest away from the data points x_i, that is, that has maximal margin (Fig. 2). The underlying idea is that a hyperplane far from any observed data points should minimize the risk of making wrong decisions when classifying new data. To be precise, in SVMs the distance to the closest data points is maximized.

Suppose l patterns are given and each pattern consists of a pair $\{x_i, y_i\}_{i=1}^N$: a vector $x_i \in \mathbb{R}^n$ and the associated label $y_i \in \{-1, 1\}$. Let $X \subset \mathbb{R}^n$ is the space of patterns, $Y \subset \{-1, 1\}$ is the space of labels. The SVM approach aims to find a classifier of the following form:

$$y(x) = \text{sign}\left[\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right] \tag{1}$$

where α_i are positive real constants and b is a real constant, in general, $K(x_i, x) = \langle \phi(x_i), \phi(x) \rangle$, $\langle \cdot, \cdot \rangle$ represents the inner product operation, and $\phi(x)$ is a nonlinear map from the original space to the high dimensional space.

Assume the data set can be separated by a linear hyperplane in the high dimensional space, and that will cause:

$$y_i\left[w^T \phi(x_i) + b\right] \geq 1, \quad i = 1, \ldots, N \tag{2}$$

In case of such separating hyperplane does not exist, a slack variable ξ is introduced, namely

$$\begin{aligned} &y_i[w^T \phi(x_i) + b] \geq 1 - \xi_i, \quad i = 1, \ldots, N \\ &\xi_i \geq 0, \quad i = 1, \ldots, N \end{aligned} \tag{3}$$

According to the structural risk minimization principle, the risk bound is minimized by the following minimization problem:

$$\min_{w,\xi} J_1(w, \xi) = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i \tag{4}$$

Subject to (3), one constructs the Lagrangian function as follows:

$$L_1(w, b, \xi, \alpha, \beta) = J_1(w, \xi) - \sum_{i=1}^{N} \alpha_i \left\{ y_i[w^T \phi(x_i) + b] - 1 + \xi_i \right\} - \sum_{i=1}^{N} \beta_i \xi_i \tag{5}$$

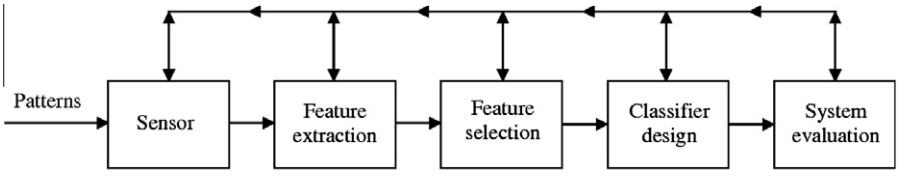

Figure 1 The basic stages involved in the design of a classification system.

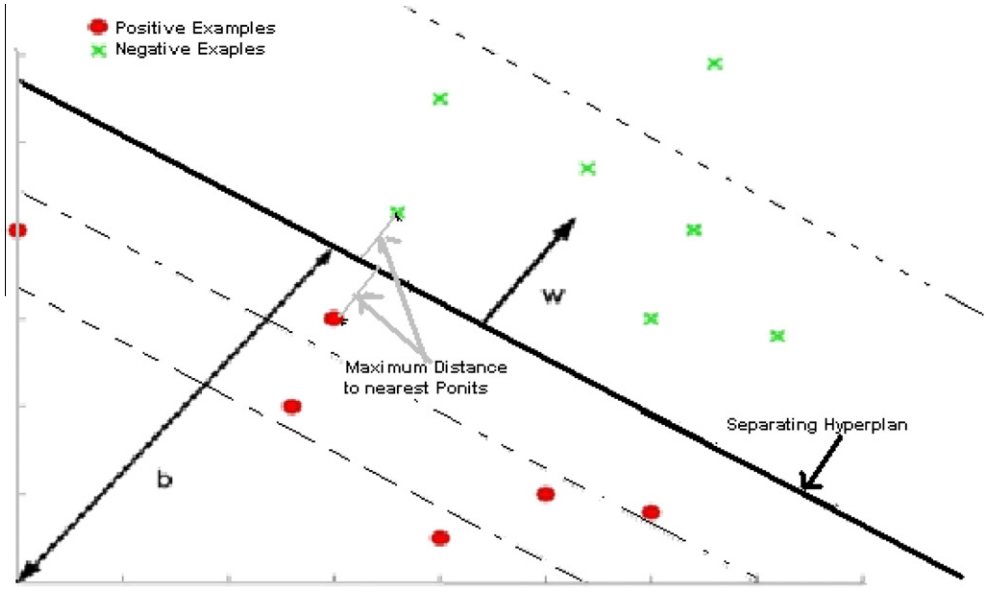

Figure 2 A separating hyperplane (w, b) for a two dimensional training set [5].

where $\alpha_i > 0, \beta_i > 0, (i = 1, \ldots, N)$ are the Lagrangian multipliers of (3). The optimal point will in the saddle point of the Lagrangian function, i.e.

$$\max_{w,\beta} \min_{w,b,\xi} L_1(w, b, \xi, \alpha, \beta) \tag{6}$$

By equating the partial differentiation with zeros, the following equalities will obtained:

$$\begin{aligned} &\frac{\partial L_1}{\partial w} = 0, \quad w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \\ &\frac{\partial L_1}{\partial b} = 0, \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\ &\frac{\partial L_1}{\partial \xi_i} = 0, \quad 0 \le \alpha_i \le c, \quad i = 1, 2, \ldots, N \end{aligned} \tag{7}$$

Substituting (7) in (5), the following quadratic programming (QP) problem will arise:

$$\min_{\alpha} Q_1(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{8}$$

where $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ is called the kernel function (Fig. 3 shows the architecture of SVM with kernel function).

By solving the above QP problem Eq. (8) subject to the given constraints in (7), a hyperplane in the high dimensional space and the classifier in the original space as in (1) are obtained .

4. Survey of training algorithms of SVM

Implementing an SVM learning algorithm requires solving a QP problem. Initially, existing general-purpose quadratic optimization algorithms were applied to solve the SVM problem [6]. For example, quasi-Newton methods such as MINOS [7] or primal–dual interior point methods such as LOQO [8] are applicable for small data sets (1000 s of points). Their advantage is that they are off the shelf and so can be immediately exploited, and they also provide high numerical precision. However, these algorithms are no longer suitable when the kernel matrix (or original data matrix for linear SVM) does not fit in main memory. In order to solve larger problems, special-purpose algorithms have been created that take advantage of unique aspects of the SVM problem. These can be divided into three categories.

4.1. Subset selection algorithms

Subset selection methods sacrifice some precision in the solution (in terms of the Lagrange multipliers α_i) in order to break the optimization problem up into manageable pieces. One optimization approach for SVM, called Chunking [9], relies on the observation that only the support vectors contribute to the final model and other data points are inconsequential to the solution. So in Chunking, an arbitrary subset of the data is first used to generate an SVM solution with a general-purpose QP package. Then only the support vectors are retained and the rest of the data are discarded. Additional data are then added to complete the subset and a new QP solution is determined. This is repeated until the Kuhn–Tucker conditions are met for each data sample. The Chunking approach works as long as the kernel matrix for the support vectors can be stored in main memory. If this is not the case, then alternative methods are required, such as decomposition.

In decomposition approaches, the data (and correspondingly the parameters) are split into a number of fixed-size sets, each called a working set. Optimization occurs on each working set while holding the other parameters fixed. This effectively performs coordinate descent on subsets of the parameters. The popular software implementation SVMLight [10] and SVMTorch [11] uses the decomposition strategies. The sequential minimal optimization (SMO) algorithm [12] is an extreme form of decomposition using working sets of two data points. The smallest working set that can be optimized is 2 if the constraints (3) for SVM classification are to hold. SMO takes advantage of the fact that under this condition the optimization sub_problem for standard SVM can be solved

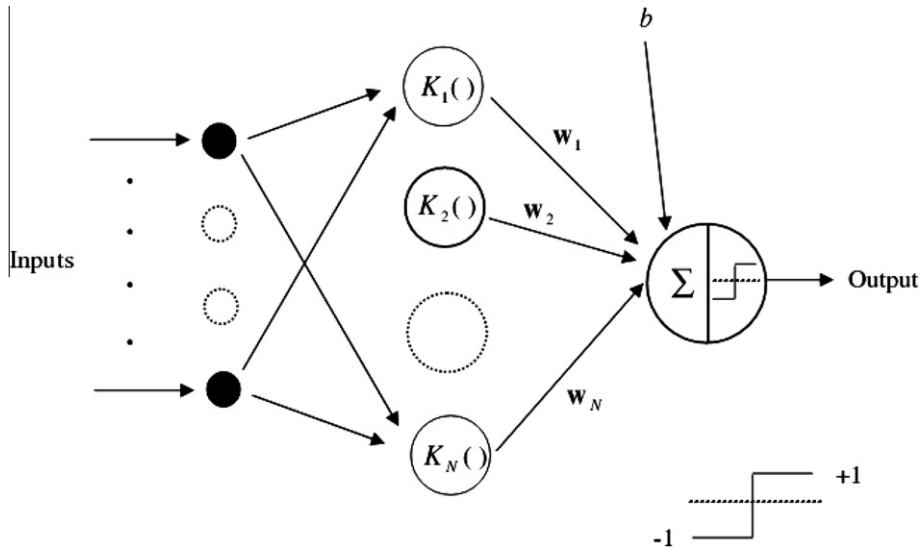

Figure 3 Architecture of SVM.

analytically. SMO exhibits better scaling properties and has reduced demand on main memory than Chunking. The popular software implementation LIBSVM [13] implements a variant of SMO for classification, regression, and single class learning settings.

4.2. Iterative algorithms

Gradient descent can be applied to the primal SVM optimization problem resulting in an iterative algorithm. The main advantage of iterative methods is that they result in algorithms with few steps and so are simple to implement. The disadvantage is that in general they exhibit linear convergence and so are slower than standard QP solvers.

4.3. Exploiting alternative SVM formulations

By modifying Eq. (4) subject to constraint in Eq. (3), it is possible to simplify the resulting optimization problem. This may involve simplifying or reducing the number of constraints by modifying the error functional or penalization. For example, an approach called Lagrangian SVM (LSVM) [14] uses a learning formulation, which results in an optimization problem that depends on solving systems of linear inequalities.

LSVM has formulation:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}(\|w\|^2 + b^2) + \frac{C}{2}\sum_{i=1}^{\ell}\xi_i^2 \\ &\text{s.t.} \quad y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \forall i \end{aligned} \tag{9}$$

The benefit is that objective function is strongly convex and equality constrain disappears in its dual:

$$\begin{aligned} &\max_{\alpha} \; e^T\alpha - \frac{1}{2}\alpha^T Q\alpha \\ &\text{s.t.} \quad \alpha \geq 0 \end{aligned} \tag{10}$$

LSVM algorithm is based directly on Karush–Kuhn–Tucker necessary and sufficient optimality conditions for the dual problem, it can be written in the following equivalent form. For any positive β,

$$\alpha = Q^{-1}(e + ((Q\alpha - e) - \beta\alpha)_+) \tag{11}$$

This leads to the following iterative scheme,

$$\alpha^{i+1} = Q^{-1}(e + ((Q\alpha^i - e) - \beta\alpha^i)_+) \tag{12}$$

Mangasarian established the global linear convergence from any starting point under condition $0 < \beta < 2/C$. LSVM requires nothing more complex than the inversion Q^{-1} computed once at the beginning of the algorithm, and Sherman–Morrison–Woodbury identity will be used. For linear decision boundaries, this algorithm can solve problems with millions of samples in minutes on a desktop computer. The drawback is that the learning problem is modified to minimize the square of the original SVM loss function and regularization is also applied to the constant offset b. It is still an open question how these modifications affect generalization performance.

5. Training of SVM methods

Different methods for training SVM will be examined. Where training of SVM consists of determining the optimal value of non-negative multipliers α in Eq. (6).These methods are:

i. Particle swarm optimization (categorizing in iterative methods).
ii. Quantum-behaved particle swarm optimization (subset selection methods).
iii. Quadratic program using active set strategy (subset selection methods).
iv. Least square version of SVM (alternative SVM formulations).

For comparison we try to pick one method from different category of training algorithms mentioned in the survey above.

5.1. Particle swarm optimization

The particle swarm optimization (PSO) method has been introduced by Kennedy and Eberhart [15] and is inspired by the emergent motion of a flock of birds searching for food. As a

stochastic search scheme, PSO has characters of simple computation and rapid convergence capability.

PSO is a population-based heuristic search technique; each particle represents a potential solution within the search space. Each particle has a position vector $\mathbf{X}_i$, a velocity vector $\mathbf{V}_i$, the position at which the best fitness pbesti encountered by the particle so far, and the best position of all particles gbest in current generation. The updating equations of PSO are as follows:

$$\begin{aligned}\mathbf{V}_i(t+1) &= w\mathbf{V}_i(t+1) + \mathbf{c}_1\mathbf{r}_1(\mathbf{X}_{\text{pbesti}}(t) - \mathbf{X}_i(t)) \\ &\quad + \mathbf{c}_2\mathbf{r}_2(X_{\text{gbest}}(t) - \mathbf{X}_i(t)) \\ \mathbf{X}_i(t+1) &= \mathbf{X}_i(t) + \mathbf{V}_i(t+1)\end{aligned} \tag{13}$$

where the parameters $\mathbf{c}_1$ and $\mathbf{c}_2$ are set to constant value, which are normally taken as 2, $\mathbf{r}_1$ and $\mathbf{r}_2$ are two random values, uniformly distributed in [0, 1], w is inertia weight which controls the influence of previous velocity on the new velocity. For the second part in Eq. (13) is called "cognition" character and the third part is "social" character of particles.

5.1.1. Particle swarm for training SVM [16]

Because the Lagrange multipliers α, constitute a vector $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_l]$ in one-dimensional space, the optimization (8) can be solved by PSO. Differing from the general PSO, all particles of the PSO training SVM must satisfy both constraints $\sum_{i=1}^{N}\alpha_i y_i = 0$ and $C \geqslant \alpha_i \geqslant 0, \forall i$. Thus, the PSO algorithm must be improved according to constraint $C \geqslant \alpha_i \geqslant 0, \forall i$, Eq. (9) can be written in the following form [16]:

$$\begin{aligned}\text{temp} - {}^{(t+1)}_{i,d} &= w \cdot {}^{(t)}_{i,d} + c_1\,\text{rand}_1()(\text{pbest}_{i,d} - \alpha^{(t)}_{i,d}) \\ &\quad + c_2\,\text{rand}_2()(\text{gbest}_d - \alpha^{(t)}_{i,d})\end{aligned}$$

$${}^{(t)}_{i,d} = \begin{cases} C - \alpha^{(t)}_{i,d}, & \alpha^{(t)}_{i,d} + \text{temp} - {}^{(t+1)}_{i,d} > C \\ {}^{(t+1)}_{i,d}, & \alpha^{(t)}_{i,d} + \text{temp} - {}^{(t+1)}_{i,d} < C \\ \text{temp} - {}^{(t+1)}_{i,d}, & \text{otherwise} \end{cases} \tag{14}$$

According to linear equality constraint $\sum_{i=1}^{N}\alpha_i y_i = 0$

$$\begin{aligned}\sum_{d=1}^{l} a^{(t+1)}_{i,d} y_d &= \sum_{d=1}^{l} a^{(t)}_{i,d} y_d + \sum_{d=1}^{l} {}^{(t+1)}_{i,d} y_d = \sum_{d=1}^{l} {}^{(t+1)}_{i,d} y_d \\ &= \sum_{{}^{(t+1)}_{i,d} y_d > 0} {}^{(t+1)}_{i,d} y_d - \sum_{{}^{(t+1)}_{i,d} y_d < 0} \left(-{}^{(t+1)}_{i,d} y_d\right) \\ &= \text{sum}\, V_+ - \text{sum}\, V_- \end{aligned} \tag{15}$$

Thus, the Lagrange multipliers $\alpha^{(t+1)}_{i,d}$ satisfies both constraints $\sum_{i=1}^{N}\alpha_i y_i = 0, C \geqslant \alpha_i \geqslant 0, \forall i$.

5.1.2. Algorithm to train SVM by PSO

Initialize

(a) Set constants $w_{\min}$, $w_{\max}$ is inertia weights equation (13), p: no. of iterations. C: constant defined by user such that $C \geqslant \alpha_i \geqslant 0, \forall i$, parameters $\mathbf{c}_1$ and $\mathbf{c}_2$ are set to constant value as in Eq. (13), $\text{iter}_{\max}$: maximum no. of iterations, $\mathbf{V}_o^{\max}$ maximum velocity, d: no. of particles.
(b) Set constants $t = 0$. Set random number seed.
(c) Randomly initialize particle positions $\alpha^{(0)}_i \in R^{\ell}, 0 \leqslant \alpha_i \leqslant C$.
(d) For $i = 1, 2, \ldots, p$, $d = 1, 2, \ldots, \ell$ where ℓ is equal to no. of samples.
(e) Randomly initialize particle velocities $0 \leqslant {}^{(0)}_{i,d} \leqslant {}^{\max}_{d}$ for $i = 1, 2, \ldots, p$, $d = 1, 2, \ldots, \ell$.
(f) Evaluate cost function values (Eq. (8))

$$\min_{\alpha}\phi\left(\alpha^{(0)}_i\right) = \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i$$

Using design space coordinates $\alpha^{(0)}_i$ for $i = 1, 2, \ldots, p$.
(g) Set $\phi(\alpha^{\text{pbest}}_i) = \phi(\alpha^{(0)}_i)$ and $\text{pbest}_i = \alpha^{(0)}_i$ for $i = 1, 2, \ldots, p$.
(h) Set $(\varphi^{\text{gbest}}_i)$ to minimal (φ^{pbest}_i) and gbest to corresponding $\alpha^{(0)}_i$.

Optimize

(a) Update inertia weight w using: $w = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \times t$.
(b) Update particle velocity vectors $\mathbf{V}^{t+1}_i$.
(c) Update particle position vectors $\alpha^{(t+1)}_i$. Using $\alpha^{i+1}_{i,d} = \alpha^{i}_{i,d} + {}^{t+1}_{i,d}$.
(d) Evaluate cost function $\min\phi(\alpha) = \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i$. Using design space coordinator $\alpha^{i}_{t+1}, i = 1, 2, \ldots, d$.
(e) If $\phi(\alpha^{(t+1)}_i) < \phi^{\text{pbest}}_i$ then $(\phi^{\text{pbest}}_i) = \phi(\alpha^{(t+1)}_i)$, $\text{pbest}_i = \alpha^{(t+1)}_i$.
(f) If $\phi(\alpha^{(t+1)}_i) < \phi^{\text{gbest}}_i$ then $(\phi^{\text{gbest}}_i) = \phi(\alpha^{(t+1)}_i)$. For $i = 1, 2, \ldots, p$.
(g) If all members of gbest fulfil the Karush–Kuhn–Tucker (KKT) optimality conditions of the QP problem, or the number of iterations, t, is up to iter_{max}, then go to **Report Results**.
(h) Increment t.
(i) Go to 2(a).

Report Results

Terminate

Complexity of the algorithm. The complexity of the above algorithm depends on evaluation of fitness function in each step, the evaluation of fitness function is merely evaluation of the kernel used $K(x_i, x_j)$, which requires $O(d_L)$ operations, where d_L is the size of training data.

5.2. Quantum-behaved particle swarm optimization [17]

In the quantum physics, the motion state of particles having the momentum and energy can be represented by wave function ψ (Eq. (17)). So in the quantum model, the motion state of particles can be expressed by wave function instead of denoting method of the velocity and position for the particles. At the same time, based on the Heisenberg uncertainty principle, the velocity and position of particles cannot be accurately measured simultaneously. The probability of occurrence for the particle in a case of position and time can be denoted by probability density function of the corresponding wave function ψ.

In order to facilitate the analysis, particles are considered to move in the one dimensional space. Assume p is the canter of Delta potential field, so the potential energy of particles in the Delta potential trough is:

$$V(x) = -\gamma\&\delta(x - p) \tag{16}$$

The wave function of particle can be got by the above equation:

$$\psi(x) = \frac{1}{\sqrt{L}} * \exp(-\|p - x\|/L) \tag{17}$$

In Eq. (17), the parameter L relying on the width of potential trough is used to determine the search domain. The particles move according to the following iterative equation:

$$\text{mbest} = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \ldots, \frac{1}{M}\sum_{i=1}^{M} P_{in}\right) \tag{18}$$

$$PP_{id} = \phi \times P_{id} + (1-\phi) \times P_{gd}, \quad \phi = \text{rand} \tag{19}$$

$$x_{id} = PP_{id} \pm \theta \times |\text{mbest}_d - x_{id}| \times \ln(1/u), \quad u \text{ is a random variable} \tag{20}$$

where mbest is the middle position of the particle swarm (pbest); PP_{id} is the random point between P_{id} and P_{gd}, θ is the only parameter of the QPSO algorithm. Commonly let $\theta = (1.0 - 0.5) \times (\text{MAXITER} - T)/\text{MAXITER} + 0.5$, where T is the current number of iterations, MAXITER is the maximum number of iterations.

5.2.1. Training algorithm for training SVM by QPSO

Using QPSO to solve the SVM equation (8) requires

- Criteria for optimality.
- A way to decompose the problem
- A way to extend QPSO to optimize SVM sub problem.

Criteria for optimality. The Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient for optimality. Since H is a positive semi-define matrix [18] (the kernel function used is positive semi-define).

Decompose the problem. The decomposition method presented here is due to [19], and works on the method of feasible directions. The idea of the method is to find the steepest feasible direction d of ascent on the objective function W (as defined in Eq. (3)), under the requirement that only the q components is a nonzero value. The α_i corresponding to the q components will be included in the working set. Finding an approximation to d is equivalent to solve the following problem:

$$\begin{aligned} &\text{Maximise} \quad \nabla W(\alpha)^T d \\ &\text{Subject to } y^T d = 0, \quad d_i \geq 0, \quad \text{if } \alpha_i = 0 \\ &\qquad\qquad\qquad\quad d_i \leq 0, \quad \text{if } \alpha_i = C \\ &\qquad\qquad\qquad\quad d_i \in \{-1, 0, 1\} \\ &\qquad\qquad\qquad\quad |\{d_i : d_i \neq 0\}| = q \end{aligned} \tag{21}$$

for $y^T d$ to be equal to zero, the number of elements with sign matches between d_i and y_i must be equal to the number of elements with sign mismatches between d_i & y_i. Also, d should be chosen to maximize the direction of ascent $\nabla W(\alpha)^T d$. It is necessary to rewrite the objective function Eq. (8) as a function that is only dependent on the working set. Split α into two sets $\alpha_B and \alpha_N$. If α, y and H are appropriately rearranged, we have

$$\alpha = \begin{pmatrix} \alpha_B \\ \alpha_N \end{pmatrix}, \quad Y = \begin{pmatrix} Y_B \\ Y_N \end{pmatrix}, \quad H = \begin{pmatrix} H_{BB} & H_{BN} \\ H_{NB} & H_{NN} \end{pmatrix} \tag{22}$$

Since only α_B is going to be optimized. Q_1 is rewritten in terms of α_B. If terms that do not contain α_B are dropped, the optimization problem remains essentially the same. Also, since H is symmetric, the problem is to solve:

$$\begin{aligned} \min_{\alpha_B} \quad & Q_1(\alpha_B) = \frac{1}{2}\alpha_B^T H_{BB}\alpha_B - \alpha_B^T(e - H_{BN}\alpha_N) \\ \text{Subject to } & \alpha_B^T y_B + \alpha_N^T = 0 \\ & \alpha_B \geq 0 \\ & C1 - \alpha_B \geq 0 \end{aligned} \tag{23}$$

An algorithm to train SVM with QPSO

- *Initialize*
- A feasible solution that satisfies the linear constraint $\alpha^T y = 0$, with constraints $0 \leqslant \alpha_i \leqslant C$ also met, is needed.
- Construction of initial solution:
  - Let $c \in [0, C] \subseteq R$, and γ (some positive integer) $\leqslant \min(\#$ +ve examples, i.e. $(y_i = +1)$, # of −ve examples, i.e. $(y_i = -1))$ in the training set.
  - Randomly pick a total of γ positive examples, γ negative examples, and initialize their corresponding α_i to c.
  - By setting all other α_i to zero, the initial solution will be feasible.
  - The value 2γ gives the total number of initial support vectors, and since these initial support vectors are a randomly chosen, it is suggested that the value of γ be kept small.
- *Repeat*

  Decompose the problem
  - Sorting the training vectors in increasing order according to $y_i \nabla W(\alpha)_i$.
  - Select **q** variables for the working set B.
  - The remaining $1 - \mathbf{q}$ variables (set N) are fixed at their current value.
  - Assuming q to be even, a "forward pass" select $q/2$ examples from the front of the sorted list, and a "backward pass" selects $q/2$ examples from the back of the sorted list.

  Initialize

  All particles be initialized such that $\alpha_B^T y_B + \alpha_N^T y_N = 0$ is met. This is done as follows:
  - Set each particle in the swarm to the q-dimensional vector α_B.
  - Add a random q-dimensional vector δ satisfying to each particle, under the condition that the particle will still lie in the hypercube $[0, C]^q$. Initializing the swarm in this way ensures that the initial swarm lies in the set of feasible solutions $P = P|AP = -\alpha_N^T y_N$ allowing the flight of the swarm to be defined by feasible directions.
  - Set the iteration number to zero

  Use QPSO to optimize W on B.

  Repeat

  (a) Evaluate the performance $W(\alpha)$ of each particle.
  (b) Evaluate new P_{id} of each particle.
  (c) Evaluate new P_{gd}
  (d) Evaluate mbest by Eq. (18).
  (e) Evaluate the random point PP_{id} of each particle by Eq. (19).
  (f) Move each particle to its new position, according to Eq. (20).

  Make $T = T + 1$ go to step (a) until terminal condition satisfied (set by user).

Until the KKT condition are met

- **Return the optimized α_i from B to the original set of variables**.
- **Terminate and return** α.

Complexity of the algorithm. The complexity of the above algorithm depends on evaluation of the fitness function in each step that is merely evaluation of the used kernel $K(x_i, x_j$, which requires $O(d_L)$ operations, where d_L is the size of training data. So one can deduce that the decomposition will be of order $O(N\log(N))$ and the evaluation of fitness function will be of order $O(N)$ so the total time will be of order $O(N)+O(N\log(N))$.

5.3. Quadratic program using active set strategy

The medium-scale algorithm is an active-set strategy (also known as a projection method) similar to that of Gill et al described in [20]. It has been modified for both linear programming (LP) and quadratic programming (QP) problems.

The basic idea of the algorithm is to find the active set A, i.e., those inequality constraints that are fulfilled with equality. If the set A is known, the KKT conditions reduce to a simple system of linear equations which yields the solution of the QP problem. Because the set A is unknown in the beginning, it is constructed iteratively by adding and removing constraints and testing if the solution remains feasible.

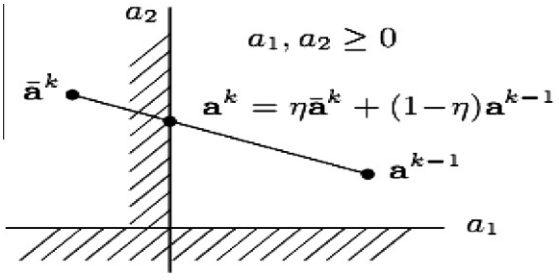

Figure 4 Affine scaling of the non-feasible solution.

Algorithm

The construction of the set A starts with an initial active set A^0 containing the indices of the bounded variables (lying on the boundary of the feasible region) whereas those in $F^0 = 1, \ldots, N \setminus A^0$ are free (lying in the interior of the feasible region) (Fig. 4).

Then the following steps are performed repeatedly for $k = 1, 2, \ldots$:

1. Solve the KKT system for all variables in F^k.
2. If the solution is feasible, find the variable in A^k that violates most of the KKT conditions, move it to F^k then go to 1.
3. Otherwise find an intermediate value between old and new solution lying on the border of the feasible region, move one bounded variable from F^k to A^k then go to 1. The intermediate solution that is obtained in step 3 is computed as follows: $a^k = \eta a^{-k} + (1-\eta)a^{k-1}$ with maximal $\eta \in [0, 1]$ (affine scaling), where a^{-k} is the solution of the linear system in step 1, i.e. the new iterate a^k lies on the connecting line of a^{k-1} and a^k, see Fig. 4.

While the optimum is found if during step 2 no violating variable is left in A^k.

Complexity of algorithm

Active-set algorithm needs $O(N_f^2 + N)$ memory, where N is the number of free unbounded variables.

5.4. Least square support vector machine (LSSVM)

The least squares version of the SVM classifier is formulated by the classification problem as [21]:

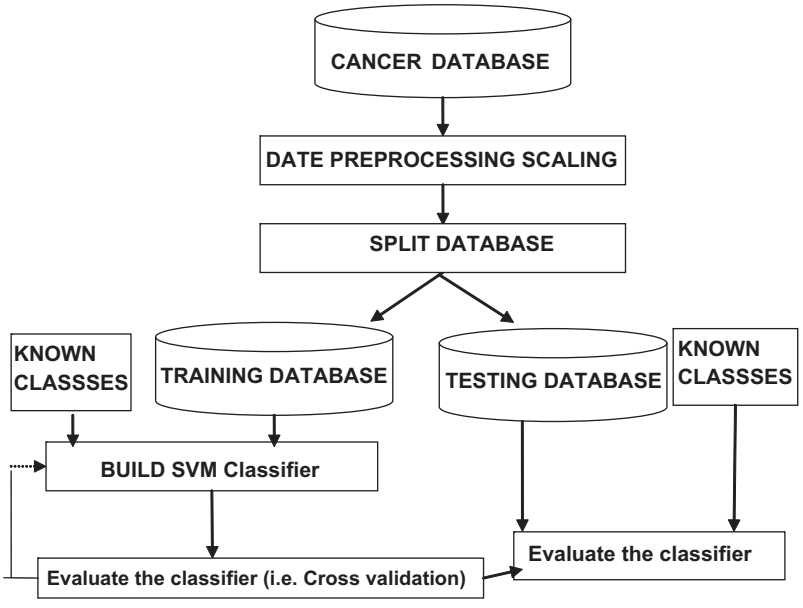

Figure 5 Methodology of cancer diagnosis model.

$$\begin{aligned} &\min \quad \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i \\ &\text{Subject to } y_i(\langle w.x_i\rangle + b) = 1 - \zeta_i, \quad i = 1, 2, \ldots, n \end{aligned} \tag{24}$$

According to Eq. (19), their dual problems are built as follows:

$$L_D \quad \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n}\zeta_i^2 - \sum_{j=1}^{n}\alpha_i\{y_i[wx_i + b] - 1 + \zeta_i\}. \tag{25}$$

where α_i are Lagrange multipliers (which can be either positive or negative) now due to the equality constraints as follows from the Kuhn–Tucker conditions [18],

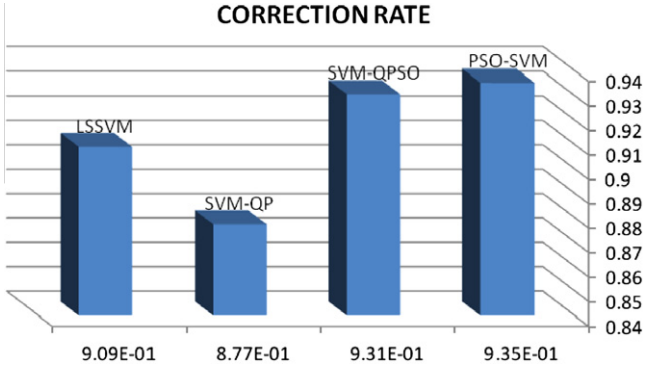

Figure 6 Correction rate value on testing data, 40–60% training test partition. The SVM-PSO shows the highest accuracy.

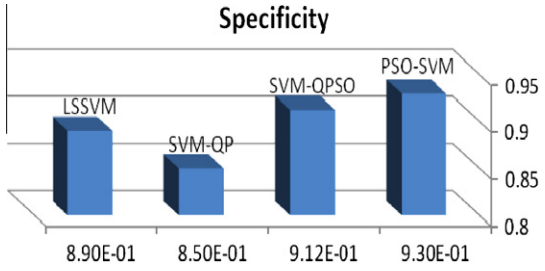

Figure 7 Specificity value on testing data, 40–60% training test partition. The SVM-PSO shows the highest accuracy.

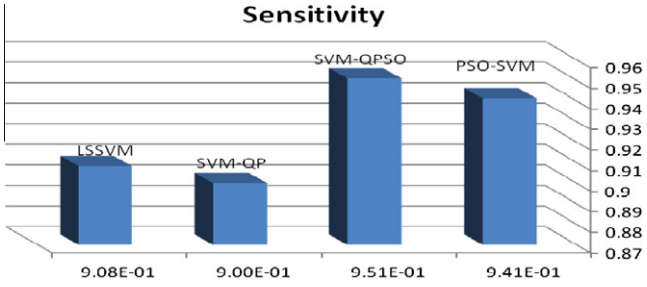

Figure 8 Sensitivity value on testing data, 40–60% training test partition. The SVM-QPSO shows the highest accuracy.

$$\begin{aligned} &\frac{\partial}{\partial w}L_P(w^*, b^*, \alpha^*, \zeta^*) \to w = \sum_i \alpha_i^* y_i x_i \\ &\frac{\partial}{\partial \zeta}L_P(w^*, b^*, \alpha^*, \zeta^*) = -\sum_i \alpha_i^* y_i = 0 \\ &\frac{\partial}{\partial \zeta}L_P(w^*, b^*, \alpha^*, \zeta^*) \to \alpha_i^* = C\zeta_i \\ &\frac{\partial}{\partial \zeta}L_P(w^*, b^*, \alpha^*, \zeta^*) \to y_i(\langle w \cdot x_i\rangle + b) - 1 + \xi_i = 0 \end{aligned} \tag{26}$$

can be written immediately as the solution to the following set of linear equations [18]

$$\left[\begin{array}{ccc|c} \mathrm{I} & 0 & 0 & -Z^{\mathrm{T}} \\ 0 & 0 & 0 & -Y^{\mathrm{T}} \\ 0 & 0 & \gamma^{\mathrm{I}} & -I \\ \hline Z & Y & I & 0 \end{array}\right]\left[\begin{array}{c} \mathrm{w} \\ \mathrm{b} \\ \mathrm{e} \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 1 \end{array}\right] \tag{27}$$

where $z = [x_1^T y_1, x_2^T y_2, , \ldots, x_n^T y_n]$, $y = [y_1, \ldots, y_n]$, $\vec{1} = [1, \ldots, 1]$, $e = [e_1, \ldots, e_N]$, $\alpha = [\alpha_1, \ldots, \alpha_N]$The solution is also given by

$$\left[\begin{array}{c|c} 0 & -Y^{\mathrm{T}} \\ \hline Y & ZZ^{\mathrm{T}} + \gamma^{-1}\mathrm{I} \end{array}\right]\left[\begin{array}{c} \mathrm{b} \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1 \end{array}\right] \tag{28}$$

Mercer's condition can be applied again to the matrix

$$\begin{aligned} &\Omega = ZZ^T, \quad \text{where} \\ &\Omega_{kl} = y_k y_\ell x_k x_\ell = y_k y_\ell K(x_k, x_\ell) \end{aligned} \tag{29}$$

LSSVMs use a set of linear equations for training while SVMs use a quadratic optimization problem, then $X = A \setminus B$ is the solution to the equation $AX = B$ computed by Gaussian elimination with partial pivoting [22] and this is the technique we use with LSSVM.

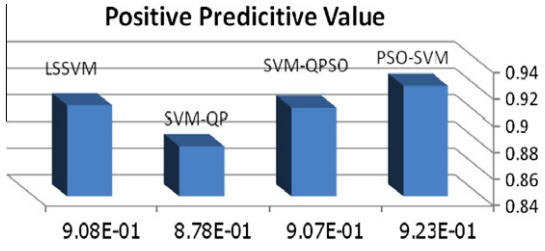

Figure 9 Positive predictive value on testing data, 40–60% training test partition. The SVM-PSO shows the highest accuracy.

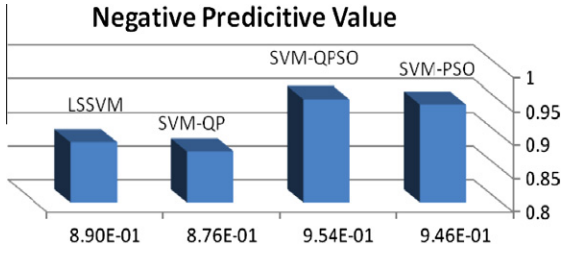

Figure 10 Negative predictive value on testing data, 40–60% training test partition. The SVM-QPSO shows the highest accuracy.

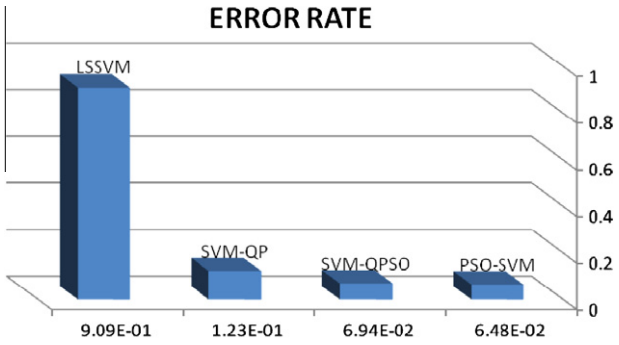

Figure 11 Error rate on testing data, 40–60% training test partition. The SVM-PSO shows the lowest error.

Complexity of algorithm

Gaussian elimination solves a system of n equations for n unknowns in "$n(n + 1)/2$" divisions, "$(2n^3 + 3n^2 - 5n)/6$" multiplications, and "$(2n^3 + 3n^2 - 5n)/6$" subtractions, for a total of approximately "$2n^3/3$" operations. So it has a complexity of order $O(N^3)$.

6. Methodology

Fig. 5 depicts the proposed methodology for Cancer Diagnosis Model by pre-processing the data using scaling (we scale linearly each attribute to the range of [0, 1]), pre-processed data are split into training and testing (independent) datasets. The training dataset is used to build SVM classifier. The validity

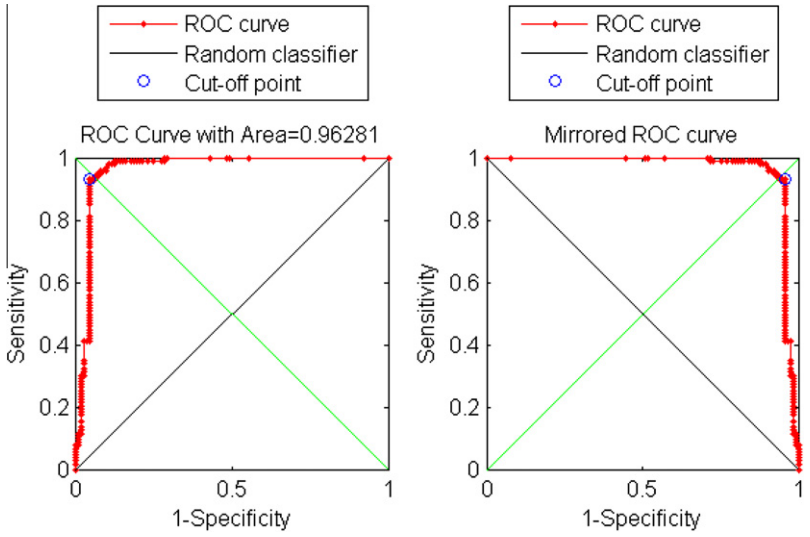

Figure 12 Receiver operating characteristic (ROC) curves for classifier resulted from training of SVM with PSO. The area under curve is 0.96281.

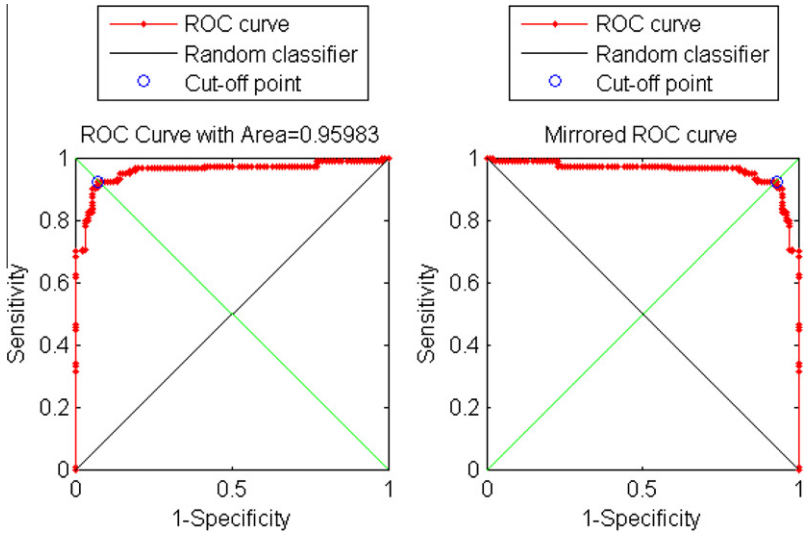

Figure 13 Receiver operating characteristic (ROC) curves for classifier resulted from training of SVM with QPSO. The area under curve is 0.95983.

of each classifier created with the classifier is evaluated using the sensitivity and specificity of the SVM classifier in distinguishing cancer patients from non-cancer controls. SVM classifiers are built for various combinations of features until the classification accuracy of the SVM classifier reaches its maximum value. Estimates of classification accuracy are calculated by using the cross validation method where a validation dataset is used to evaluate the generalization error.

Four different methods to construct classifier (i.e. training SVM) are:

1. Particle swarm,
2. Quantum behaved particle swarm,
3. Quadratic program using active set strategy,
4. Gaussian elimination with partial pivoting of set of linear equations of LSSVM.

The experiments are done on the Wisconsin Database of Breast Cancer (WDBC) from the UCl [23]. The data taken from fine needle aspirates from human breast tissue were analyzed. They have been collected by Wolberg and Mangasarian [24], at the University of Wisconsin-Madison Hospital. The data consists of 683 records of virtually assessed nuclear features of fine needle aspirates taken from patient's breasts. The results of the four methods were compared.

7. Experimental results and discussion

The effectiveness of the four different methods for training support vector machine will be evaluated and compared.

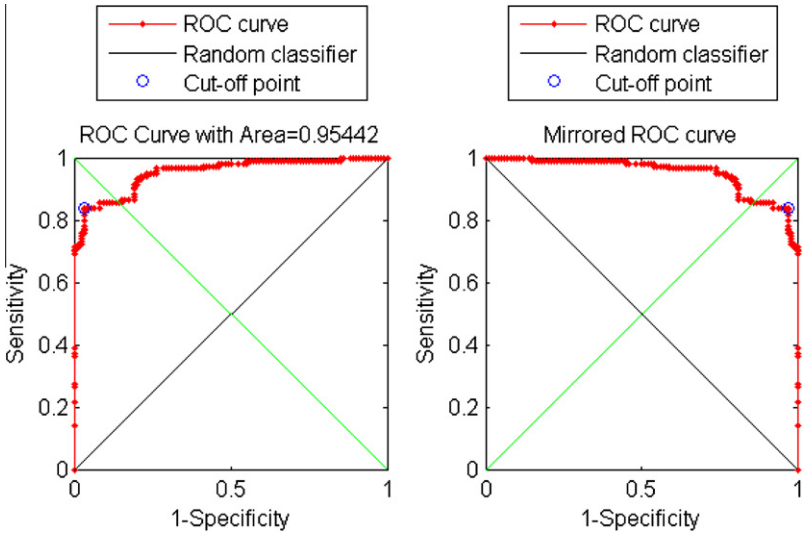

Figure 14 Receiver operating characteristic (ROC) curves for classifier resulted from training of SVM with QP. The area under curve is 0.95442.

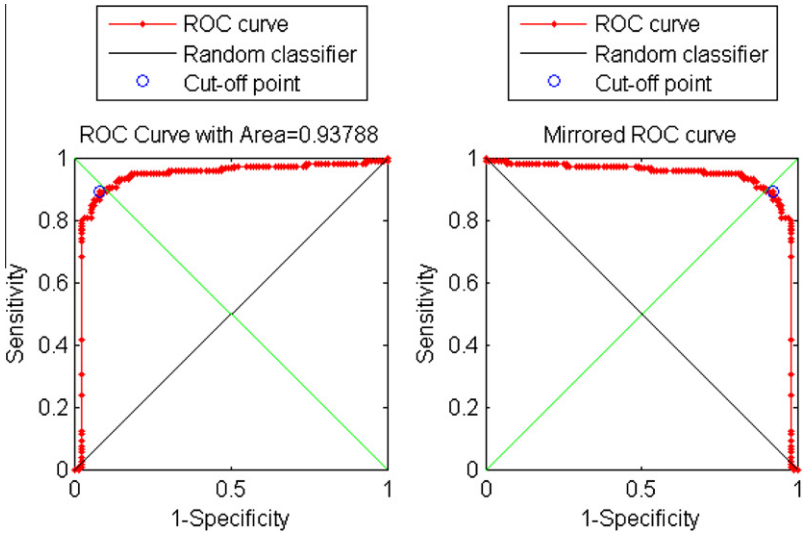

Figure 15 Receiver operating characteristic (ROC) curves for classifier resulted from training of LSSVM. The area under curve is 0.93788.

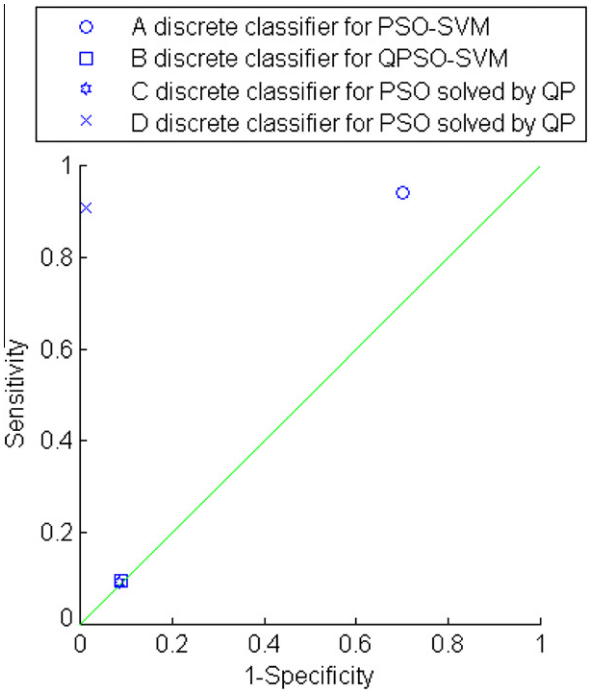

Figure 16 A basic ROC graph showing four discrete classifiers.

a. *Particle swarm optimization:* A MATLAB code was written to train SVM by PSO. The KKT conditions needed to be satisfied within an error threshold of 0.005 in order to find an optimal solution quickly. The upper bound C was kept at 100.0.
b. *Quantum behaved particle swarm:* A MATLAB code was written to train SVM by QPSO. The KKT conditions needed to be satisfied within an error threshold of 0.02.Optimization of the working set terminated when the KKT conditions on the working set were met within an error of 0.001, or when the swarm has optimized for five hundred iterations. The following parameters defined for the experimental QPSO: By letting $\gamma = 10$, a total of 20 initial support vectors were chosen to start the algorithm. The value of Contraction–Expansion Coefficient θ is set to be 0.7, along with iteration increases, the value of α linearly reduces to 0.3, so $\theta = (0.7 - 0.3) * (MAXITER - T)/MAXITER + 0.3$.

For each experiment the upper bound C was kept at 100.0.

a. *Quadratic program* using active set strategy: the BIOLEARNING toolbox under the BIOINFO of the MATLAB toolbox is used.
b. *Least square support vector machine:* The BIOLEARNING toolbox under the BIOINFO of the MATLAB toolbox is used.

Training was done with the kernel function: $k(x, x_i) = \exp\left(-\frac{\|x-x_i\|^2}{1.0^2}\right)$.

The aim of the above comparison is to place the SVM's performance trained by the quantum particle swarm and particle swarm into perspective, so two more machine learning techniques were evaluated alongside it.

7.1. Measures for performance evaluation

Several measures were used in order to evaluate the effectiveness of the given methods. These measures are classification accuracy (Fig. 6), analysis of specificity (Fig. 7), sensitivity (Fig. 8), positive predictive value (Fig. 9), negative predictive value (Fig. 10), error rate (Fig. 11), receiver operating characteristic (ROC) curves Figs. 12–15 and confusion matrix [25].

Discrete classifier: is one that outputs only a class label. Each discrete classifier produces an (fp rate, tp rate) pair corresponding to a single point in ROC space.

The classifier results from training SVM with PSO shows best area under curve which means that this classifier have better average performance.

Discrete Roc Curve (Fig. 16).

Several points in ROC space are important to note. The lower left point (0, 0) represents the strategy of never issuing a positive classification; such a classifier commits no false positive errors but also gains no true positives. The opposite strategy, of unconditionally issuing positive classifications, is represented by the upper right point (1, 1). The point (0, 1) represents perfect classification. B, C's performance is perfect as shown.

Informally, one point in ROC space is better than another if it is to the northwest (tp rate is higher, fp rate is lower, or both) of the first. Classifiers appearing on the left hand-side of an ROC graph, near the X axis, may be thought of as conservative: they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well.

Classifiers appearing on the left hand-side of an ROC graph, near the X axis, may be thought of as conservative: they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well. Classifiers on the upper right-hand side of an ROC graph may be thought of as liberal: they make positive classifications with weak evidence so they classify nearly all positives correctly, but they often have high false positive rates.

8. Conclusion

To place the SVM's performance trained by swarm intelligence into perspective, four more machine learning techniques were evaluated alongside it. The techniques selected were PSO, QPSO, active set strategy and LSSVM, PSO and QPSO records slightly higher overall accuracy (0.9352%–0.9306%) than the other techniques (0.8773%–0.9091%) on set. Considering the abnormality assessment rank feature in the proposed comparative study is beyond our plan in this work, so it will be considered in the extension of this work to re-compare the described techniques.

When using the SVM, three obstacles are confronted: how to choose the kernel function and optimal input feature subset for SVM, and how to set the best kernel parameters. These obstacles are crucial because the feature subset choice influences the appropriate kernel parameters and vice versa.

Feature selection is an important issue in building classification systems. It is advantageous to limit the number of input features in a classifier to in order to have a good predictive and less

computationally intensive model. Building a model that can handle the three obstacles at the same time is a very important issue and needs further research and work in the future.

References

[1] West D, Mangiameli P, Rampal R, West V. Ensemble strategies for a medical diagnosis decision support system: a breast cancer diagnosis application. Eur J Operat Res 2005;162:532–51.

[2] <http://www.imaginis.com/breasthealth/breast_cancer.asp> [accessed May 2010].

[3] Kordylewski H, Graupe D, Liu K. A novel large-memory neural network as an aid in medical diagnosis applications. IEEE Trans Inform Technol Biomed 2001;5(3):202–9.

[4] Wolberg WH, Street WN, Mangasarian OL. Breast cytology diagnosis via digital image analysis. Anal Quant Cytol Histol 1993;15(6):396–404.

[5] Mu T, Nandi AK. Breast cancer detection from FNA using SVM with different parameter tuning systems and SOM–RBF classifier. J Franklin Inst 2007;344(3–4):285–311.

[6] Vapnik VN. The Nature of Statistical Learning Theory. New York: Springer-Verlag; 1995.

[7] Murtaghand BA, Saunders MA. Large-scale linearly constrained optimization. Math. Programming 1978;14:41–72.

[8] Vanderbei RJ. LOQO: an interior point code for quadratic programming. Optimizat Methods Software 1999;12:451–84.

[9] Osuna E, Freund R, Girosi F. Improved training algorithm for support vector machines. In: NNSP; 1997.

[10] Joachims T. Web page on SVMLight: <http://svmlight.joachims.org>.

[11] Collobert R, Bengio S. SVMTorch: <http://www.idiap.ch/machine_learning>.

[12] Platt J, Schölkopf B, Burges CJC, Smola AJ. Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel methods support vector learning. Cambridge, MA: MIT Press; 1999. p. 185–208.

[13] Chang CC, Lin CJ. LIBSVM: a library for support vector machines; 2001. <http://www.csie.ntu.edu.tw/_cjlin/libsvm>.

[14] Mangasarian OL, Musicant DR. Lagrangian support vector machines. J Mach Learning Res 2001;1:161–77.

[15] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural network, IV[C], vol. 4(2). Piscataway, NJ: IEEE Service Center; 1995. p. 1942–8.

[16] Tang F, Chenn M, Wang Z. New approach to training support vector machine. J Syst Eng Electron 2006;17(1):200–5.

[17] Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior. In: Congress on evolutionary computation. IEEE, Piscataway, NJ, ETATS-UNIS (Monographie); 2004. p. 325–31.

[18] Fletcher R. Practical methods of optimization. Wiley; 1988.

[19] Joachims T, Scolkopf B, Rurges CJC, Smola AJ. Making large-scale SVM learning practical. In: Advances in Kernel methods support vector learning. Cambridge, MA: MIT Press; 1999. p. 169–84.

[20] Gill PE, Murray W, Saunders MA, Wright MH. Procedures for optimization problems with a mixture of bounds and general linear constraints. ACM Trans Math Software 1984;10:282–98.

[21] Suykens J, Vandewalle J. Least squares support vector machine classifiers. Neural Process Lett 1999:293–300.

[22] David G. Linear and non linear programming. Luenberger Stanford University; 2002.

[23] Blake CL, Merz CJ. UCI repository of machine learning data base. Irvine, CA: University of California; 1998. <http://www.ics.uci.edu/mlearn/MLRepository.html>.

[24] Wolberg WH, Mangasarian OL. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. PNAS 1990;87:9193–6.

[25] Hand J, Till RJ. A simple generalization of the area under the ROC curve for multiple class classification problems. Mach Learning 2001;45:171–86.