Abstract: In this paper, we consider the benefits of applying support vector machines (SVMs) and radial basis function (RBF) for breast cancer detection. The Wisconsin diagnosis breast cancer (WDBC) dataset is used in the classification experiments; the dataset was generated from fine needle aspiration (FNA) samples through image processing. The 1-norm C-SVM (L1-SVM) and 2-norm C-SVM (L2-SVM) are applied, for which the grid search based on gradient descent based on validation error estimate (GDVEE) are developed to improve the detection accuracy. Experimental results demonstrate that SVM classifiers with the proposed automatic parameter tuning systems and the RBF classifier can be used as one of most efficient tools for breast cancer detection, with the detection accuracy up to 98%.

Keywords: Breast cancer detection, Parameter tuning, Radial basis function networks, Support vector machines.

1 Introduction

Worldwide, breast cancer is the most common form of cancer and the second most common cause of cancer deaths in females, which affects approximately 10% of all women at some stage of their life in the western world. Breast cancer may be detected via a cautious 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. FNA uses a needle smaller than those used for blood tests to remove fluid, cells, and small fragments of tissue for examination under a microscope. Research works on the Wisconsin diagnosis breast cancer (WDBC) data grew out of the desire of Dr. Wolberg to diagnose breast masses accurately based solely on FNA. Previously, researchers from the University of Wisconsin, Madison, applied image processing techniques to derive the WDBC dataset directly from digital scans of FNA slides, and then employed machine learning techniques to differentiate benign from malignant samples, which could be the earliest study of machine learning application to breast cancer detection. Later, several works on computational intelligence were developed in the area of breast cancer, including the multilayer perceptron [1], radial basis function (RBF) networks [2], fuzzy classifiers [3,6], clustering algorithms [4], evolutionary computation [5], principal component analysis [6], and different kernel-based methods [7]. Support vector machines (SVMs) are highly successful in solving various nonlinear and non-separable problems in machine learning, In addition to the original C-SVM learning method proposed by Cortes and Vapnik [8] in 1995. In order to use SVM, one needs to decide the value of the regularization parameter (C for C-SVM) and the kernel function before training the SVM classifier. Thus, choosing a suitable kernel function is imperative to the success of the learning process,
which is equivalent to selecting suitable kernel parameters for a parametric family of kernels (such as the RBF kernel). The regularization parameter also plays an important part in the SVM learning procedure, which controls the tradeoff between the complexity of an SVM and the number of nonseparable points.

In this study, different parameter tuning systems are developed for different SVM classifiers, such as the 1-norm C-SVM (L1-SVM) and 2-norm C-SVM (L2-SVM) to improve the detection accuracy of malignant masses for breast cancer diagnosis. The SVM parameter tuning procedure includes the choice of the tuning criterion and the tuning method. In our research, the RBF network is employed to process the clustering result, which can be viewed as RBF-based dimensionality reduction processing step for a single-layer perceptron (SLP).

This paper is organized as follows: Section 2 provides an explanation of the classification problems studied in this work. Section 3 gives a description of the theory of the SVM, RBF, and GD approaches. The proposed SVM parameter tuning systems which uses gradient descent based validation error estimate (GDVEE), as well as the RBF classifiers, are explained in Section 4. Experimental results and comparative analysis are provided in Section 5.

2 The Classification Problem

The classification problem addressed in this study is the detection of malignant breast tumors from a set of benign and malignant samples, called the WDBC dataset, which was obtained from the University of Wisconsin Hospitals, Madison, available at (ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/). Features in this dataset were computed from digitized FNA samples, as follows.

After the FNA sample was taken from a breast mass, the material was mounted on a microscope slide and stained to highlight the cellular nuclei. A portion of well differentiated cells was scanned using a digital camera. The image analysis software system Xcyt was used to isolate individual nuclei. An approximate boundary of each nucleus was provided as input and taken to convergence to the exact nuclear boundary, using a semi-automatic segmentation procedure called “snakes”. Beginning with a user defined approximate boundary as an initialization, the snake locates the actual boundary of the cell nucleus. In order to evaluate the size, shape and texture of each cell nuclei, 10 characteristics were derived, including the radius, perimeter, area, compactness, smoothness, concavity, concave points, symmetry, fractal dimension, and texture. Based on the features described above, the detection of malignant breast tumors can be viewed as a binary classification problem. Given a set of 1 labeled training samples \( z = \{ (x_i, y_i) \}_{i=1}^n \) in \((\chi \times Y)\), where \( \chi \) is the n-dimensional real feature space \((n = 30)\) with a binary label space \( Y = \{1, -1\} \), \( y_i \in Y \) is the label assigned to the sample \( x_i \in \chi \), the purpose is to seek a function \( \psi: \chi \rightarrow Y \) that best predicts the label for an input sample.

3 Theory

3.1 Support vector machines (SVMs)

The original idea of the SVM is to construct a hyper plane as the decision surface in such a way that the margin of separation between the positive and negative samples is maximized in an appropriate feature space. As most of the pattern recognition problems to be solved in practice are of nonlinear nature, kernel functions are employed to perform the nonlinear mapping, which computes the inner product matrix, the so-called kernel matrix, on pairs of samples in the transformed feature space. For kernel-based methods, the kernel matrix acts as a bottleneck. All the information available must be extracted from the kernel matrix. Different kernel functions have been designed based on their closure properties, among which the RBF kernel is commonly used, given by

\[
K(x, y) = \exp \left( -\frac{\|x - y\|^2}{\delta^2} \right)
\]

where \( \delta \) is the RBF kernel width set by the user. In the transformed kernel feature space \( k \), with a nonlinear mapping \( \phi : \chi^a \rightarrow k \), the separating function \( f(x) \) has the following form

\[
f(x) = \phi^T(\omega)\phi(x) + b = K(\omega, x) + b
\]

where \( \omega \) is the vector of weights and \( b \) is the bias. In the SVM learning procedure, first, \( f(x) \) is obtained from the training result; the label for an input sample is then estimated by the step function \( \text{sgn}(x) \):

\[
A_z(x) = \text{sgn}(f(x))
\]

where \( \text{sgn}(x) \) is 1 when \( x \geq 0 \), and -1 otherwise. This estimate \( A_z \) of \( \psi \) is attained by minimizing a
regularized empirical risk $R_{reg}$, of which different choices lead to different SVM classifiers.

### 3.1.1 Hard-margin SVM

The hard-margin SVM is used for clearly separable cases, of which the regularized optimization risk with expected to be minimized, is

$$R_{reg} = \frac{1}{2} \| w \|_k^2$$  

subject to

$$y_i (k(\omega, x_i) + b) \geq 1, \ i = 1, 2, ..., l$$

This is equivalent to maximizing

$$Q(\alpha) = \sum_{i=1}^{l'} \alpha_i - \frac{1}{2} \sum_{i=1}^{l'} \sum_{j=1}^{l'} y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$  

subject to

$$\sum_{i=1}^{l'} y_i \alpha_i = 0, \ \alpha_i \geq 0, \ i = 1, 2, ..., l$$

### 3.1.2 C-SVM learning

The C-SVM is a soft-margin SVM for non-separable cases, including the L2-SVM and L1-SVM, which introduces the margin slack vector $\xi$ to allow for the possibility of samples violating the following inequality:

$$y_i (k(\omega, x_i) + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, 2, ..., l$$

With the soft-margin loss

$$C(x, y, f) = \begin{cases} 0, & yf(x) \geq 1 \\ 1 - yf(x), & \text{otherwise} \end{cases}$$  

The L1-SVM attempts to minimize the regularized optimization risk by involving the 1-norm of the margin slack vector $\xi$ as

$$R_{reg} = \frac{1}{2} \| w \|_k^2 + C \sum_{i=1}^{l'} \xi_i$$  

subject to

$$y_i (k(\omega, x_i) + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, 2, ..., l$$

This is equivalent to maximizing

$$Q(\alpha) = \sum_{i=1}^{l'} \alpha_i - \frac{1}{2} \sum_{i=1}^{l'} \sum_{j=1}^{l'} y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$  

subject to

$$\sum_{i=1}^{l'} y_i \alpha_i = 0, \ 0 \leq \alpha_i \leq C, i = 1, 2, ..., l$$

where $C$ is the positive regularization parameter set by the user. The L2-SVM attempts to minimize the regularized optimization risk by involving the 2-norm of the margin slack vector $\xi$ as

$$R_{reg} = \frac{1}{2} \| w \|_k^2 + C \sum_{i=1}^{l'} \xi_i^2$$  

subject to

$$y_i (k(\omega, x_i) + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, 2, ..., l$$

This is equivalent to maximizing

$$Q(\alpha) = \sum_{i=1}^{l'} \alpha_i - \frac{1}{2} \sum_{i=1}^{l'} \sum_{j=1}^{l'} y_i y_j \alpha_i \alpha_j k(x_i, x_j) + \frac{1}{C} \delta_i$$  

subject to

$$\sum_{i=1}^{l'} y_i \alpha_i = 0, \ \alpha_i \geq 0, \ i = 1, 2, ..., l$$

where $C$ is the regularization parameter set by the user, and $\delta_i$ is the Kronecker $\delta$, which is 1 when $i = j$, and 0 otherwise. In fact, the L2-SVM can be viewed as a special case of the hard-margin SVM with the modified kernel matrix

$$K' = K + \frac{1}{C} I$$

where $K$ is the original kernel matrix, and $I$ is the identity matrix.

### 3.1.3 Decision of the Separating Hyper plane

The concept of support vectors (SVs) is derived from the Karush–Kuhn–Tucker (KKT) condition. Letting $\alpha^0$ denote the optimal solution of $Q(\alpha)$, SVs are defined as the training samples with non-zero $\alpha^0_i$. Margin SVs are defined as the training samples with $\alpha^0_i$ not equal to zero, but less than $C$ for the L2-SVM, which are distributed along the margin (see Figure 1). Non-margin SVs are defined as the training samples with $\alpha^0_i$ equal to $C$ for the L2-SVM, which are distributed either inside the margin but on the correct side of the decision surface, or on the wrong side of the decision surface (see Figure 1).

For the C-SVM learning methods, the optimal weights $\phi(\alpha^0)$ and bias $b^0$ of the separating hyper plane in the transformed kernel feature space can be derived from

$$\phi(\alpha^0) = \sum_{i=1}^{l'} \alpha^0_i y_i \phi(x_i)$$  

$$b^0 = -\frac{1}{2S} \sum_{x \in S} \sum_{x \in \hat{S}} \alpha^0_i y_i K(x, x_i)$$

where $S$ and $\hat{S}$ are two sets of SVs with the same size as that of $S$, but different labels of -1 and 1, respectively. The optimal values of the slack vector $\xi^0_i$ can be derived from the KKT conditions: for the L1-SVM,

$$\xi^0_i = \max \{0, 1 - y_i f(x_i)\}, i = 1, 2, ..., l$$  

where $y_i f(x_i)$ is the decision function of the SVM.
3.2 RBF Networks

RBF networks involve three layers, of which the structure is shown in Figure 2. (1) The input layer is made up of sensory units that connect the network to the environment. (2) The hidden layer acts as a nonlinear transformation from the input space to the hidden space \( \chi \rightarrow \chi' \), similar to a kernel function. (3) The output layer supplies the response of the network to the input pattern, which is to seek for an optimal separating function \( f: \chi' \rightarrow y \) in the transformed hidden space.

The nonlinear transformation \( \phi: \chi \rightarrow \chi' \) is made up of \( m \) real-valued functions
\[
\phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_m(x)]^T
\]
where \( G(\cdot) \) is known as the RBF, and \( \{t_i\}_{i=1}^m \) are taken to be the centers of the RBFs. There are different choices of the RBF, among which the Gaussian function is commonly used, as given by
\[
G(x, t_i) = \exp\left(-\frac{||x - t_i||^2}{2\delta^2}\right)
\]
The RBF learning procedure seeks a separating function
\[
f(x) = \sum_{i=1}^m \omega_i G(||x - t_i||) + b
\]
by minimizing the cost function
\[
\varepsilon(f) = \sum_{i=1}^l (y_i - f) + \lambda \|Df\|^2
\]
where \( D \) is a linear differential operator.

3.2 GD Approach

The GD approach is a popular derivative-based optimization method, which performs much faster than GS. For the GD-based SVM parameter tuning problem, the step performed in each iteration is proportionally decided by the gradient of certain optimization objective \( O(\theta) \) with respect to the SVM parameter vector \( \theta \), instead of the fixed step size used in GS. The GD approach is executed by first initializing the SVM parameter vector to some value \( \theta_0 \), secondly training the L2-SVM classifier with \( \theta_k \) at the kth iteration, and thirdly updating each element of \( \theta_k \), so that \( O(\theta_k) \) is minimized based on the rule
\[
\theta_{k+1,p} = \theta_{k,p} - \eta_p \exp\left(-\frac{k}{\tau}\right) \frac{\partial O(\theta_k)}{\partial \theta_{k,p}}, p = 1, 2, \ldots, |\theta_k|\tag{18}
\]
where \( |\theta_k| \) denotes the number of elements in the vector \( \theta_k \), and \( \eta_p \) and \( \tau \) are used to control the convergence speed (set by the user). However, the GD approach only works for differentiable systems.

4 Methodology

4.1 SVM Parameter Tuning System

The performance of an SVM with a particular dataset depends on the training procedure of the hyper plane weights and bias, as well as the tuning process of the regularization and kernel parameters \( \theta = [\delta, C]^T \). The regularization parameter controls the trade off between the complexity of the SVM classifier and the number of non-separable points. The kernel parameter decides the nonlinear
mapping into an appropriate feature space. The hyper plane weights and bias can be decided by minimizing the regularized empirical risk \( R_{\text{reg}} \) using quadratic programming. However, the problem of SVM parameter selection is still of much interest. Generally, an SVM parameter tuning system is composed of two blocks, of which one is for criterion generation (the generalized error estimate in this study), and the other is a rapid and accurate tuning process. If there are enough training data, it is possible to split a part of the training data for validation, and obtain a measure known as the validation error estimate (VEE), which is unbiased as the number of validation samples increases. However, if there are not enough training data, one would like to employ all of the available data for training, and generate the error estimate completely on the training set without using any validation set. In this work, SVM tuning systems are developed based on combination GDVEE tuning system.

The overall structure of the proposed SVM parameter tuning systems is shown in Figure. 3. The GDVEE tuning systems are only developed for the L2-SVM, as it is difficult to calculate \( \partial \alpha^0_{\theta} / \partial C \) analytically for the L1-SVM classifier, which are necessary in the derivative-based optimization procedure.

4.2 GDVEE Tuning System

The GDVEE tuning system combines the GD approach and the VEE criterion, of which the structure is shown in Figure 3. This tuning system is developed only for differentiable SVM classifiers. If k-fold cross-validation is employed, letting U denote the number of folds, the objective function \( O(\theta) \) for tuning of the SVM parameters is calculated by averaging the validation errors over U groups of training and validation data, where each group’s VEE is given by

\[
T(\theta) = \frac{1}{V} \sum_{(x',y') \in z'} \varphi(-y'_i \left( k\left( \sum_j \alpha^0_j y_j x_i x'_j \right) + b(\rho) \right))
\]

where \( z' \) is the set of \( V \) labeled validation samples with no intersection with the training samples in the space \( \chi \times y \), and \( j \) is the sigmoid function used to smooth the validation error instead of the step function \( \text{sgn}(\cdot) \), given as

\[
\varphi(x) = \frac{1}{1 + e^{-Ax}}
\]

where \( A \) is selected based on experimental results. Then, the derivatives of the k-fold cross validation error \( O(\theta_k) \) with respect to the \( p \)th element of the SVM parameter \( \theta \) at the \( k \)th iteration are given by

\[
\frac{\partial O(\theta_k)}{\partial \theta_{k,p}} = \frac{1}{U} \sum_{i=1}^{U} \frac{\partial T(\theta_k)}{\partial \theta_{k,p}}
\]

5 Experiments and Analysis

In the experiment, the GDVEE tuning systems were employed to tune the kernel and regularization parameters for the L2-SVM classifier. For the GDVEE tuning system, the validation data were selected by first training an L2-SVM classifier using all training samples for each group (with no restriction on the SVM parameter values, \( C = 2.4 \), and \( \sigma = 6.3 \)), then sorting the training samples based on their values of the separating function \( f(x_i) \), and finally selecting the validation data by choosing every ninth sample from the sorted training samples (1:9:512). The sigmoid function given in Eq. (20) was employed to smooth the tuning criteria of VEE, of which the factor \( A \) was set as 10 for VEE, respectively. In Figure 4, the smoothed VEE criteria are compared with the test error on the same training and test group, versus different values of one parameter with the other fixed at the selected values.

For GD tuning, the convergence-controlling parameter \( \eta \) was set as 1 for both \( C \) and \( s \) Different values of 500, 800, and 1000 were used for \( \tau \). The stopping condition was decided jointly by the iteration number, which was set as 300 in this experiment. We repeated the GD iterative procedure three or four times, each time with a different initial setting of \( \theta \), which is equivalent to finding an optimal area
by converging from different directions. The selected parameters and the corresponding L2-SVM performance using the parameter tuning GDVEE are recorded: \( \log_{10} \sigma = 0.87 \) and \( \log_{10} C = 1.42 \).

The performance accuracies for the WDBC data obtained in our experiments are compared with those obtained by the linear SVM classifier [9], the fuzzy classifier using an evolutionary scatter partition of the feature space [5], and the edited nearest-neighbor (ENN) with pure filtering [10] in Table 1.

**Table 1: Performance Comparison**

<table>
<thead>
<tr>
<th>Performance</th>
<th>Mean</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2-SVM /GDVEE (RBF)</td>
<td>98.1</td>
<td>100</td>
<td>94.1</td>
</tr>
<tr>
<td>SVM (linear)</td>
<td>94.0</td>
<td>95.0</td>
<td>N/A</td>
</tr>
<tr>
<td>SVM (RBF)</td>
<td>97.7</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>95.8</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>ENN</td>
<td>95.6</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Figure 4. Comparison of the smoothed VEE criteria and test error, versus different values of one L2-SVM parameter, and with the other fixed.

References


