**Open Access** **Baghdad Science Journal** **P-ISSN: 2078-8665**
**2020, 17(4):1255-1266** **E-ISSN: 2411-7986**

DOI: http://dx.doi.org/10.21123/bsj.2020.17.4.1255

# A Modified Support Vector Machine Classifiers Using Stochastic Gradient Descent with Application to Leukemia Cancer Type Dataset

***Ghadeer Jasim Mohammed Mahdi***

Department of Mathematics, College of Education for the Pure Science Ibn Al-Haitham, University of Baghdad, Iraq
E-mails: gmahdi@ihcoedu.uobaghdad.edu.iq , ghadeer.jasim@yahoo.com , mahdighadeer@gmail.com
ORCID ID: https://orcid.org/0000-0003-4870-4034

Received 18/10/2019, Accepted 9/2/2020, Published 1/12/2020



## Abstract:

Support vector machines (SVMs) are supervised learning models that analyze data for classification or regression. For classification, SVM is widely used by selecting an optimal hyperplane that separates two classes. SVM has very good accuracy and extremally robust comparing with some other classification methods such as logistics linear regression, random forest, k-nearest neighbor and naïve model. However, working with large datasets can cause many problems such as time-consuming and inefficient results. In this paper, the SVM has been modified by using a stochastic Gradient descent process. The modified method, stochastic gradient descent SVM (SGD-SVM), checked by using two simulation datasets. Since the classification of different cancer types is important for cancer diagnosis and drug discovery, SGD-SVM is applied for classifying the most common leukemia cancer type dataset. The results that are gotten using SGD-SVM are much accurate than other results of many studies that used the same leukemia datasets.

**Key words:** Classification, Dimension Reduction, Feature Selection, Leukemia Diagnosis, Stochastic Gradient Descend.

## Introduction:

For every subject in a given dataset, suppose information on $p$ dimensional covariate vector, $X_{p\times 1}$ exists, and a response $y$ that has two possible categories are given. In statistics, besides SVMs there are many classifier methods such as ANN (1), LDA (2), PCA (3), random forest (4), naïve Bayes (5), and NN (6). SVM is a supervised learning technique. It means a classifier based on a training dataset can be created, then that classifier can be used for future observations. The goal of the SVM is to create an algorithm, so that given information for a new observation, the category of the response can be predicted.

For example, let's define $y = +1$ if the response is in the first category, and $y = -1$ if the response is in the second category. The aim is to design a classifier rule $f(X)$ as follows,
$y = +1$ if $f(X) > 0$ and $y = -1$ if $f(X) < 0$.

This can be used to determine the response category given the covariate information. A geometric procedure is used in SVM that finds the classifier according to some optimization criterion; unlike LDA which uses a distribution for $X$ given its category (7). A linear SVM (hard margin SVM), i.e., $f(X) = \beta_0 + \beta_1 X$ where $\beta_0$ and $\beta_1$ are unknown parameters, creates a hyperplane in the $X$-space that acts as a separator between the two response categories. Linearity is a simplifying assumption, and in some cases, it may work sufficiently well; hence the non-linearity case does not need to be considered (8). However, if the data is not linearly resorbable, $f$ as non-linear SVM (soft margin SVM) should be assumed (3). Soft margin SVM is a tool for many real-world applications.

This paper is organized as follows, first, both hard and soft margins SVM are discussed. For nonlinear SVM, some types of kernels are introduced and used. Then our modified method is explained, SGD-SVM. It was tested on two simulation datasets with 50 and 100 observations. SGD-SVM is applied, in the end, on a real dataset, the most common cancer type (Leukemia dataset) (9). SGD-SVM was compared with some existing methods which are K-nearest neighbor, random forest, and naïve Bayes.

### Hard Margin SVM

The hard margin SVM is the case when two classes are linearly separable. If $y_i = \{-1, +1\}$, and $x_i \in \mathbb{R}^d$, then $D = \{(x_i, y_i)\}_{i=1}^n$ is the $n$ data points. Figure 1.a shows the case when the labels of $y$ are

linearly separable. Several hyperplanes (e.g., $H_1, H_2, H_3,$ and $H_4$) can be defined to separate the data points with two classes (10).

Our goal is to find the best separable hyperplane; i.e., the hyperplane should be in the middle of the two classes, so that the distance from the closest point on either side to the hyperplane is the same (11).

By assuming $x_1$and $x_2$ are two points that lie on the optimal hyperplane $H_0 = \beta^T \mathrm{x} + \beta_0$, then $\beta^T x_1 + \beta_0 = \beta^T x_2 + \beta_0 = 0$, $\beta^T (x_1 - x_2) = 0$. It means that $\beta^T \perp (x_1 - x_2)$. Let $x$ be any point on one side of the hyperplane, then $\beta^T x + \beta_0 = 0$ implies that $\beta_0 = -\beta^T x$. To find the distance of a point to the hyperplane ($d_i$), a point can be chosen, say $x_0$, so that

$$d_i = \frac{\beta^T (x - x_0)}{|\beta|} = \frac{\beta^T x - \beta^T x_0}{|\beta|} = \frac{\beta^T x + \beta_0}{|\beta|}$$

Since the data point can be either side, and y can take a positive and negative sign, the distance of $x_0$ to the hyperplane is $d_i y_i$. Therefore, the margin can be defined as follows,

$$\text{Margin} = min\{y_i d_i\} = min\{\frac{y_i(\beta^T x_i + \beta_0)}{|\beta|}\} \quad \dots \quad (1)$$

For any point that is not on the hyperplane, $\mathrm{y_i}(\beta^T x_\mathrm{i} + \beta_0) \geq K$. This implies that:

$\mathrm{y}_i \left(\frac{\beta^T}{K} x_i + \frac{\beta_0}{K}\right) \geq 1$; i.e., $\mathrm{y}_i\left(\beta'^T x_i + \beta_0'\right) \geq 1$ for some $\beta' = \frac{\beta^T}{K}$ and $\beta_0' = \frac{\beta_0}{K}$.

Hence, there exist some $\beta$ and $\beta_0$ such that $y_i(\beta^T x_i + \beta_0) \geq 1 \; \forall \, x_i$. Eq.(1) can be written as follows,

$$\text{Margin} = min\left\{\frac{1}{|\beta|}\right\} \quad \dots \quad (2)$$

Since the goal is to maximize the margin, $\frac{1}{2}|\beta|^2$ should be minimized such that $y_i(\beta^T x_i + \beta_0) \geq 1$. This constrained optimization problem is called Quadratic Programming (QP) problem. It is named according to the quadratic objective function (12). The Lagrange multiplier method can be used to solve this QP problem as follows,

$$L_p(\beta, \beta_0, \alpha_i) = \frac{1}{2}|\beta|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i(x_i^T \beta + \beta_0)) \quad \dots \quad (3)$$

where $\alpha_i \geq 0$ is the Lagrange coefficients for $i = 1, \dots, n$. Let

$\frac{\partial}{\partial \beta} L_p(\beta, \beta_0) = 0$ and $\frac{\partial}{\partial \beta_0} L_p(\beta, \beta_0) = 0$

Therefore, above forms leads to,

$\beta = \sum_{j=1}^{n} \alpha_j y_j x_j$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

Substituting these two forms into Eq.(3) (the primal problem), the following dual problem is gotten,

$$maximize \; L_d(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad \dots (4)$$

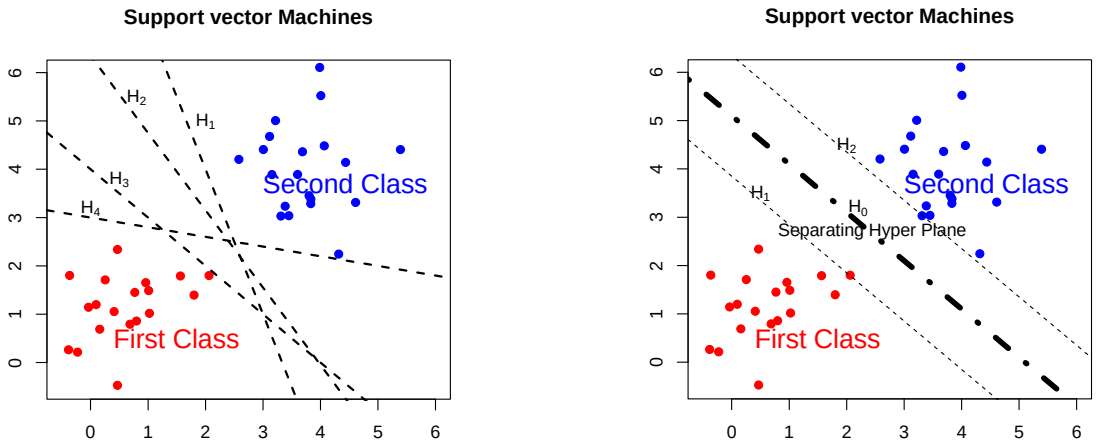
subject to $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$. $L_d$ is the greatest lower bound (infimum) of $L_p$ for all $\beta$ and $\beta_0$; i.e., the dual problem is related to the prime problem by $L_d = \inf L_p(\beta, \beta_0, \alpha)$. Solving Eq.(4), $\alpha_i$ can be gotten, from which $\beta$ of the optimal plane can be found.

In Fig.1b, the points $x_i$ on either of the two boundary hyperplanes $H_1$ and $H_2$ are called support vectors, and they correspond to positive Lagrange multipliers $\alpha_i > 0$. All the points that are far away from the $H_1$ and $H_2$ are not important. So, the training model will depend only on the support vectors. For a support vector $x_i$ on either $H_1$ and $H_2$, the constraining condition is

$$y_i(x_i \beta + \beta_0) = 1 \quad \dots \quad (5)$$

where $i \in \delta$, and $\delta$ is the set of all indices of support vectors $x_i$ that corresponding to $\alpha_i > 0$. Substituting $\beta = \sum_{j=1}^{n} \alpha_j y_j x_j = \sum_{j \in \delta} \alpha_j y_j x_j$ into Eq.(5), it becomes

$$y_i \left( \sum_{j \in \delta} \alpha_j y_j x_i^T x_j + \beta_0 \right) = 1 \quad \dots \quad (6)$$




a. Hyper planes that separate first and second classes ( $H_1, H_2, H_3, and\ H_4$).

b. Separating hyperplane ($H_0$) and two boundary hyperplanes ($H_1\ and\ H_2$).

**Figure 1. Classification by Support Vector Machine**

By simplifying Eq.(6),

$$y_i \sum_{j\in\delta} \alpha_j y_j x_i^T x_j = 1 - y_i \beta_0 \quad \dots (7)$$

For the optimal values of $\beta$ and $\beta_0$, define

$$|\beta|^2 = \beta^T \beta = \sum_{i\in\delta} \alpha_i y_i x_i^T \sum_{j\in\delta} \alpha_j y_j x_j$$

$$= \sum_{i\in\delta} \alpha_i y_i \sum_{j\in\delta} \alpha_j y_j\, x_i^T x_j$$

$$= \sum_{i\in\delta} \alpha_i (1 - y_i \beta_0) = \sum_{i\in\delta} \alpha_i - \beta_0 \sum_{i\in\delta} \alpha_i y_i$$

Since $\sum_{i=1}^{n} \alpha_i y_i = 0$, $|\beta|^2$ equals to $\sum_{i\in\delta} \alpha_i$. So, the margin, the distance between $H_1$ (or $H_2$) and the optimal decision hyperplane $H_0$, is

$$\frac{1}{|\beta|} = \frac{1}{\sqrt{\sum_{i\in\delta} \alpha_i}} \quad \dots (8)$$

**Soft Margin SVM**

The condition of the optimal hyperplane can be relaxed by including an extra term, when the response classes are not linearly separable as follows,

$$y_i(\beta^T x_i + \beta_0) \geq 1 - \eta_i, \qquad \forall i = 1, \dots, n.$$

To get a minimum error, $\eta_i \geq 0$ should be minimized as well as $|\beta|$, so the objective function becomes,

$$minimize\ |\beta|^2 + +K \sum_{i=1}^{n} \eta_i^s \quad \dots (9)$$

subject to $y_i\left(x_i^T \beta + \beta_0\right) \geq 1 - \eta_i$ and $\eta_i \geq 0$ where $i = 1, \dots, n$. $K$ is a regularization parameter that controls the agreement between minimizing the training error and maximizing the margin. Small $K$ influence to emphasize the margin while ignoring the outliers in the training dataset $D$, while large $K$ may tend to overfit the training dataset $D$. When $s = 1$, Eq.(9) is called *first norm soft margin*, and when $s = 2$, it is called *second norm soft margin*. The algorithm based on the first norm soft margin is less sensitive to outliers in the training dataset where it ignores the outliers. In the next section, the first and second norm soft margins are going to be discussed.

**First Norm Soft Margin**

From Eq.(9), when $s = 1$, it follows that:

$$minimize\ |\beta|^2 + +K \sum_{i=1}^{n} \eta_i \quad \dots (10)$$

Subject to $y_i\left(x_i^T \beta + \beta_0\right) \geq 1 - \eta_i$ and $\eta_i \geq 0, \quad \forall\ i = 1, \dots, n.$

And hence, the prime Lagrangian becomes

$$L_p(\beta, \beta_0, \eta, \alpha, \gamma) = \frac{1}{2}|\beta|^2 + K \sum_{i=1}^{n} \eta_i - \sum_{i=1}^{n} \alpha_i [y_i(\beta^T x + \beta_0) - 1 + \eta_i] - \sum_{i=1}^{n} \gamma_i \eta_i \quad \dots (11)$$

with $\alpha_i \geq 0$ and $\gamma_i \geq 0$.

Substituting

$$\frac{\partial L}{\partial \beta} = \beta - \sum_{i=1}^{n} y_i \eta_i x_i = 0; \qquad \frac{\partial L}{\partial \eta} = K\eta - \alpha = 0; \quad \frac{\partial L}{\partial \beta_0} = \sum_{i=1}^{n} y_i \alpha_i = 0 \quad \dots (12)$$

in to Eq.(11), the following dual problem is gotten

Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986

$$\begin{aligned} &maximize\ L_d(\alpha,\gamma) \\ &= \sum_{i=1}^{n} \alpha_i \\ &- \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j\, \alpha_i \alpha_j x_j^T x_i \\ &- \sum_{i=1}^{n} \alpha_i \eta_i - \sum_{i=1}^{n} \gamma_i \eta_i \\ &+ K \sum_{i=1}^{n} \eta_i \end{aligned} \quad \ldots(13)$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j\, \alpha_i \alpha_j x_j^T x_i$$

subject to $0 \le \alpha_i \le K, \sum_{i=1}^{n} \alpha_i y_i = 0$.

Solving Eq.(13) for $\alpha_i$, the optimal decision hyperplane, $\beta$ and $\beta_0$, with the margin becomes

$$\frac{1}{|\beta|} = \frac{1}{\sqrt{\sum_{i\in\delta}\sum_{j\in\delta} \alpha_i \alpha_j y_i y_j x_i^T x_j}}$$

**Second Norm Soft Margin**

By plugging $s = 2$ in Eq.(9),

$$minimize\ |\beta|^2 + + K \sum_{i=1}^{n} \eta_{i^2} \quad \ldots(14)$$

subject to $y_i\left(x_i^T \beta + \beta_0\right) \ge 1 - \eta_i, \ \forall \quad i = 1, \ldots, n$. Notice that the condition $\eta_i \ge 0$ is dropped and set $\eta = 0$ if it is less and equal to 0; hence the objective function is further reduced. In this case, the prime Lagrangian is

$$\begin{aligned} &L_p(\beta, \beta_0, \eta, \alpha, \gamma) \\ &= \frac{1}{2}|\beta|^2 \\ &+ \frac{K}{2}\sum_{i=1}^{n} \eta_i^2 \\ &- \sum_{i=1}^{n} \alpha_i [y_i(\beta^T x + \beta_0) \\ &- 1 + \eta_i] \end{aligned} \quad \ldots(15)$$

Substituting the condition that given in Eq.(12) into Eq.(15), the following dual problem will be gotten

$$\begin{aligned} &maximize\ L_d(\alpha) \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j\, \alpha_i \alpha_j x_j^T x_i \\ &- \frac{1}{2K}\sum_{i=1}^{n} \alpha_i^2 \end{aligned} \quad \ldots(16)$$

subject to $\eta_i \ge 0, \ \sum_{i=1}^{n} \alpha_i y_i = 0$. Solving Eq.(16) for $\alpha_i$, the optimal values for $\beta$ and $\beta_0$ with the margin becomes

$$\frac{1}{|\beta|} = \frac{1}{\sqrt{\sum_{i\in\delta} \alpha_i - \frac{1}{K}\sum_{i\in\delta} \alpha_i^2}}$$

**Karush-Kuhn-Tucker (KKT) conditions**

In the previous sections, the cases where the datasets are linearly separable are discussed, so the solution has been found by solving the dual form of Lagrangian. This can be done by minimizing a quadratic function subject to a set of constraints. To find the dual objective function, the following conditions (KKT conditions) should be satisfied: i. stationarity, ii. dual feasibility, iii. complementary slackness, and iv. primal feasibility (13). To minimize $f(x)$ subject to the constraint $g_i(x) \ge 0\ \forall x$, then the Lagrangian function becomes $L(x, \alpha_i) = f(x) - \sum_i \alpha_i g_i(x)$. If $x^\star$ is a point where $\beta$ is optimal for our cost function, the necessary KKT conditions for $x^\star$ to be the local minimum are i. Stationarity: $\frac{\partial L}{\partial x}(x^\star) = 0$, ii. Dual Feasibility: $\alpha_i \ge 0$, iii. Complementary Slackness: $\alpha_i g_i(x^\star) = 0$, and iv. Primal Feasibility: $g_i(x^\star) \ge 0$. The primal function is not convenient if any of the KKT conditions is not satisfied. In general, if the dataset has $N$ variables, the computational complexity is $O(N^3)$ (14).

**Non-linearly SVM**

When a dataset is not linearly separable, the techniques introduced in the previous section do not converge. However, a hyperplane (decision surface) can be found by transforming the data to the high dimensional spaces by using an appropriate kernel. Even though the original dataset is not linearly separable, a hyperplane can be found to separate the mapped datasets (12). Fig.2 shows an example of transforming data points from two-dimensional (2D) spaces to three-dimensional (3D) spaces. In the 2D, the data points are not linearly separable, but the data points are easily can be separated by a surface when they are transformed into 3D.

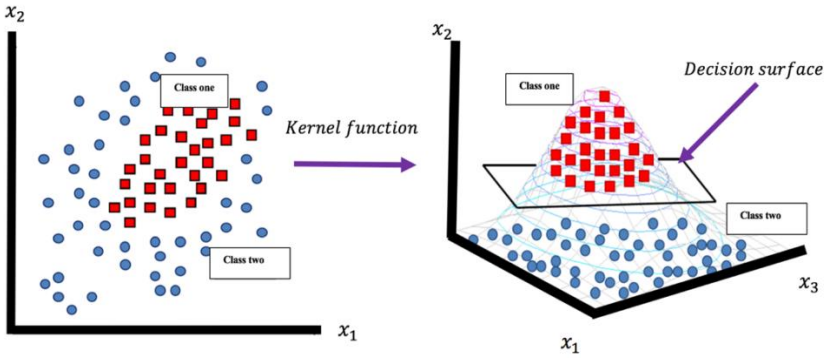



**Figure 2. A kernel function that transfers dataset from 2D to 3D.**

If $\Phi$ is a kernel function that satisfies $\Phi(x_i, x_j) = \phi^T(x_i).\phi(x_j)$, then the corresponding dual problem is

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_i, x_j) \quad \ldots(17)$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$, and $\alpha_i \geq 0 \ \forall i = 1, \ldots, n$. In terms of the unknown parameters, the function $L(\alpha)$ is convex and quadratic, so the problem can be solved through QP as shown in the previous sections (15). Moreover, the set of KKT conditions for Eq.(17) give the following decision rule,

$$L(x, \alpha_i^\star, \beta_0) = \sum_{i=1}^{N_\delta} y_i \alpha_i^\star \Phi(x_i, x) + \beta_0 \quad \ldots(18)$$

where $N_\delta$ denotes to the number of support vectors, and $\alpha_i$ denotes to the non-zero Lagrange multipliers that corresponding to the support vectors. The most common choices of the kernel are:

1. Linear kernel: $\Phi(x_i, x_j) = x_i^T x_j$
2. Polynomial kernel: $\Phi(x_i, x_j) = (1 + x_i^T x_j)^p$
3. Radial Basis Function (RBF) (or Gaussian) kernel: $\Phi(x_i, x_j) = exp\{-\frac{(x_i - x_j)^T (x_i - x_j)}{2\sigma^2}\}$
4. Multi-Layer Perceptron (MLP) (or Sigmoid) kernel: $\Phi(x_i, x_j) = \tanh(k_1 x_i^T x_j + k_2)$

In the implicit mapping of the data to feature space, kernel functions that satisfy Mercer's conditions ensure the convexity of the $L(\alpha)$ which leads to the unique optimum (16). Mercer condition state that a kernel function $\Phi(x, y)$ must be continuous, symmetric, and positive semi-definite, so the matrices do not have non-negative eigenvalues (17) (18).

**Gradient Descent**

The goal is to minimize the following function,

$$L(\beta) = \frac{1}{2}\beta_0^T \beta_0 + K \sum_{i} \max(0, 1 - y_i \beta^T x_i) \quad \ldots(19)$$

Eq.(19) is convex in $\beta$, and it is a quadratic optimization problem. In previous methods, the technique from QP is used, but it is very slow. Whenever there are no constraints, the gradient descent (GD) can be used (19). In general, the gradient is the direction of the steepest increase in the function, so it goes in the opposite direction to get to the minimum as shown in Fig.3.

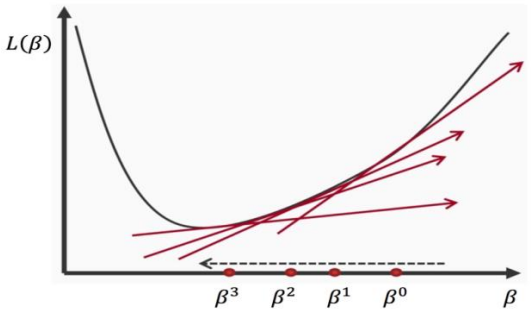



**Figure 3. The general strategy for minimizing a function L(β)**

The general GD strategy for minimizing Eq.(19) starts with an initial value for $\beta$, say $\beta^0$, then iterate until convergence. GD-SVM is faster than QP, and it is still slow. The GD-SVM in Algorithm 1 can be summarized as follows:

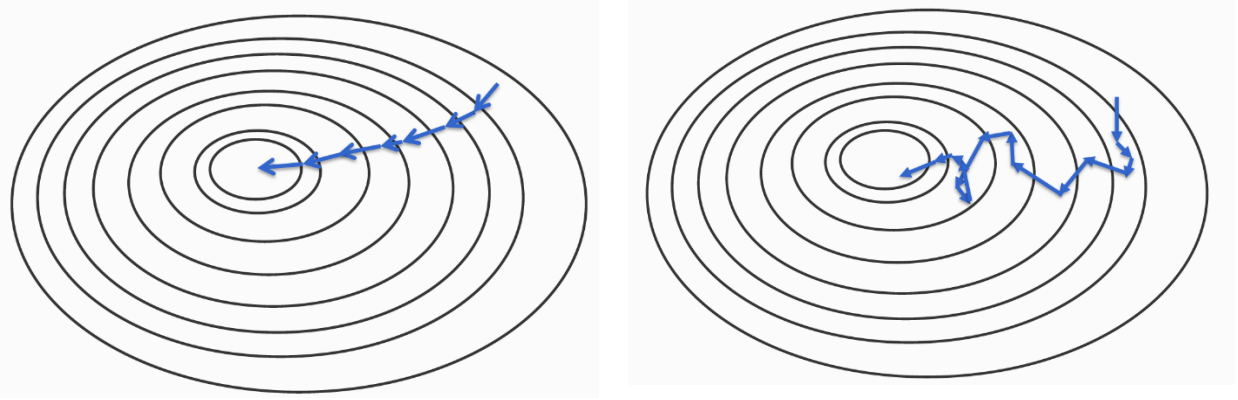
**Algorithm 1: GD-SVM**

Iterate until convergence
Initialize $\beta^0$
For $i = 1, \ldots, d$:
Evaluate: $\nabla(\beta^j) = \frac{\partial f(\beta, \beta^0)}{\partial \beta^j} = \beta^j + K \sum_{i=1}^{n} \frac{\partial L(x_i, y_i)}{\partial \beta^j}$, where $K$ is a regularization parameter.
Update $\beta$ as follows: $\beta^j \leftarrow \beta^j - \eta \nabla(\beta^j)$, where $\eta$ is the learning rate parameter.
Return final $\beta$.

Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986

**Stochastic Gradient Descent**

Computing $\nabla(\beta^j)$ takes $O(n)$ time, where $n$ is the size of the training dataset, so GD-SVM is slow when $n$ is large. In the GD-SVM algorithm, the value of the objective function is improved at every step. It takes fewer steps to converges, as can be shown in Fig.4., but each step takes much longer time to be computed (20). To speed the GD-SVM algorithm, the gradient is evaluated for each training example instead of evaluating it for all examples. This process is called stochastic gradient descent SVM (SGD-SVM). It improves the value of the objective function noisily. This process takes many more updates than gradient descent, but each update is less computationally expensive, and hence SGD-SVM is faster than GD-SVM. SGD-SVM algorithm (Algorithm 2) is guaranteed to converge to the minimum of $J$ if $\gamma_t$ is small enough (21) (22).



a. Gradient Descent b. Stochastic Gradient Descent

**Figure 4. Gradient Descent vs. Stochastic Gradient Descent.**

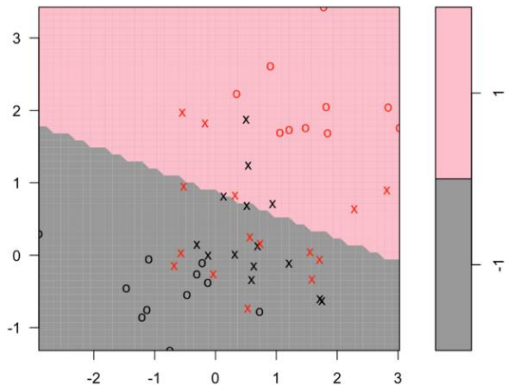
**Algorithm 2: SGD-SVM**

Given a training set $S = \{(x_i, y_i): x \in \mathbb{R}^n \; and \; y \in \{-1, +1\}\}$
Initialize: $\beta^0 = 0 \in \mathbb{R}^n$
For $t = 1, \ldots, T$:
Pick a random example $(x_i, y_i)$ from the training set $S$.
Repeat $(x_i, y_i)$ to make a full dataset and take the derivative of the SVM objective at the current $\beta^{t-1}$ to be $\nabla \mathrm{J}^{\mathrm{t}}(\beta^{t-1})$.
$\nabla \mathrm{J}^{\mathrm{t}}(\beta^t) = \frac{1}{2}\beta_0^T\beta_0 + K \cdot N \cdot \max(0, 1 - y_i\beta^T x_i)$, where $N$ is the number of training examples.
Update $\beta$ as follows: $\beta^t \leftarrow \beta^{t-1} - \gamma_t \nabla \mathrm{J}^{\mathrm{t}}(\beta^{t-1})$.
Return final $\beta$.

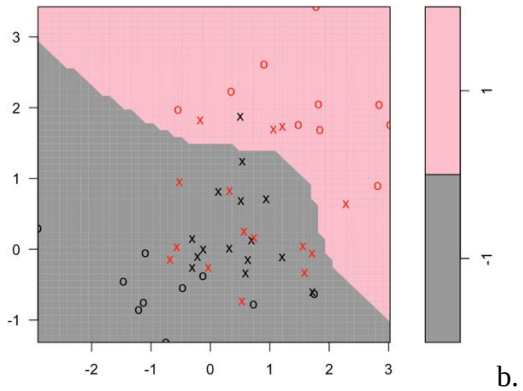
**Simulation Studies**

To test the GD-SVM and SGD-SVM methods, two simulation datasets are generated with 50 and 100 observations. Each dataset has a set of variables and a response that has two classes. In real life, it is unknown whether the datasets are linearly or nonlinearly separable. So, as can be shown in Fig.5 and Fig.6, complex datasets (nonlinearly separable) are generated. GD-SVM and SGD-SVM are applied to the two datasets by using different types of kernels. In both methods, the RBF kernel gives the best accuracy with two datasets. In Tables 1 and 2, GD-SVM and SGD-SVM are compared concerning the best value of $K$, the number of support vectors, using the sensitivity (also called the true positive rate) and specificity (also called the true negative rate).

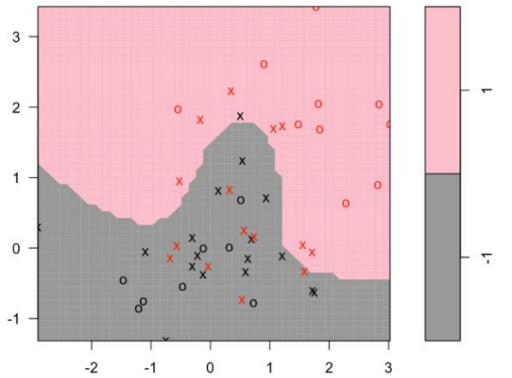
Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986



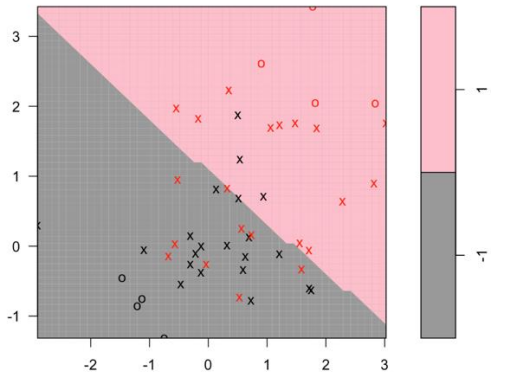
a. Linear kernel



b. Polynomial kernel

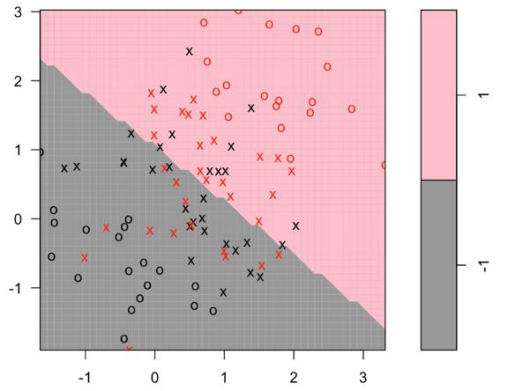


c. RBF kernel

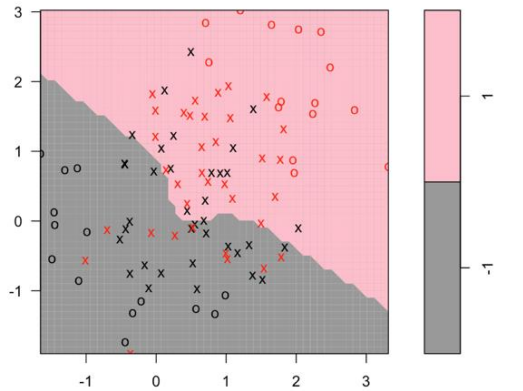


d. MLP kernel

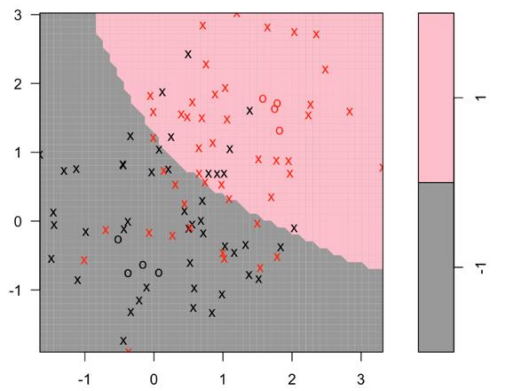
**Figure 5. SGD-SVM for a data set with 50 observations.**



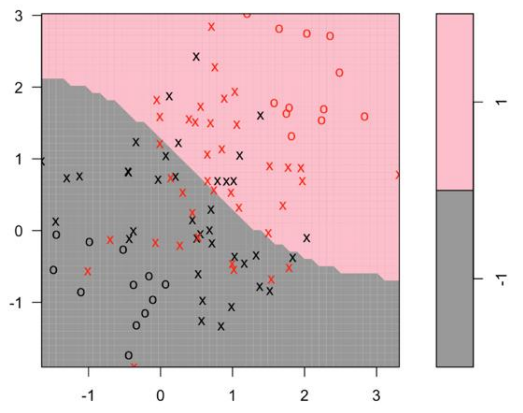
a. Linear kernel



b. Polynomial kernel



c. RBF kernel



d. MLP kernel

**Figure 6. SGD-SVM for a data set with 100 observations.**

**Table 1. GD-SVM & SGD-SVM for a data set with 50 observations.**

| Method | Kernel Type | Best *K* Value | Number of Support Vectors | Sensitivity % | Specificity % |
|---|---|---|---|---|---|
| GD-SVM | Linear | 5 | 20 | 50.33 | 63.49 |
| | Polynomial | 1 | 25 | 68.40 | 90.58 |
| | **RBF** | **10** | **39** | **71.44** | **94.01** |
| | MLP | 1 | 36 | 68.95 | 74.33 |
| SGD-SVM | Linear | 5 | 29 | 66.66 | 75.00 |
| | Polynomial | 5 | 30 | 68.57 | 93.33 |
| | **RBF** | **10** | **32** | **75.00** | **94.44** |
| | MLP | 0.1 | 42 | 72.41 | 80.95 |

**Table 2. GD-SVM & SGD-SVM for a data set with 100 observations.**

| Method | Kernel Type | Best *K* Value | Number of Support Vectors | Sensitivity % | Specificity % |
|---|---|---|---|---|---|
| GD-SVM | Linear | 5 | 33 | 60.30 | 69.43 |
| | Polynomial | 1 | 60 | 68.40 | 83.58 |
| | **RBF** | **1** | **79** | **72.74** | **74.01** |
| | MLP | 0.1 | 40 | 70.95 | 74.33 |
| SGD-SVM | Linear | 1 | 62 | 75.00 | 77.77 |
| | Polynomial | 1 | 71 | 77.77 | 72.72 |
| | **RBF** | **0.1** | **92** | **74.54** | **80.00** |
| | MLP | 0.1 | 76 | 72.54 | 80.00 |

**Real Dataset**

One of the universal cancer types is Leukemia dataset. Its diagnosis and classification are complex. For experimental evaluation, The Leukemia dataset is used in this section. The dataset was published by Golub et al in 1999 (9). It comes from a proof of concept study. It shows how the gene expression monitoring (via a DNA microarray) can classify the new cases of cancer, providing a common approach for assigning tumors to known classes (11). Using this type of data set, patients can be classified into Acute Myeloid Leukemia (AML) and Acute Lymphoblastic Leukemia (ALL) (13)(23). The complete Golub-Merge dataset is available in the golubEsets packages. It has 3051 genes and 72 observations. Working with large datasets confront many difficulties such as time-consuming and inefficient results (24).

**Analyzing Golub Datasets**

To analyze the Leukemia dataset, the most significant genes for cancer type need to be selected. This means the genes that are differentially expressed across classes should be considered. Since the differentially expressed genes between two groups, a t-test seems like the common choice. However, the t-test requires normality assumption which might not be a logical assumption. A $3051 \times 2$ histogram cannot be plotted to have an idea about the justification for normality. Mann Whiteny U test looks more adequate in this case (25), and it is nearly as efficient as the t-test.

After running the test and adjusting p-values according to Benjameni-Hochberg method, it is ended up with only 329 significant genes. The majority of genes does not seem to have different mean values across classes. The same result for the median is gotten. The median differences between classes are clustered around zero, Fig.8. Only a few numbers of genes seem interesting, and they can be easily indicated. Fig.7 provides a quick summary of the selected genes (26). Some important genes in Fig. 9 were plotted. As can be seen, the most significant genes id that influences the model are G4847, G3252, G1882, G6855, and G2288.

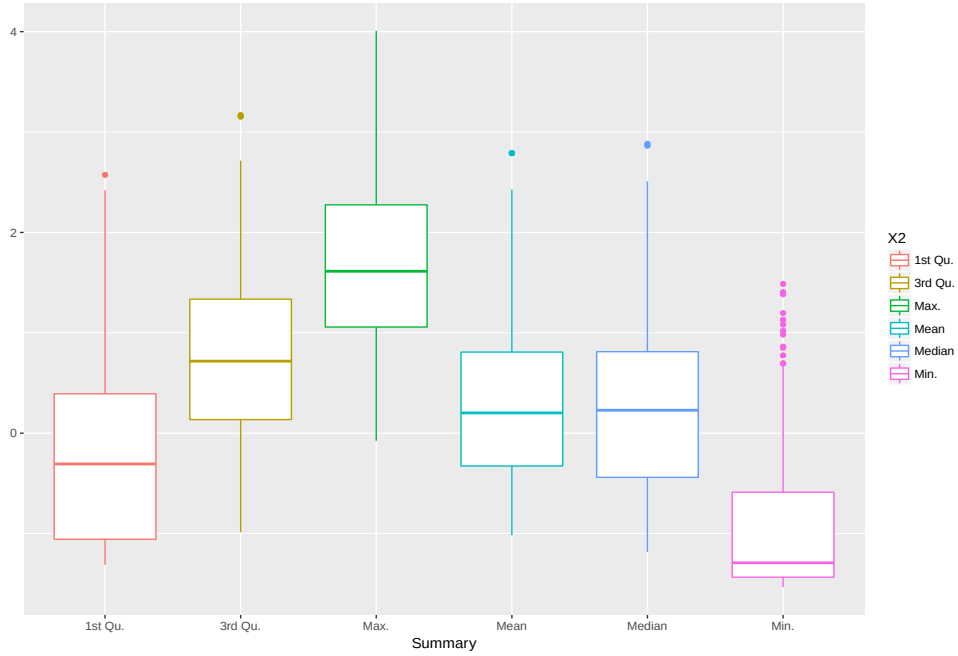



**Figure 7. Summary of selected genes.**

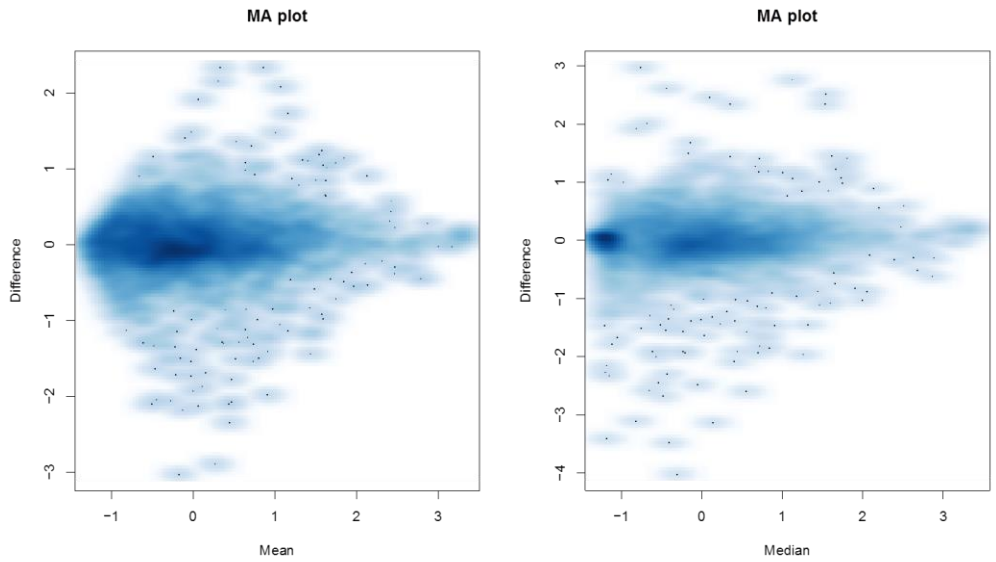



**Figure 8. Plot for the differences in Mean and Median.**

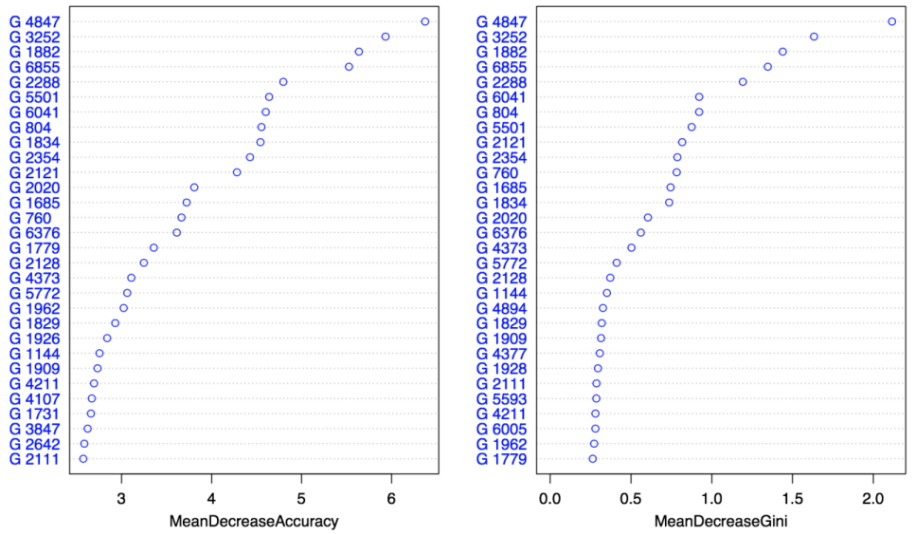



**Figure 9. Importance genes for Leukemia dataset.**

## Results and Discussion:

SGD-SVM was applied to the Leukemia dataset. A comparison between some different kernel functions had been done. The most common kernel which are linear, polynomial, RBF and MLP kernels were used. Also, our method was compared with some existing methods. These methods used the same dataset for leukemia classification which are k-nearest neighbor random forest and naïve Bayes (13). In Fig.10, SGD-SVM models for the Leukemia dataset classification are plotted. Only the most two important genes in which their id are G3252 and G4847 are used. These two genes expressions are relevant to judge if a new sample related to ALL or AML classes. As it is shown from the plot (Fig.10) and the table (Table 3) the best SGD-SVM model is satisfied when the RBF kernel is applied. The RBF kernel gets 100% accuracy which is the highest performance compared with other kernels. SGD-SVM performed 96.93% accuracy for the linear kernel, 97.00% accuracy for the polynomial kernel, and 97.00% accuracy for MLP kernel. SGD-SVM performed much better than other methods where k-nearest neighbor performed 86.80% accuracy, random forest performed 87.10% accuracy, and naïve Bayes performed 84.6% accuracy.

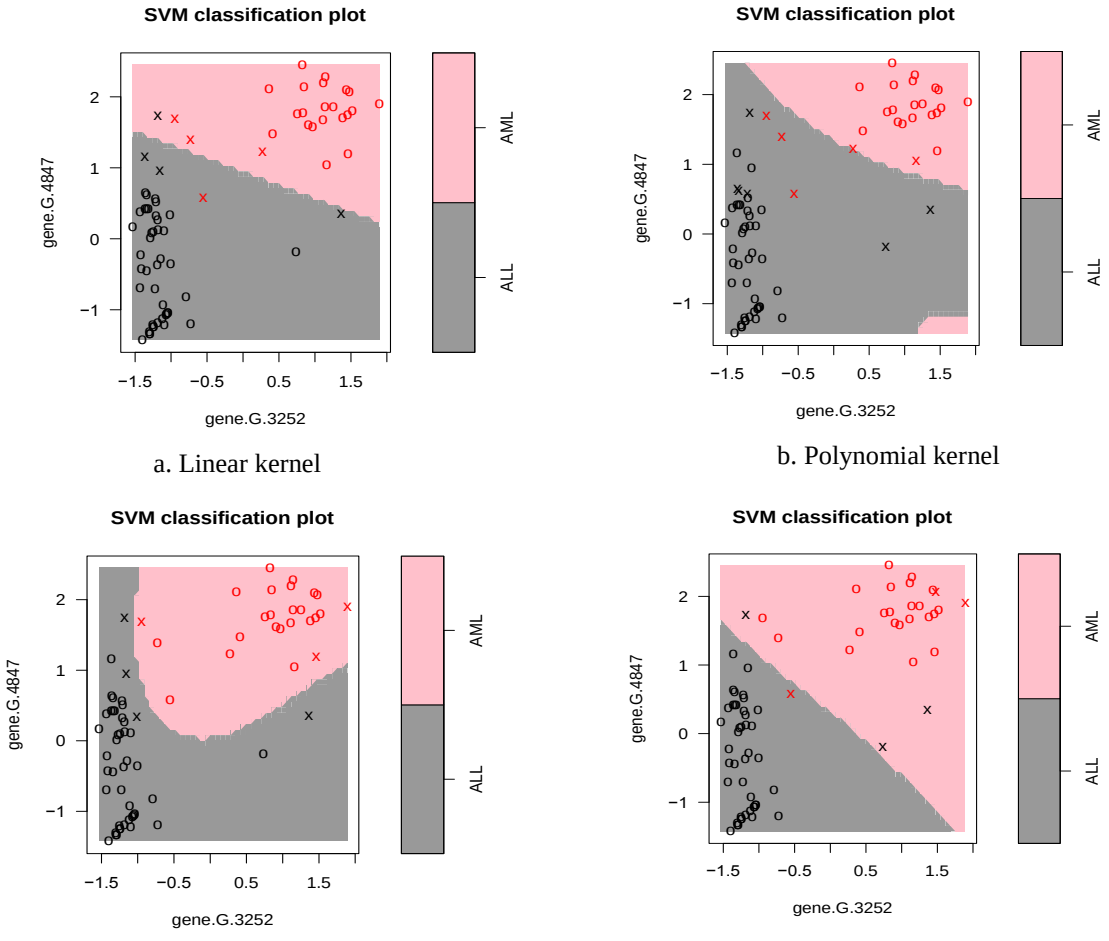



**Figure 10. Classification Leukemia Cancer type using SGD-SVM**

**Table 3. Comparison between SGD-SVM, k-nearest neighbors, random forest, and naive Bayes for classification leukemia cancer type.**

| Methods | | Number of Support Vectors | Accuracy Rate % | Sensitivity % | Specificity % |
|---|---|---|---|---|---|
| SGD-SVM | Linear kernel | 8 | 96.93 | 97.87 | 96.00 |
| | Polynomial kernel | 11 | 97.00 | 94.00 | 1 |
| | **RBF kernel** | **7** | **1** | **1** | **1** |
| | MLP kernel | 6 | 93.33 | 97.77 | 88.88 |
| k-nearest neighbors | | | 86.80 | 85.20 | 88.40 |
| random forest | | | 87.10 | 89.33 | 84.87 |
| naive Bayes | | | 84.60 | 86.49 | 82.71 |

## Conclusion:

In this paper, the SGD-SVM method was proposed. The method was developed using a stochastic Gradient descent process. Two simulation datasets were used to test the performance of the method. The results showed that SGD-SVM has a larger accuracy rate comparing with the regular SVM method. By applying the SGD-SVM on Leukemia datasets, it found that the best accuracy exists for classification of the two types of Leukemia cancer when the RBF kernel has been applied.

## Author's declaration:

- Conflicts of Interest: None.
- I hereby confirm that all the Figures and Tables in the manuscript are mine. Besides, the Figures and images, which are not mine, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Baghdad.

## References:

1. Bala R, Kumar DD. Classification Using ANN: A Review. IJCIR. 2017;13(7):1811-20.
2. Okwonu FZ, Othman AR. A Model classification technique for linear discriminant analysis for two groups. IJCSI. 2012 May 1;9(3):125
3. Barshan E, Ghodsi A, Azimifar Z, Jahromi MZ. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. Pattern Recognition. 2011 Jul 1;44(7):1357-71.
4. Liaw A, Wiener M. Classification and regression by randomForest. R news. 2002 Dec 3;2(3):18-22.
5. Karim M, Rahman RM. Decision tree and naive bayes algorithm for classification and generation of actionable knowledge for direct marketing. IJSEA. 2013 Apr 25;6(04):196.
6. Guo G, Wang H, Bell D, Bi Y, Greer K. KNN model-based approach in classification. In OTM Confederated International Conferences. On the Move to Meaningful Internet Systems. 2003 Nov 3 (pp. 986-996). Springer, Berlin, Heidelberg.
7. Jain R. Simple tutorial on svm and parameter tuning in python and r, 2017. URL https://www. hackerearth. com/blog/machine-learning/simple-tutorial-svm-parameter-tuning-python-r/. Visited. 2018;20.
8. Mahdi, Ghadeer J. Hierarchical Bayesian Regression with Application in Spatial Modeling and Outlier Detection. Diss. University of Arkansas, Fayetteville, 2018.
9. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. 1999 Oct 15;286(5439):531-7.
10. Ad'hiah AH, Mahmood AS, Al-Kazaz AK, Mayouf KK. Gene Expression and Polymorphism of Interleukin-4 in a Sample of Iraqi Rheumatoid Arthritis Patients. Baghdad Sci. J. 2018;15(2):130-7.
11. Zhi J, Sun J, Wang Z, Ding W. Support vector machine classifier for prediction of the metastasis of colorectal cancer. Int J Mol Med. 2018 Mar 1;41(3):1419-26.
12. Mathiasen A, Larsen KG, Grønlund A. Optimal Minimal Margin Maximization with Boosting. InInternational Conference on Machine Learning 2019 May 24 (pp. 4392-4401).
13. Zararsiz G, Elmali F, Ozturk A. Bagging support vector machines for leukemia classification. IJCSI. 2012 Nov 1;9(6):355.
14. Huang S, Cai N, Pacheco PP, Narrandes S, Wang Y, Xu W. Applications of support vector machine (SVM) learning in cancer genomics. GPB. 2018 Jan 1;15(1):41-51.
15. Thang PQ, Thuy NT, Lam HT. A modification of solution optimization in support vector machine simplification for classification. In Information Systems Design and Intelligent Applications 2018 (pp. 149-158). Springer, Singapore.
16. Salman AN, Taha TA. On Reliability Estimation for the Exponential Distribution Based on Monte Carlo Simulation. IHJPAS. 2018 Apr 25:409-19.
17. Tawfiq LN, Rashid TA. On Comparison Between Radial Basis Function and Wavelet Basis Functions Neural Networks. IHJPAS. 2017 May 24;23(2):184-92.
18. Zaidan AA, Atiya B, Bakar MA, Zaidan BB. A new hybrid algorithm of simulated annealing and simplex downhill for solving multiple-objective aggregate production planning on fuzzy environment. NCA. 2019 Jun 1;31(6):1823-34.
19. Huang X, Zhang L, Wang B, Li F, Zhang Z. Feature clustering-based support vector machine recursive feature elimination for gene selection. APPL INTELL. 2018 Mar 1;48(3):594-607.
20. Sopyła K, Drozda P. Stochastic gradient descent with Barzilai–Borwein update step for SVM. Information Sciences. 2015 Sep 20; 316:218-33.
21. Lopes FF, Ferreira JC, Fernandes MA. Parallel Implementation on FPGA of Support Vector Machines Using Stochastic Gradient Descent. Electronics. 2019 Jun;8(6):631.
22. Patwary MK, Haque MM. A Semi-Supervised Machine Learning Approach Using K-Means Algorithm to Prevent Burst Header Packet Flooding Attack in Optical Burst Switching Network. Baghdad Sci. J. 2019;16(3 Supplement):804-15.
23. Pandiyan V, Caesarendra W, Tjahjowidodo T, Tan HH. In-process tool condition monitoring in compliant abrasive belt grinding process using support vector machine and genetic algorithm. J. Manuf. Process. 2018 Jan 1; 31:199-213.
24. Okwonu FZ, Othman AR. A Model classification technique for linear discriminant analysis for two groups. IJCSI. 2012 May 1;9(3):125.

25. MacFarland TW, Yates JM. Introduction to nonparametric statistics for the biological sciences using R. Cham: Springer; 2016 Jul 6.
26. Aytug H. Feature selection for support vector machines using Generalized Benders Decomposition. Eur. J. Oper. Res. 2015 Jul 1;244(1):210-8.

# تطوير شعاع الدعم الالي للتصنيف باستخدام الانحدار العشوائي مع تطبيقات على بيانات سرطان الدم

**غدير جاسم محمد مهدي**

قسم الرياضيات، كلية التربية للعلوم الصرفة - ابن الهيثم ، جامعة بغداد، بغداد، العراق.

**الخلاصة:**

شعاع الدعم الالي **(**SVM**)** هو أحد تطبيقات معادلة الانحدار للتعليم الاستنتاجي الذي يحلل البيانات ويستخدم في التصنيف ومعادلة الانحدار. في التصنيف، يستخدم SVM بشكل واسع بأختيار مقطع مثالي للفصل بين مجموعتين. وهو يمتلك دقة عالية و مستقر بصورة هائلة بالمقارنة مع طرق التصنيف الأخرى مثل الانحدار اللوجستي الخطي، random forest، k-nearest neighbor و naïve model .على أي حال، عند العمل على بيانات هائلة تتولد مشاكل كبيرة كاستهلاك للوقت وأيضا النتائج تكون غير دقيقة. في هذا البحث SVM طورت بأستخدام طريقة الانحدار العشوائي. الطريقة المحدثة، SGD-SVM اختبرت بأستخدام مجموعتين من البيانات. ولأن تصنيف أنواع السرطان مهم بالنسبة لتشخيص السرطان واستكشاف الدواء. SGD-SVM طبقت لتصنيف بيانات تكسر كريات الدم الشهيرة. النتائج التي حصلنا عليها من طريقة SGD-SVM كانت دقتها اعلى من النتائج التي تم الحصول عليها من بعض الدراسات السابقة التي استخدمت نفس البيانات.

**الكلمات المفتاحية:** التصنيف، تقليل الأبعاد، اختيار الميزات، تشخيص سرطان الدم، الانحدار العشوائي.