Research Inventy: International Journal of Engineering And Science
Vol.9, Issue 1 (Jan 2019), PP -01-18
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Design and Development of an Efficient Data Mining Algorithm for Ranking Problems

Prof.Er.Dr.G.Manoj Someswar
DEAN (R&D), Global Research Academy – Scientific & Industrial Research Organisation [Autonomous], (Approved by Ministry of Science & Technology, Govt. of India), Hyderabad, Telangana State, India.

ABSTRACT: *Order is the way toward finding (or preparing) an arrangement of models (or capacities) that portray and recognize information classes or ideas. That is to be ready to utilize the models to foresee the obscure class marks of cases.*
We manage the positioning issue in this proposal. The positioning issue is an exceptional instance of the classification issue, where the class marks are positions or appraisals, spoke to by numbers from 1 to q. The positioning issue can likewise be given a role as the way toward preparing a rank-forecast show that appoints each example a rank that is as close as could be expected under the circumstances "to the occurrence's genuine rank. Well known uses of the positioning issue incorporate positioning the significance of website pages, assessing the financial credit of a man, and positioning the dangers of speculations.
Two mainstream groups of strategies to take care of positioning issues are Multi-Criteria Decision Aid (MCDA) techniques and Support Vector Machines (SVMs). The execution of effective MCDA techniques, for example, Utilities Additives Discriminates (UTADIS) and Generalized Utilities Additives Discriminates (GUTADIS), is accomplished by abusing the foundation information that de-copyists the relationships between are the qualities and the positions. Lamentably, the foundation information is case-subordinate, thus it is probably going to be unveil capable, vague or difficult to be demonstrated by and by. This limits the application of MCDA techniques. SVMs, rather, don't require any foundation learning. Their great execution is accomplished by keeping balance between limiting the observational misfortune and augmenting the partition edge. Normally, a multi-class Support Vector Machine Classifier is developed by combining a few twofold Support Vector Machine Classifiers. In the SVM-based approach the positioning data isn't utilized.
This proposition endeavors to build an efficient calculation for positioning issues. We look at the properties of existing calculations for positioning problems and propose a half breed calculation that joins the multi-class SVM (M-SVM) and the UTADIS demonstrate. In the new calculation, the double SVM classifiers are joined into a multi-class classifier in light of the fluffy voting strategy. The ideal fluffy voting technique is sought by settling a Linear Program (LP). The new calculation is called Fuzzy Voting based Support Vector Ranking (FVSVR) technique. We likewise expand the possibility of Fuzzy Voting from positioning issues to nonexclusive multi-class classification issues, which prompts a Fuzzy Voting based Support Vector Machine (FVSVM) strategy. The benefits of FVSVR and FVSVM are exhibited by trial comes about in view of a few databases of handy classification issues.
Keywords: *Utilities Additives Discriminates (UTADIS), Fuzzy Voting based Support Vector Ranking (FVSVR), Fuzzy Voting based Support Vector Machine (FVSVM), multi-class SVM (M-SVM), Multi-Criteria Decision Aid (MCDA)*

Date Of Submission: 06-05-2019 Date Of Acceptance: 21-05-2019

I. INTRODUCTION

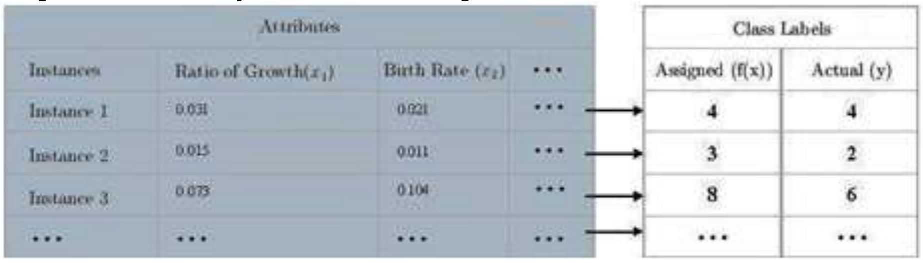
In this part we first give a concise prologue to the positioning issue and survey two related ideas: multiclass classification and relapse. At that point, we depict the inspiration of our examination and blueprint the association of the proposal.

Classification and Regression

As we have said in theory, classification is the assignment of developing an arrangement of models to isolate different classes. The utilizations of classification incorporate example acknowledgment, picture division, and characteristic dialect preparing.

In classification, an occurrence (an example) is normally spoken to by a 1 fixed number of characteristics x = [x1; x2;:::; xn], together with a class name y. An occurrence is called named if its class name is known, generally unlabeled.

Lists a few samples in the country risk classification problem



Attributes					Class Labels	
Instances	Ratio of Growth(x_1)	Birth Rate (x_2)	•••		Assigned (f(x))	Actual (y)
Instance 1	0.031	0.021	•••	→	4	4
Instance 2	0.015	0.011	•••	→	3	2
Instance 3	0.073	0.104	•••	→	8	6
•••	•••	•••	•••	→	•••	•••

Figure 1: Samples in the country risk classification problem

The process of classification has two steps: training and test use. In the training step, a set of labeled instances (the training set) is provided. Based on the training set, a mapping from the attribute space X to the class label space Y is established by training. The training process can be formulated as an optimization problem that we will discuss in the next section. The resultant mapping f is called a classifier. For example, for $X = R^d$ and Y = f0; 1g we might have a binary classifier f: X ! Y. We call it a binary classifier since all the instances are in 2 classes, either in class 0" or in class 1", i.e., the cardinality of Y is 2. The problem to construct such a binary classifier is called as the binary classification problem. Correspondingly, if the cardinality of Y is q, the problem is a q-class multi-class classification problem.[1]

After the training step, the obtained classifier f may be used to assign a 2 class label y^ to any instance x by y^ = f(x). The performance of this classifier is tested by another set of labeled instances (test set), and the prediction accuracy is assessed based on the comparison between the actual class labels and the class labels assigned by f. If the obtained accuracy is acceptable, this classifier can be used in practice. The whole classification process is demonstrated by Figure 1.

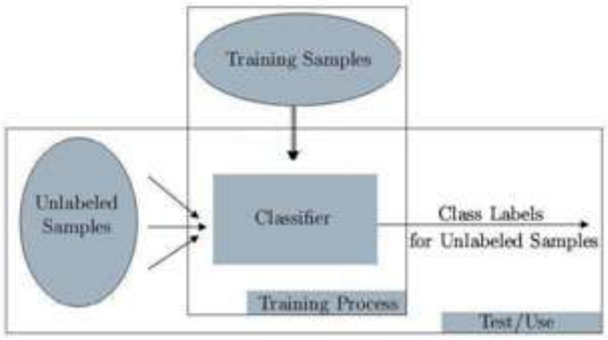


Figure 2: The classification system

In the field of data mining, the regression problem is to train a regression function that assigns a value to an instance that is as close as possible to its actual value. The applications of regression include traffic °ux estimation and air temperature prediction. Similar to the classification problem, the regression problem is to find a function that describes or distinguishes certain concept.[2] The difference between classification and regression is that, classification is to predict an in-stance's class label, which is usually discrete or nominal, while, regression is used to predict an instance's target value, which is usually continuous. In classification, the set of class labels Y is a set of discrete class labels such as f¡1; 1g or fA; B; Cg. In regression, instead, Y is a continuous domain, such as [¡1; 1] or (¡1; +1).

The Prediction Loss

The training processes of both the classification problem and the regression problem can be cast as optimization problems. Specifically, the task of the training process is to find a set of optimal parameters for the classification regression model (function) so that the expected risk of errors is minimized. We usually use a loss function, which determines the amount of loss when prediction errors take place on an instance, to evaluate the expected risk of errors of a prediction model. The most popular loss function for classification problems is the 0-1 loss function. Suppose that l(^y; y) denotes the loss function of assigning a label y^ to a sample with an actual label y.

The 0-1 loss

function is:

>
>
<
8 1; if y^ 6= y;
l(^y; y) = > (1.1)
> 0; if y^ = y:
>

misclassification> causes identical loss of value 1.

This means that each :

There are two popular loss functions for regression problems. One is 4
the least square loss function defined as:

l$_{LSL}$(^y; y) = jy^ ¡ yj^2; (1.2)

and the other one is the "-intensive loss function defined as:

>
>
<
^{8}0; if jy^ ¡ yj · ";
l"(^y; y) = > (1.3)
>
>
>jy^ ¡ yj ¡ "; if jy^ ¡ yj > ";
y
where " is a constant and ^ : y denotes the distance between the assigned

j ¡ j

class label and the actual class label. Figure 3 depicts the "-intensive loss function graphically. Both of these loss functions are distance-based ones,

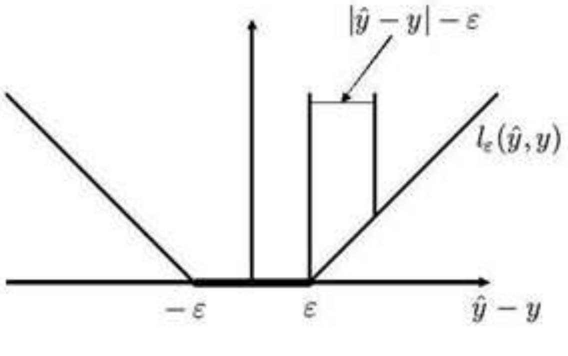


Figure 3: The "-intensive loss"

Since they are monotone functions of jy^¡yj. Compared to the 0-1 loss function for classification, the distance-based loss functions can guide the optimization process of regression training so that the obtained regression function may assign a label value to an instance that is as close as possible" to its actual label value.

The Ranking Problem

There are numerous applications where it is attractive to rank as opposed to arrange examples, for example: positioning the significance of site pages, assessing the financial FICO assessment of a man, and positioning the danger of speculations.

The positioning issue is the errand of taking in a rank-expectation display that relegates an occasion a discrete rank as close as could be allowed" to its real rank.[2]

In this postulation we define the positioning issue as a multi-class classification issue with ordinal class marks and a separation based misfortune work. We can see that positioning issues are predication issues that offer properties from both classification issues and relapse issues. In positioning issues, the qualities to be anticipated are named as positions or levels. They are discrete qualities as in classification issues. Be that as it may, the misfortune elements of positioning issues are separate based as in relapse issues. [3]

The uncommon properties of positioning issues show that the calculations for classification and relapse are not exactly reasonable for positioning issues. The proof for this contention is exhibited in the rest of our proposition.

Motivation and Organization

Two mainstream calculations for positioning issues are the Utilities Additives Discriminates (UTADIS) and the multi-class Support Vector Machine (SVM) techniques. We offer a short talk of the preferences and weaknesses of these two techniques in this segment to propose the inspiration of our exploration.

The UTADIS display, which is an effective Multi Criteria Decision Aid (MCDA) strategy, joins the foundation information in positioning issues by improving an arrangement of piecewise-direct monotone model capacities that speak to the monotone connections between's the qualities and the positions. Since for the most part the monotonicity supposition of the UTADIS display does not hold, its precision isn't empowering. The exactness of GUTADIS show is superior to UTADIS; however its preparation procedure is tedious.

SVMs have great speculation execution for pair wise classification issues. Regularly, a multi-class Support Vector Machine classifier is constructed by joining a few double Support Vector Machine classifiers. The issue with the prevalent Max-Wins" voting mix technique is that it doesn't take the ordinal connection among classes into thought and it doesn't utilize the separation based misfortune work. Subsequently this strategy isn't useful for positioning issues either.[4]

The primary reason for this proposal is to build an efficient calculation for 7 positioning issues accepting that the positioning data of classes is accessible. Specifically, we examine a few issues on the best way to consolidate paired SVM classifiers. We propose a cross breed calculation that takes the upside of both of UTADIS and SVM, and joins parallel SVM classifiers by fuzzy voting". A similar thought is reached out to the non specific multi-class classification.

Whatever is left of this proposal is sorted out as takes after. In a survey of the UTADIS and the GUTADIS models for positioning issues is exhibited. It paired SVM, multi-class SVM and SVR are audited and talked about. We propose our cross breed calculation for positioning issues and for non specific multi-class classification issues in this research paper.

II. UTADIS AND GUTADIS

In this part, we survey the UTADIS and GUTADIS strategies for positioning issues.

The UTADIS Model

Prior Knowledge in Ranking Problems

Earlier learning (foundation learning in a few references) alludes to general data about the ideas that we are worried about. Instructions to join the foundation information into the development of a classification display is a functioning exploration subject. In positioning issues, the most habitually utilized foundation information is the relationship between's the characteristics and the class names. For instance, to anticipate the age of an abalone from its physical estimations, we may utilize the earlier learning that a more noteworthy size shows a more established age.[5] Forcing the earlier learning on the classifier can influence the classification to come about predictable with specialists' information and straightforward. At the point when the preparation set is moderately little, the foundation learning can control the many-sided quality of the classification demonstrate with the goal that the over fitting issue is kept away from.

In the creators proposed UTADIS, a classification demonstrates that incorporates the foundation information in work space. In the UTADIS display, the connections between's the qualities and the class names are thought to be monotone. The supposition is steady with the foundation information of the endeavor applications.

The Model of UTADIS

The possibility of UTADIS is to prepare an utility capacity U(x) to decide the class name of test x. The idea of utility capacity originates from choice science, where the positioning issue is considered to settle on the best decision (the best class) from a rundown of decisions (classes). A presumption is that any chief unknowingly utilizes utility capacity:

U(x) = U(x_1; x_2; :::; x_n);

which aggregates his preferences in different aspects (attributes) of a choice, to rank all the optional choices. Once we get the utility function, the class label of a sample x is decided by the following rules:

8 if U(x) ¸ $¹_1$; then x 2 C_1;
>
>
>

> (2.1)
> if $¹_k$ · U(x) < $¹_{k¡1}$; k; 2 · k · q ¡ 1; then x 2 C_k;
>
>
>
<
> if U(x) < $¹_{q¡1}$; then x 2 C_q;
>
>
>
>
>
>
>
:

where the utility function U(x) is usually normalized, i.e., 0 · U(x) · 1, and $¹_1$; :::; $¹_{q¡1}$ are the decision boundaries for the q classes. Then, the main task of UTADIS training is to estimate such a utility function U(x) and a set of decision boundaries f$¹_1$; :::; $¹_{q¡1}$g. Thus, we have to define a classification function space as our search space. In the UTADIS model, U(x) is searched in the following function space:

U(x^c) = X_i U_i(x^c); (2.2a)

U_i(x^c) = U_i(x^c) = U^j_i $\frac{x_i^{j+1} ¡ x_i^c}{x_i^{j+1} ¡ x_i^j}$ + U^{j+1}_i $\frac{x_i^c ¡ x_i^j}{x_i^{j+1} ¡ x_i^j}$; (2.2b)

I

where U(x) is the additive combination of a set of criterion functions: fU_i(x); i = 1; :::; qg, and U_i(x) is a piecewise-linear monotone function that characterizes the correlation between the ith attribute and the class label. According to (2.2b), when x^j_i is fixed, the piecewise-linear function U_i(x) is decided by the sequence of U_i^j; j = 1; :::; r_i, which are the variables to be optimized. It is obvious that searching for an optimal piecewise-linear function U_i(x) is equivalent to searching for a set of optimal variables fU_i^j; j = 1; :::; r_ig. Figure 2 demonstrates a one-dimensional utility function decided by a monotone sequence of 6 points.

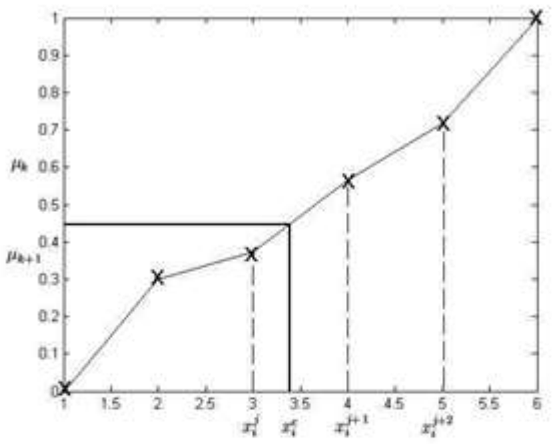


Figure 4: A piecewise monotone criterion function

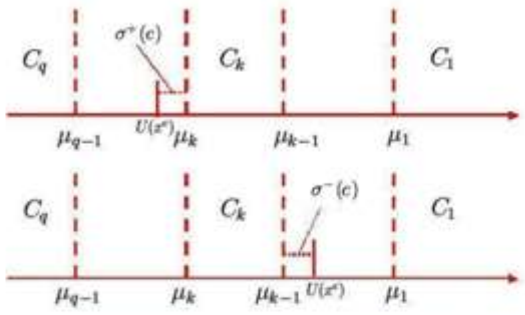
Note that by using the additive form of the utility function, we impose another implicit assumption, i.e., each criterion (attribute) affects the utility function value independently. This assumption makes the model simple and easy to solve. It is typically not true in practice, though.

The Loss Function of UTADIS

The loss function of the UTADIS model is distance-based. To describe this, we need to introduce the right range" for an instance. For an instance x in class k, its right range" is the range: [¹$_k$; ¹$_k$ ¡ 1], where ¹$_k$ is the boundary values introduced in (2.1). According to (2.1), if the utility function

value U(x) is in the right" range of x, the instance is correctly classified. Correspondingly, the classification loss of x is zero. If the value of U(x) is out of the right range" of x, the misclassification error is linearly punished.[6] Figure 4 demonstrates the right range" for instances in class k, denoted by C$_k$, and the misclassification errors when U(x) is smaller than ¹$_i$ or greater than ¹$_{i¡1}$, denoted by ¾$^+$(c) and ¾^i(c) respectively. The loss function of UTADIS is quite similar with the "-intensive loss (1.3).

Figure 5: The right range" for class k and misclassification errors



Then the UTADIS training is to find the optimal estimation of the 13 utility functions U(x) and the set of the optimal boundary values f¹$_1$; :::; ¹$_{q¡1}$g, so that the total loss of the misclassified training instances is minimized. This gives the following LP problem.

The Linear Program for UTADIS Training

The UTADIS training is a Linear Program:

min ¾$^+$(c) + ¾^i(c) (2.3a)

¾;¹;U $_c$ C :k<qc C :k>1

2 ^{X}k 2^X$_k$

n

s.t. U$_i$(x$_i^c$) ¡ ¹$_k$ + ¾$^+$(c) ¸ 0; 81 · k · q ¡ 1; 8c 2 C$_k$; (2.3b)

=1
^{X}i
n

$U_i(x_i^c)$ ¡ $^1{}_{k¡1}$ ¡ $¾^i(c)$ · ¡±; 82 · k · q; 8c 2 C_k; (2.3c)

=1

X_i

$^1{}_{k¡1}$ ¡ $^1{}_k$ ¸ s; k = 2; ¢ ¢ ¢ ; q ¡ 1; (2.3d)

U_i^{j+1} ¡ U_i^j ¸ 0; 8i = 1; ¢ ¢ ¢ ; n; 8j = 1; ¢ ¢ ¢ ; r_i ¡ 1; (2.3e)

U_i^1 = 0; 8i = 1; ¢ ¢ ¢ ; n; (2.3f)

n

$_U$ri = 1; (2.3g)
i

=1

X_i

$¾^+(c)$ ¸ 0; $¾^i(c)$ ¸ 0; 8c; (2.3h)

where n is the number of the attributes, q is the number of classes, and $U_i(x^c)$ is the criterion function defined by (2.2 b), 8c 2 C_k is used to denote for all the instances x^c whose class label are k ". Misclassification error for training instance x^c is captured by $¾^+(c)$ or $¾^i(c)$ in constraints (2.3 b,c), which is linearly punished in the objective function (2.3 a). The boundary values are defined in (2.3 d). Constraint (2.3 e) guarantees that the piecewise-linear criterion functions are monotone. Normalization constraints (2.3 f,g) define the contribution of each attribute to the resulting utility function value.

After the training step, the obtained U(x) can predict the class label of unknown samples. For an unknown instance x, we compare the utility function value U(x) with the boundary values, $^1{}_1$; :::; $^1{}_{q¡1}$, and assign x a class label according to (2.1).

The prediction accuracy of the UTADIS model is usually not good, since both the naive monotone assumption and the independent assumption are not satisfied in many applications. These unrealistic assumptions should be removed or generalized.

The GUTADIS Model

To deal with the non-monotone criteria, Wang introduced the GUTADIS model that extends the monotone assumption to unmoral. In GUTADIS model, we still use the additive combination of criterion functions to estimate the utility function, which means we still impose the independent assumption. The loss function of GUTADIS is the same as the UTADIS model. The only difference is that we search the piecewise-linear unmoral function space for optimal criterion functions.

The optimization problem of GUTADIS training is:

min $¾^+(c)$ + $¾^i(c)$ (2.4a)
¾;¹;U $_c$ C :k<q C :k>1
2 X_k c2 X_k
n
^{X}i

s.t. Ui (c) ¡ $^1{}_k$ + $¾^+(c)$ ¸ 0; 81 · k · q ¡ 1; 8c 2 C_k; (2.4b)
=1
n
X_i

U_i (c) ¡ $^1{}_{k¡1}$ ¡ $¾^i(c)$ · ¡±; 82 · k · q; 8c 2 C_k; (2.4c)
=1

$^1{}_{k¡1}$ ¡ $^1{}_k$ ¸ s; 8k = 2; ¢ ¢ ¢ ; q ¡ 1; (2.4d)

U_i^j ¡ $U_i^{j¡1}$ ¸ 0; 8i 2 I_m; 8j = 2; ¢ ¢ ¢ ; r_i; (2.4e)

$Y_{ij}(U_i^j$ ¡ $U_i^{j¡1})$ ¸ 0; 8i 2 I_u; 8j = 2; ¢ ¢ ¢ ; r_i; (2.4f)

Yij ¸ Yi;j+1; 8i 2 I_u; 8j = 2; ¢ ¢ ¢ ; r_i ¡ 1; (2.4g)

0 · U_i^j · 1; 8i = 1; ¢ ¢ ¢ ; n; 8j = 1; ¢ ¢ ¢ ; r_i; (2.4h)

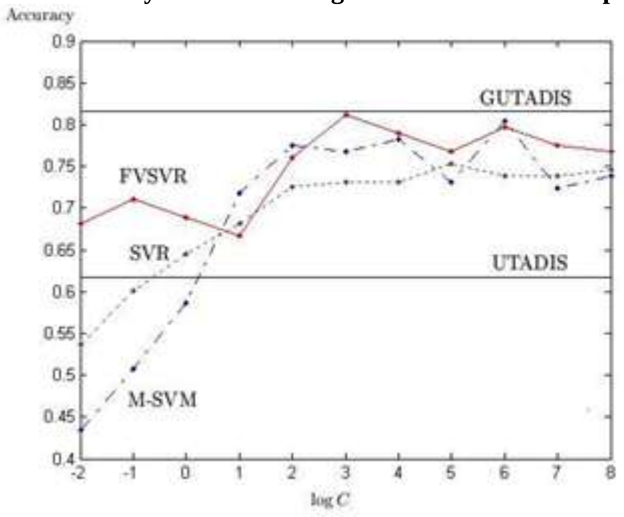
Y_{ij} 2 f¡1; 1g; 8i 2 I_u; 8j = 2; ¢ ¢ ¢ ; r_i; (2.4i)

¾$^+$(c) ¸ 0; ¾$^¡$(c) ¸ 0; 8c; (2.4j)

which is an integer nonlinear programming problem. Integer variable Y_{ij} is employed to control the increasing and decreasing of the criterion function. Constraints (2.4 f,g and i) exactly ensure that the sequence: U_i^j; j = 1; :::; r_i is a unimodal sequence.[7]

The performance of GUTADIS model is much better than the original UTADIS model, because the unimodal assumption is much less restrictive than the monotonicity assumption. Nevertheless, there is one major concern for GUTADIS, i.e, the integer nonlinear programming problem (2.4) is typically intractable which excludes the algorithm from applications with a large data set.

Figure 5: The accuracy of candidate algorithms with different penalty factor



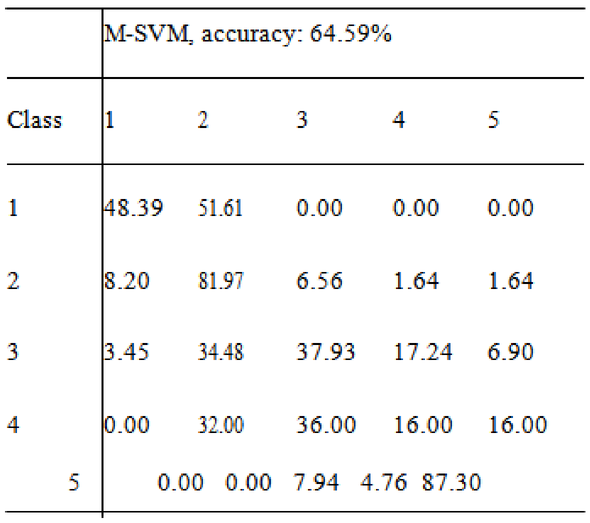
Algorithms	M-SVM	FVSVR	UTADIS	GUTADIS	SVR
Training Time	1.602	2.415	3.623	24.57	3.472

Table 1: Training time of candidate algorithms (seconds per training)

For example, the misclassification rate from class 8 to class 5 is as great as 66:67%. These serious misclassifications can be avoided by employing the distance-based loss function. We can see the effect of the distance-based loss function from the experimental results of FVSVR in Table 1. The prediction performance of FVSVR with quadratic loss function is reported by Table 1.

The performance of the UTADIS model is not as good as that of M-SVM and FVSVR, (61:59% in average).[8] Although the prediction accuracy of GUTADIS is pretty good (81:16% in average), it is quite time-consuming in computation. The prediction accuracy of "-SVR is 75:36% in the average, which is not very competitive. The accuracy of candidate algorithms is com-pared by Figure5. We can see that FVSVR achieves the best performance (81:16%).

In the first stage of the training process of FVSVR, we construct only 7 binary SV classifiers, compared by 28 binary classifiers in M-SVM. In the decision process, we solve a LP problem with 614 constraints and 332 variables compared by the LP problem in the original UTADIS model which has 2599 constraints and 2356 variables. The training times of the candidate algorithms are listed in Table1, which shows that FVSVR is as efficient as M-SVM, UTADIS and SVR. And it is much more efficient than GUTADIS.



	M-SVM, accuracy: 64.59%				
Class	1	2	3	4	5
1	48.39	51.61	0.00	0.00	0.00
2	8.20	81.97	6.56	1.64	1.64
3	3.45	34.48	37.93	17.24	6.90
4	0.00	32.00	36.00	16.00	16.00
5	0.00	0.00	7.94	4.76	87.30

Table 2: Performance of M-SVM on the Computer Hardware database

Experiments with the Computer Performance Es-Limitation Problem

This problem is to estimate the relative performance capabilities of computers in terms of their cycle time, memory size,[9] etc. We treat this problem as a five-levels ranking problem by grouping the relative performance of the items into 5 ranks: (0; 20); [20; 40); [40; 60); [60; 80); [80; M ax).

We have 209 instances in 5 ranks. Each instance has 7 numeric attributes including the class label. The performance of M-SVM, FVSVR and FVSVR with quadratic loss function for this problem is presented in Table1, Table 2 and Table 3 respectively. Since the correlations between the attributes and the ranks are monotone, the obtained accuracy of UTADIS and GUTADIS (62:20%) is as good as M-SVM (64.59 %by LIBSVM and 63.94 % by SMO in Weka) but still worse than FVSVR (67.94%). The accuracy of "-SVR is 59:41% which is not good. The accuracy is compared by Figure 8 We can see that the FVSVR achieves the best performance (67:94%).

	FVSVR, accuracy: 67.94%				
Class	1	2	3	4	5
1	58.06	41.94	0.00	0.00	0.00
2	11.48	70.49	11.48	6.56	0.00
3	0.00	20.69	51.72	20.69	6.90
4	0.00	16.00	32.00	44.00	8.00
5	0.00	0.00	4.76	7.94	87.30

Table 3: Performance of FVSVR on the Computer Hardware database

	FVSVR, accuracy: 67.94%				
Class	1	2	3	4	5
1	58.06	38.71	3.23	0.00	0.00
2	8.20	70.49	14.75	6.56	0.00
3	0.00	17.24	48.28	27.59	6.90
4	0.00	16.00	32.00	48.00	4.00
5	0.00	0.00	6.35	6.35	87.30

Table 4: Performance of FVSVR with quadratic loss on the Computer Hard-ware database

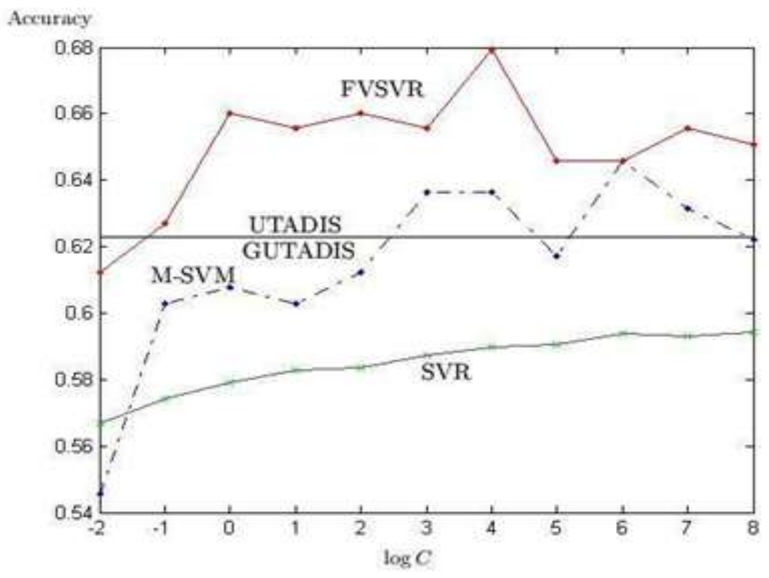


Figure 5: The accuracy of candidate algorithms with different penalty factor

Algorithms	M-SVM	FVSVR	UTADIS	GUTADIS	SVR
Training Time	0.1156	0.8412	1.7525	6.6850	0.2983

Table 5: Training time of candidate algorithms (seconds per training)

In the first stage of the training process of FVSVR, we construct only 4 binary SV classifiers, compared by 10 binary classifiers in M-SVM. In the decision process, we solve a LP problem with 871 constraints and 450 variables compared by the LP problem in the original UTADIS model which has 1146 constraints and 730 variables.[10] The training times of the candidate algorithms are listed in Table 5, which shows that FVSVR is a little time-consuming than SVM and SVR, but is more efficient than UTADIS and GUTADIS.

	Predicted by M-SVM, accuracy: 64.32%					
Class	1	2	3	4	5	6
1	83.02	16.98	0	0	0	0
2	13.27	70.41	14.29	2.04	0	0
3	1.28	17.95	51.28	25.64	2.56	1.28
4	0	2.60	15.58	70.13	9.09	2.60
5	0	0	1.79	16.07	55.36	26.79
6	0 0 2.78 5.56 41.67 50.00 Table 5.10:					

Table 6: Experiments with the Auto-mpg Estimation Problem

The Auto-mpg database concerns city-cycle fuel consumption in Miles Per Gallon, to be predicted in terms of such attributes as horsepower and weight of a vehicle.[11]

Table 7: Performance of FVSVR on the Auto-mpg database

	Predicted by FVSVR, accuracy: 66.08%					
Class	1	2	3	4	5	6
1	81.13	18.87	0	0	0	0
2	11.22	71.43	17.35	0	0	0
3	1.28	12.82	61.54	21.79	2.56	0
4	0	0	22.08	64.94	11.69	1.30
5	0	0	1.79	19.64	57.14	21.43
6	0	0	2.78	0	41.67	55.56

We treat this problem as a six-levels ranking problem by grouping the MPGs into 6 levels: (0; 15); [15; 20); [20; 25); [25; 30); [30; 35); [35; M ax).[12] We have 398 instances in 6 ranks. Each instance has 8 attributes including the class label. The performance of M-SVM, FVSVR and FVSVR with quadratic loss are presented in Tables 4, 5, and 6 respectively. Since the background knowledge about the correlations between the attributes and the ranks is unavailable in this problem, to test the performance of the UTADIS model, we have to arbitrarily assume that each attribute has an increasing correlation with the rank of MPGs that is a greater attribute value indicates a higher rank. Thus, the obtained accuracy of UTADIS is not good (52:76%).

	Predicted by FVSVR, accuracy: 67.84					
Class	1	2	3	4	5	6
1	83.02	16.98	0.00	0.00	0.00	0.00
2	13.27	71.43	15.31	0.00	0.00	0.00
3	0.00	15.38	61.54	20.51	2.56	0.00
4	0.00	0.00	22.08	62.34	14.29	1.30
5	0.00	0.00	1.79	14.29	66.07	17.86
6	0.00	0.00	2.78	5.56	27.78	63.89

Table 8: Performance of FVSVR with quadratic loss on the Auto-mpg database

The accuracy of "-SVR on this database is 64:82% that is a little bit better than M-SVM (64:32% by LIBSVM and 63:47% by SMO in Weka), but still worst than FVSVR (67:84%). The GUTADIS method achieves an accuracy of 63:47%. The accuracy is compared by Figure 6 In the first stage of the training process of FVSVR, we construct only 775 binary SV classifiers, compared by 15 binary classifiers in M-SVM. In the decision process, we solve a LP problem with 1636 constraints and 836 variables compared by the LP problem in the original UTADIS model which has 1954 constraints and 1160 variables. The training times of the candidate algorithms are listed in Table 7, which shows that FVSVR is a little time-consuming than SVM and SVR, but is more efficient than UTADIS and GUTADIS.[13]

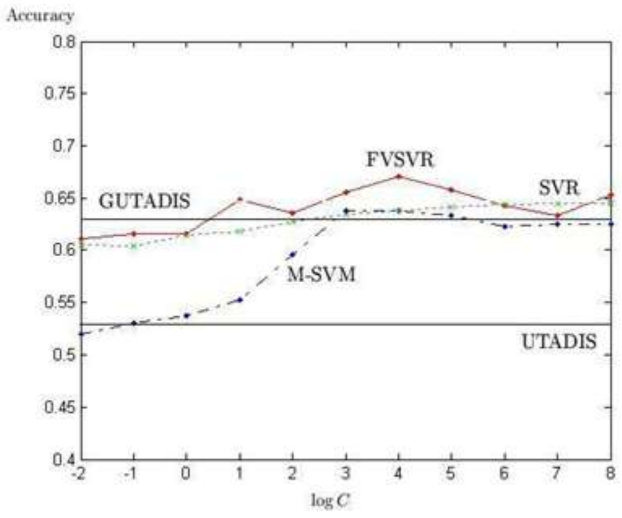


Figure 6: The accuracy of candidate algorithms with different aplenty factor c

Algorithms	M-SVM	FVSVR	UTADIS	GUTADIS	SVR
Training Time	0.3305	1.8426	3.5823	34.3292	1.6123

Table 9: Training time of candidate algorithms (seconds per training)

Generic Multi-class Classification Problems

The trial is to think about the forecast execution of M-SVM and FVSVM for bland multi-class classification issues. The M-SVM and FVSVM are contrasted and a few functional issues which has been utilized as a part of to test the execution of SVM-based strategies. They are the glass identification issue, the DNA identification issue, the vowel acknowledgment issue, the vehicle acknowledgment issue, the iris identification issue, the wine identification issue, the fragment acknowledgment issue, and the sat image identification issue. They are non specific multi-class classification issues, that is, the positioning data of classes are unavail-capable or vague. Every one of them are from the UCI Machine Learning Repository.[14]

Experiments with the Glass Identification Problem

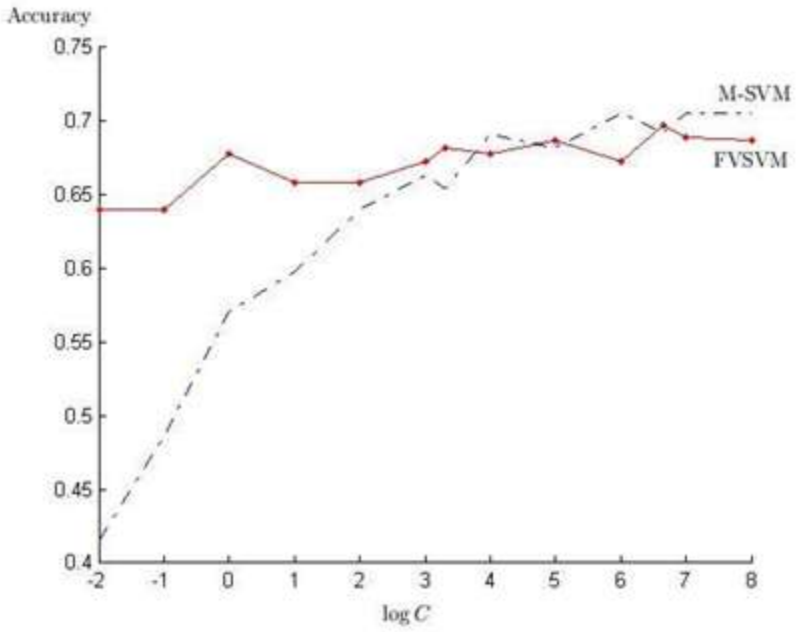


Figure 7: The comparison of M-SVM and FVSVM with glass

The investigation of grouping of sorts of glass was spurred by criminal-intelligent examinations. At the scene of the wrongdoing, the glass left can be utilized as proof, on the off chance that it is accurately distinguished. So the errand of glass distinguishing proof is to recognize the sort of glass by its oxide content (i.e., Na, Fe, K, and so forth). In this database, we have 214 occasions in 6 classes. Each occurrence has 10 characteristics including the class mark. The exhibitions of M-SVM and FVSVM are looked at and evil spirit started in Figure 4, which demonstrates the expectation precision of these two algorithms with different estimations of the punishment factor C. We can see when C is little (under 22) the forecast exactness of M-SVM (demonstrated by the dash-speck line) is unpleasant. In the interim the forecast precision of FVSVM (indicated by the strong line) is constantly worthy. At the point when C achieves it ideal esteem. M-SVM and FVSVM accomplishes comparative precision.

We legitimize this outcome by the accompanying exchange. Remind that the \fuzzy voting" strategy enhances the execution of the first M-SVM technique by removing more data from the twofold classifiers. At the point when C is far from its ideal esteem, the double detachment limits may not be ideal, at that point the fuzzy voting" method can utilize the extra information from the choice capacity to modify the mix of the parallel partitions so that got multi-class classifier accomplishes a superior performance.[15] Be that as it may, when C is near its ideal esteem, the paired divisions are almost ideal, the double partition limits themselves are represent sufficiently dative to build the multi-class detachment. At that point, the data from the choice capacities is typically repetitive, and subsequently FVSVM can barely complete a superior employment then M-SVM in this circumstance.

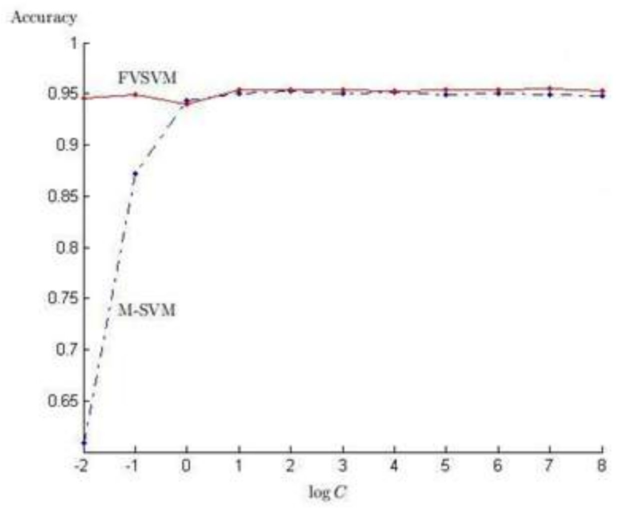
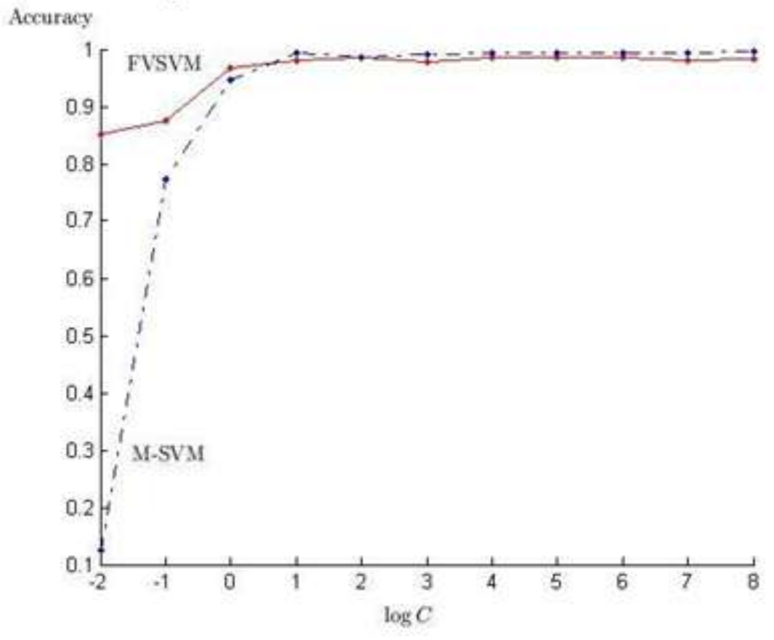


Figure 8: The comparison of M-SVM and FVSVM with DNA

Experiments with the DNA Identification Problem

In this problem, we have 2000 training instances and 1186 testing instances from 3 classes. Each of them has 181 binary attributes including the class label. The comparison of M-SVM and FVSVM is demonstrated by Figure 8, from which, we can see that FVSVM is much more robust.

Figure 9: The comparison of M-SVM and FVSVM with vowel



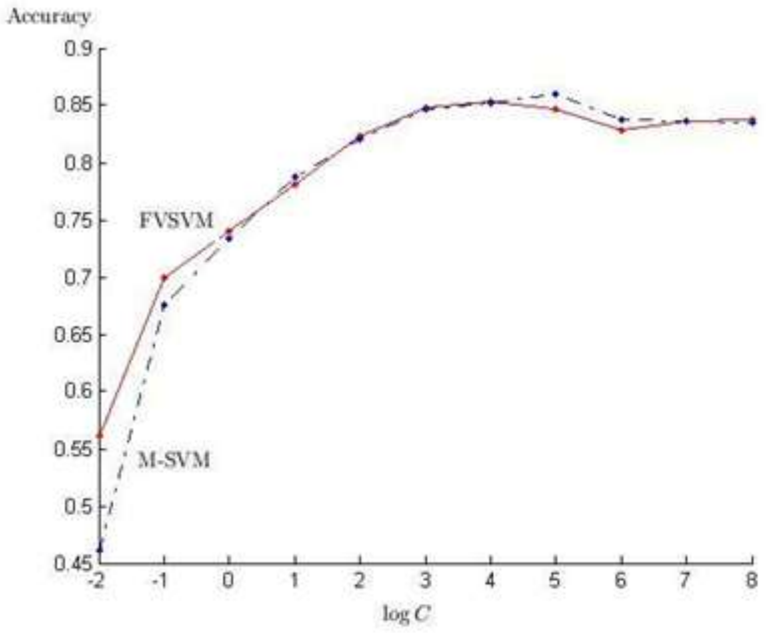
Experiments with the Vowel Recognition Problem

In this problem, we have 528 instances from 11 classes. Each of them has 11 attributes including the class label. The comparison of M-SVM and FVSVM is demonstrated by Figure 9, from which, we can see that FVSVM is more robust.

Experiments with the Vehicle Recognition Problem

In this problem, we have 846 instances from 4 classes. Each of them has 19 attributes including the class label. The comparison of
M-SVM and FVSVM is demonstrated by Figure 10, from which, we can see that FVSVM is a little more robust.

Figure 10: The comparison of M-SVM and FVSVM with vehicle



Experiments with the Iris Identification Problem
In this problem, we have 150 instances from 3 classes. Each of them has 5 attributes including the class label. The comparison of
M-SVM and FVSVM 84 is demonstrated by Figure10, from which, we can see that FVSVM is much more robust.

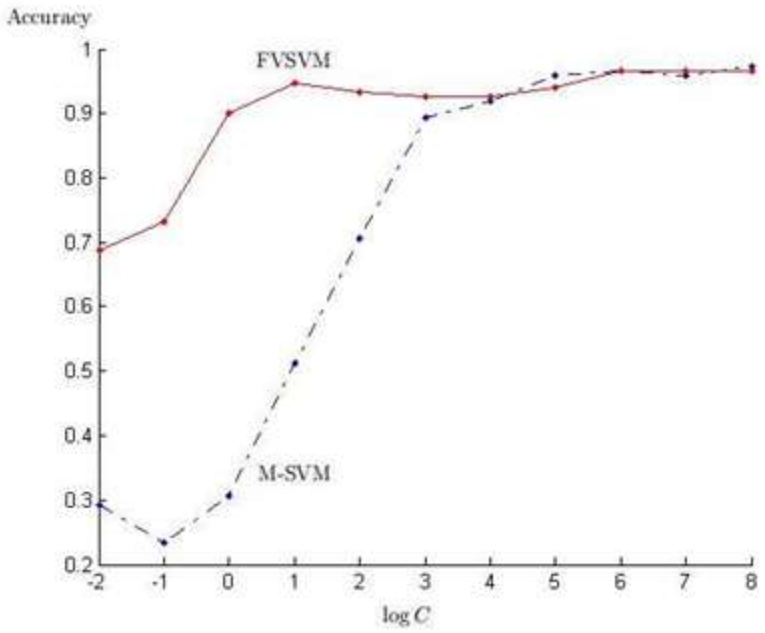
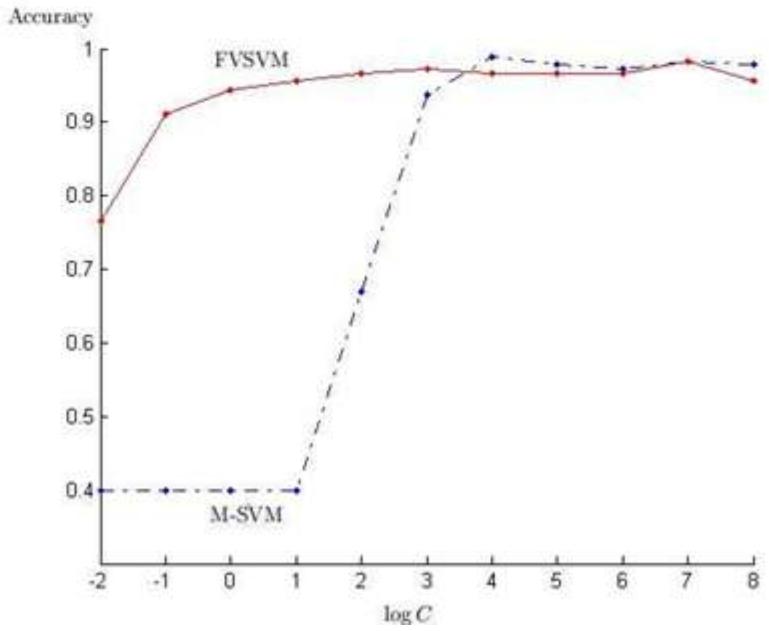


Figure 11: The comparison of M-SVM and FVSVM with iris

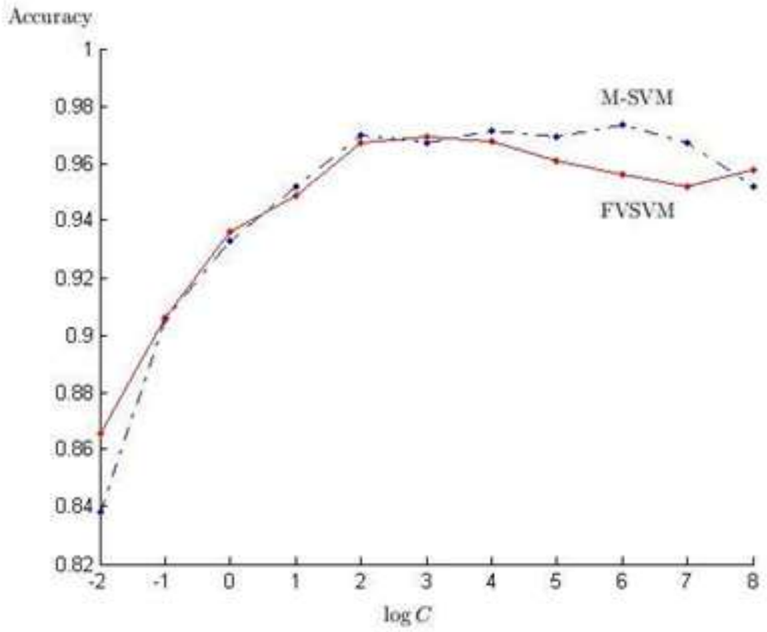
Experiments with the Wine Identification Problem
In this problem, we have 178 instances from 3 classes. Each of them has 14 attributes
including the class label. The comparison of M-SVM and FVSVM is demonstrated by Figure 5.9, from which, we can see that FVSVM is much more robust.

Figure 12: The comparison of M-SVM and FVSVM with wine



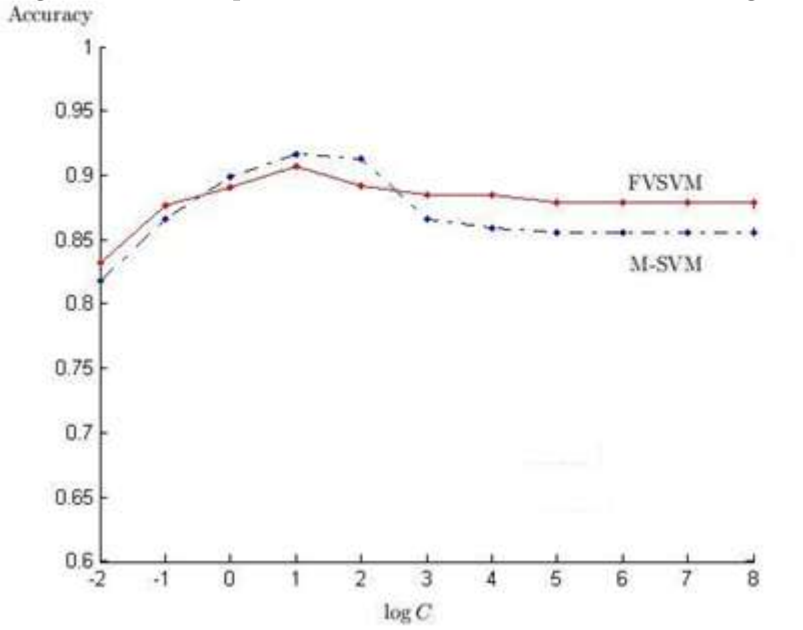
Experiments with the Segment Recognition Problem
In this problem, we have 2310 instances from 7 classes. Each of them has 20 attributes including the class label. The comparison of M-SVM and FVSVM is demonstrated by Figure 13, from which, we can see that FVSVM and M-SVM have similar performance.

Figure 13: The comparison of M-SVM and FVSVM with segment



Experiments with the Sat image Recognition Problem
In this problem, we have 4435 training instances and 2000 testing instances from 6
classes. Each of them has 37 binary attributes including the class label. The comparison of M-SVM and FVSVM is demonstrated by Figure14, from which, we can see that FVSVM is more robust.

Figure 14: The comparison of M-SVM and FVSVM with sat image



Time Complexity

It is obviously that FVSVM is more time-consuming than SVM since they use the same set of binary SV classifiers and an additional LP problem has to be solved in the decision process of FVSVM. In practice, the running time of FVSVM training for each problem is roughly twice of that of M-SVM. So our FVSVM is a promising trade-off between the robust city and the time-complexity of the M-SVM method.

III. CONCLUSIONS AND FUTURE WORK

In this proposition, we contrast UTADIS based techniques and Support Vector Ma-chine based strategies for positioning issues. We presume that in spite of the fact that the multi-class SVM (M-SVM) has great execution for non specific multi-class classification issues, it neglects to control the genuine misclassifications for positioning issues. This is mostly because of the way that M-SVM isn't driven by separate based misfortune, henceforth it can't particular the genuine misclassifications from the others. We propose a half breed calculation that consolidates M-SVM and UTADIS to take care of this issue. In this new calculation, paired SVM classifiers are joined into a multi-class classifier by the fluffy voting strategy rather than the correct voting method in M-SVM. Thusly, the new calculation is named as Fuzzy Voting based Support Vector Ranking (FVSVR) technique. Exact outcomes on the databases of three run of the mill positioning issues demonstrate that the FVSVR technique accomplishes preferable execution by and by over M-SVM, UTADIS and SVR. We likewise broaden the possibility of Fuzzy Voting from positioning issues to non specific multi-class classification issues, which brings about the supposed Fuzzy Voting based Support Vector Machine (FVSVM) strategy. Our exact outcomes demonstrate that FVSVM is uncaring to the decision of the punishment factor C.

There are as yet a couple of issues we have to investigate for both the FVSVR and FVSVM strategies. To begin with, as we have specified previously, there are three techniques by which twofold SVM classifiers are joined to multi-class classifiers: one-against-one", one-against-all", and DAG". The FVSVM technique we proposed is only the fluffy voting variant of the one-against-one" procedure. A similar thought can be reached out to the \one-against-all" and the DAGSVM" systems.

Furthermore, in FVSVR and FVSVM the enhancement issues of hunting the parallel SVM classifiers and that of looking through the ideal voting are two separate advances. It is a fascinating issue to investigate how these two stages can be joined together into a solitary streamlining issue.

REFERENCES

[1]. S. Alexe, P. L. Hammer, A. Kogan and M. A. Lejeune, A Nonrecursive Regression Model for Country Risk Rating, Technical Report, Rutgers University, Center for Operations Research, Piscataway, New Jersey, 2003.

[2]. J. Bi and K. Bennett, A Geometric Approach to Support Vector Regression, Neurocomputing, 55, pp. 79-108, 2003.

[3]. L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. MÄuller, E. SÄackinger, P. Simard, and V. Vapnik, Compari-son of Classifier Methods: A Case Study in Handwritten Digit Recognition ICPR, pp. 77-87, 1994.

[4]. C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 2, pp. 121-167, 1998.

[5]. C. Chang and C. Lin, User Manual for LIBSVM, 2004. http://www.csie.ntu.edu.tw/~cjlin/libsvm/ / / Accessed on Dec 2005

[6]. W. Cohen, R. E. Schapire and Y. Singer, Learning to Order Things, Advances in Neural Information Processing Systems 10, Morgan Kaufmann, 1998.

[7]. C. Cortes, V. Vapnik, Support-Vector Networks, Machine Learning, 20, 273-297, 1995

[8]. K. Crammer and Y. Singer, Pranking with Ranking, In Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2001.Link: http://citeseer.ist.psu.edu/crammer01pranking.html

[9]. J. Czyzyk, M. Mesnier, and J. Morµe, The neos server, IEEE Journal on Computational Science and Engineering 5, pp. 68-75, 1998. http://www-neos.mcs.anl.gov/

[10]. G. E. Dallal, Introduction to Simple Linear Regression, http://www.tufts.edu/~gdallal/slr.htm

[11]. M. Doumpos and C. Zopounidis, Multicriteria Decision Aid Classification Methods, Vol. 73 of Applied Optimization, Kluwer Academic Publishers, Norwell, MA, U.S.A. 2002.

[12]. J. Han and M. Kamber, Data Mining, Concepts and Techniques, The 92 Morgan Kaufmann Series in Data Management Systems, pp. 279-281, 2001.

[13]. C. Hsu and C. Lin, A Comparison of Methods for Multi-class Support Vector Machines, IEEE Transactions on Neural Networks, 13, pp. 415-425, 2002.

[14]. R. L. Keeney and H. Rai®a, Decisions with Multiple Objectives, Preferences and Value Tradeo®s, Cambridge University Press, Cambridge, UK, 1993. http://www.boosting.org/papers/CohSchSin98.pdf

[15]. R. Kirkby and E. Frank, Weka Explorer User Guide for Version 3.4, 2005. http://www.cs.waikato.ac.nz/ml/weka/

Prof.Er.Dr.G.Manoj Someswar" Design and Development of an Efficient Data Mining Algorithm for Ranking Problems" Research Inventy: International Journal of Engineering And Science, vol. 09, no. 01, 2019, pp. 01-18