SCIT

ASSIGNMENT COVERSHEET

STUDENT SURNAME

STUDENT DETAILS			
STUDENT NAME			
STUDENT NUMBER			
UOW EMAIL			
ASSIGNMENT DETAILS			
SUBJECT NAME			
SUBJECT NUMBER			
ASSIGNMENT NAME			
TUTORIAL / LAB TIME			
LECTURER / TUTOR			
SUBMISSION DETAILS			
DATE DUE			
STUDENT SIGNATURE			
STAFF ONLY			
RECEIVED BY		DATE/TIME LODGED	
MARKER		GRADE	

By signing this, I declare that:

1. This assignment meets all the requirements of the subject as detailed in the relevant subject outline, which I have read.
2. (a) This assessment item is entirely my work, except where I have included fully-documented references to the work of others.
   (b) The material contained in this assessment item has not previously been submitted for assessment.
3. I acknowledge that:
   (a) The marker of this assessment item may, for the purpose of assessing this assignment, reproduce this assignment and provide a copy to another member of academic staff.
   (b) If required to do so, I will provide an electronic copy of this assessment item to the marker.
4. Where Turnitin is used by the faculty: I acknowledge that the marker of this assessment item may communicate a copy of this assignment to a text-matching service such as Turnitin (which may retain a copy of this assignment on its database for the purpose of future text-matching).
5. I am aware that late submission without an authorised extension from the subject co-ordinator may incur a penalty. (See your subject outline for further information).

Please note: Assignments are not to be submitted by fax and must be submitted during lectures/tutorials/laboratories or directly to the academic. Only under special circumstances will the Administrative Staff collect assignments.

STUDENT REFERENCE			
STUDENT NAME		TUTOR	
STUDENT NUMBER		RECEIVED BY (Academic Signature)	
ASSIGNMENT NAME			
SUBJECT NUMBER		DATE / TIME	

FACULTY OF ENGINEERING & INFORMATION SCIENCES | UNIVERSITY OF WOLLONGONG

Abstract

The report is composed of five parts. Firstly, we briefly introduce the basic methods, application and paradigms of five mainstream machine learning algorithms. Secondly, we list how industries use machine learning technology to provide more customized services. Then our main focus is located on deeply understanding the main principles based on mathematical perspective. Then by doing some practical experiments on some sample data, we analyze the advantages and disadvantages of the researched ones. Finally, we propose our future work based on the research findings.

Contents

Introduction:

In recent years, digital transaction is becoming the standard method of purchasing goods and service. This trend is eventually leading e-commerce segment to explore how AI can enhance brand competitiveness and customer loyalty. These technologies also reduce the overall human error as well as human efforts. By using AI technology, systems or models can connect the dots to more intuitively help customers and make services more time-saving and customized. The most popular approach to build customers' portfolio is to learn their preferences from online browsing habits. Some systems, such as Movielens, Fab, Entree restaurant recommender and so on, derive various input data from customers, including four types of data, demographic data, rating data, behaviour pattern data and transaction data[1]. Besides the common input data discussed above, some researchers proposed more complex input data indicators such as move movement or time spent reading each section of page.

User recommendation system usually works in three steps, which are obtaining preference from input data of users, filtering and processing the recommendation using proper techniques, and finally presenting tailored recommendations to them.

The typical method of filtering user recommendation, according to most popular classified standard, which has been widely accepted by most researchers, is sorted into collaborative filtering, contend-based filtering and hybrid approach[1].

One of the earliest and successful recommendations technique is collaborative filtering. It uses user dataset from the stored information of other previous users and their preferences and compare between their usage and active user. By applying collaborative algorithms, customized recommendations are given to active users.

Content-based filtering method was initially used in information retrieval and information filtering. The item recommended by content-based filtering often indicates textual information, or content-based information. And these items usually describe with keywords and weights. To avoid the disadvantages of existing approaches between collaborative filtering and content-based filtering such as the cold starting problems, sparsity problems, gray sheep problems and scalability problems, some researchers have combined these two methods and introduced the hybrid filtering approach.

There are two filtering methods for user recommendation, Heuristic-based and Model-based approaches. The main difference between them is that the heuristic-based approach uses the formulations to compute results on entire database for each user, while the model-based approach just inputs some data from active user into the model, and the selected model can give the prediction value and give recommendations to active users[1].

In the current proposal we are using model-based methods for recommendations to users. The research includes SVM, Bayes decision rule, decision tree, gradient descent algorithm and

neural network.

Practical applications:

There are many technologies that are being used. Among all the technologies we are focusing on a few of them given below.

SVM (support vector machine)

Nowadays web pages are more than just plain text documents, web classification methods must consider using other context features of web pages, such as hyperlinks and HTML tags. There are multiple approaches can be used to sort this problem. One of the method is using SVM to classify the web pages by text and content[3]. Another method is using weighted vote schema for various features and then training the SVM model[2]. SVM model is trained using latent semantic analysis and web age feature selection. Latent semantic analysis is used to find the semantic relations between keywords, and between documents.

Bayes decision rule and decision tree

One of the investment strategies is to integrate supply chain with finance to control the market timing. Current study uses different indexes in fundamental and technical analysis, rough set theory and artificial neural networks inference system[6]. As some part of it, Bayes decision rule can also be used to determine the probabilistic of how beneficial the current investment cycle can be, and can it be changed.

Gradient descent algorithm

These days users are posting newly bought products onto micro-blogs with an e-commerce link to product web pages, boosting the overall sales of products. Approaches like cross-site cold-start product recommendation use gradient descent algorithm. Cold-start product recommendations aim to recommend products from e-commerce websites to users at social

networking sites in “cold-start” situations[5]. Cold-start product recommendation also links users across social networking sites and e-commerce websites, bringing convenience to users. Cold-start product recommendation experiments being carried out on Chinese largest micro-blogging service SINA WEIBO and the largest Chinese B2C e-commerce website JINGDONG have significant improvement in the user access[5].

Neural network:

Neural network is a bio-simulation model composed of multi-layer neural network network, computational neural nodes. Artificial neural network is widely applied in different fields, such as in users’ behaviors predictions, image recognization, regression analysis and so on.

Algorithms research:

Gradient descent:

In linear regression model, the main process is to find the most appropriate weight vector θ which can generate the highest accuracy in testing phase. The model of linear regression model is as follow,

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + .. + \theta_n x_n, \qquad (1)$$

where y is the outcome in training sets and $x_{1..n}$ are all attributes datasets have. If transforming the model formula by using the array multiplication, we can get

$$\widehat{y} = h_0(x) = \theta^T.x \qquad (2)$$

in vectorized form. According to the addition and multiplication rule of linear algebra, θ is equal to the following formula.

$$\widehat{\theta} = (x^T.x)^{-1}.x^T.y \qquad (3)$$

Then, the mean value of all the test cases can be achieved by using the L2 norm and summing up the errors. So, the Mean Square Error formula are shown as follow

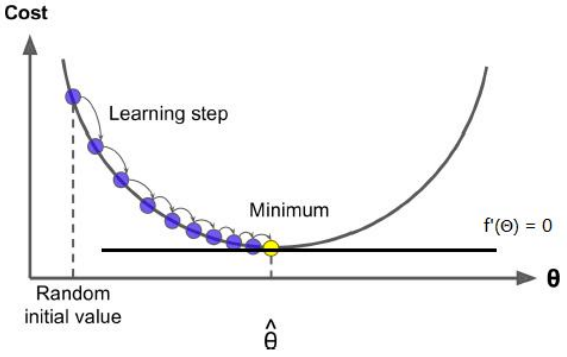
$$MSE(x,h_0) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T.x^{(i)} - y^{(i)}\right)^2, \qquad (4)$$

which is also known as cost function.

However, calculating the inverse of a array is computationally expensive, especially to those

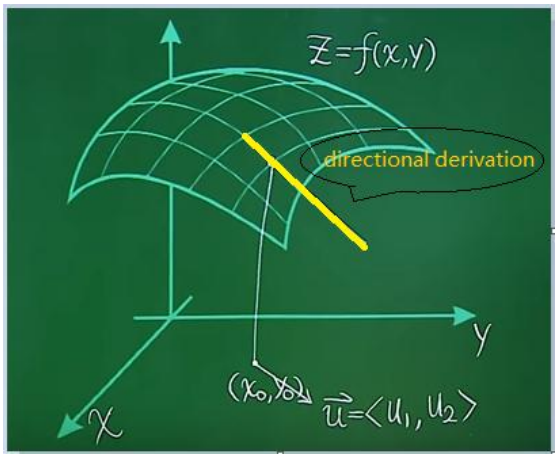
data sets with thousands of samples and many attributes in each of them. There are two solutions to find the best θ, stochastic gradient descent algorithm (SGD), mini batch gradient descent algorithm. The main idea of SGD is to reach the global minimum point of the cost function eventually by initially randomizing the weight vector θ and optimizing them gradually while mini patch can update data weights by using mean after going through a mini batch of sample data[9].

For example, if there is a two-dimensional bowl-shaped curve, the minimum cost lies in the intersection between the horizontal tangent line f'(θ) and the curve. The derivative of function f with respect to variable θ reflects the slope of all tangent lines. When it is equal to zero, we can get the value of θ and the corresponding cost.



Nevertheless, in the muti-dimensional space, getting the derivation of a function is computationally hard. But mathematicians provide a convenient way to reach the bottom point of the curve by using gradient and directional derivation.

In a curved surface, the slope of the tangent line, which starts from a specific point and follows the direction of an unit vector, is defined as the directional derivation of that point towards the unit vector. The geometrical meaning is shown as follow.



The function of the curved surface is z=f (x, y), where x and y are variants and z is the corresponding value of two inputs x and y. There is an unit vector, named u = $<u_1, u_2>$ and its norm is one. The directional derivation is the slope of the yellow line here which is the tangent line of the point $(x_0, y_0, f(x_0, y_0))$. Mathematically speaking, we define directional

derivation in the form of limitation as follow.

$$D_u f(x_0,y_0) = \lim_{h\to 0} \frac{f(x_0+hu_1,y_0+hu_2)-f(x_0,y_0)}{h} \qquad (5)$$

Also, we can represent directional derivation in the form of inner product

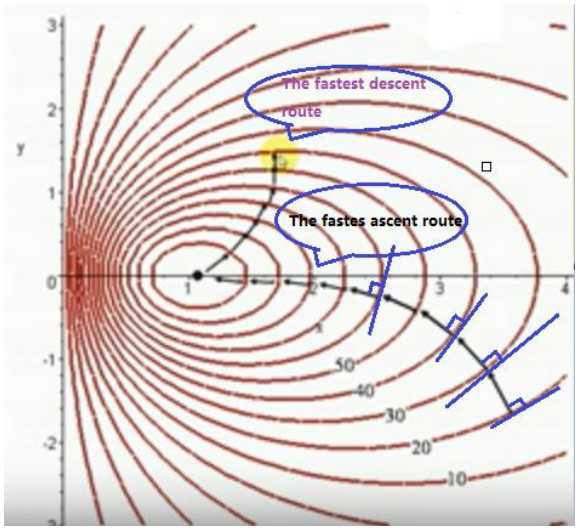
$$D_u(x,y) =< f_x,f_y >*< u_1,u_2 >= |\nabla f|cos\theta` \quad (6)$$

where f_x and f_y are the partial derivation f to x and y respectively and the vector $<f_x, f_y>$ is the gradient on the point (x_0, y_0) and $|\nabla f|$ is the length of the gradient vector. cos θ` is the cosine of the angle between vector $<f_x, f_y>$ and $<u_1, u_2>$.

Theoretically speaking, when the direction of the gradient vector and unit vector have the same direction, the directional derivative will be maximized, since cos(0)=1. However, if the angle between them is π, it indicates the directional derivation pointing towards to the global minimum point. Practically speaking, $<f_x, f_y>$ is in the direction of the fastest increase or decrease of the function on a specific point.

Theorem: the optimal direction of gradient on a given point is always orthogonal with the tangent line of the point on the isogram of the function.

For example, the isogram of the function z= f (x, y) is shown as follow, where all points on every circular line have the same height or z value.



So, the solution is that by calculating the partial derivation and setting the “suitable” learning rate, we can move down from the start point to the target point gradually until we hit it. The partial derivatives of the cost function MSE(θ) to θ_j is shown as follow,

$$\frac{\partial}{\partial\theta_j}MSE(\theta) = \frac{2}{m}\sum_{i=1}^{m}\left(\theta^T.x^{(i)} - y^{(i)}\right)x_j{}^{(i)}. \qquad (7)$$

where m is the sample index. Since we have to carry out the partial derivatives to the rest θ_p where p is from 0 to n and to every sample in the data set repeatedly, the gradient of a sample on its hyper plane is shown as follow,

$$\theta MSE(\theta) = \begin{pmatrix} \frac{\partial}{\partial\theta_0}MSE(\theta) \\ \frac{\partial}{\partial\theta_1}MSE(\theta)\ldots \\ \frac{\partial}{\partial\theta_2}MSE(\theta) \end{pmatrix} = \frac{2}{m}x^T.(x.\theta - y). \quad (8)$$

After getting the gradient vector, we will decide the size and direction of next step according to the following formula, where η is learning rate and keep taking a step off until the algorithm totally converge, which means that we reach the minimum of the cost function.

$$\theta^{(next\ step)} = \theta - \eta\ \ \theta MSE(\theta). \qquad (9)$$

Bayes decision rule:

In contrast with finding the best solution or address an optimization problem in a geometric and algebraic way by using derivative in gradient descent algorithm, the main concept of Bayes decision rule is that when having the full knowledge of the distribution of all classes in dataset, we can assign a test object into a chosen class by the highest correct probability. In probability theory, conditional probability is a measure of the probability of an event given that another event has occurred (***from Wikipedia***). The probability of A event under B event is written as P(A|B). The Bayes decision rule is shown as follow[7][8],

$x \in w_j$, if, $P(w_j|x) > P(w_k|x)$ k = 1, 2...., C; k≠ j, where there are C classes denoted as w_k.

There are three important probabilities in Bayes decision rule.

- **$p(w_i|x)$** named posteriori probability

- **$p(w_i)$** named priori probability

- **$p(x|w_i)$** named class-conditional density functions.

Because $p(w_i|x)p(x)=p(w_i \cap x)$ and $p(x|w_i)p(w_i)=p(x \cap w_i)$, we get

$$P(w_i|x) = \frac{p(x|w_i)p(w_i)}{p(x)} \qquad (10)$$

where

$$p(x) = \sum_{j=1}^{c} p(x|w_j)P(w_i) \qquad (11)$$

Then we can transform decision rule into class-conditional density function. We can assign sample x to class w_j, if $p(x|w_j)p(w_j) > p(x|w_k)p(w_k)$, where w_k denotes any other class, which is named Bayes' rule for minimum error.

However, sometimes, minimum error rule cannot reflect the real cost if we make the wrong decision such as assigning x to wrong class w_a, since various decisions have different costs. To solve corresponding cost brought by different decisions, we introduce a loss matrix λ_{ji} = cost of assigning a pattern x to w_i when $x \in w_j$. So, if we have no idea of the correctness of one decision, conditional risk of assigning a pattern x to a class is defined as

$$I_i(x) = \sum_{j=1}^{c} \lambda_{ji} P(w_j|x) \qquad (12)$$

Then we have the Bayes decision rule for minimum risk r*

,

$$r^* = \int_x \min_{i=1,\ldots,c} \sum_{j=1}^{c} \lambda_{ji} P(w_j|x)p(x)\,dx \qquad (13)$$

which indicates that sum up all the losses of assigning x to all different classes from c_1 to c_C (if i=j the loss will be 0; if not, it is 1) and find the minimum loss as our final decision of assigning x to class c_i.

For example, if there are only two classes (binary classification), we can simplify the risk or loss equation into the following formulas.

$$I_1(x) = \lambda_{11}P(w_1|x) + \lambda_{21}P(w_2|x) \qquad (14)$$

$$I_2(x) = \lambda_{12}P(w_1|x) + \lambda_{22}P(w_2|x) \qquad (15)$$

The minimum risk decision rule is to simply decide w_1 if $I_1(x) < I_2(x)$. This can be expressed in terms of posterior probabilities as: Decide w_1 if,

$$(\lambda_{11} - \lambda_{12})p(w_1|x) < (\lambda_{22} - \lambda_{21})p(w_2|x) \qquad (16)$$

In terms of the prior probabilities and conditional densities we decide w_1 if,

$$(\lambda_{11} - \lambda_{12})p(x|w_1)P(w_1) < (\lambda_{22} - \lambda_{21})p(x|w_2)P(w_2) \qquad (17)$$

Decision Tree:

Compared with first two classifiers, decision tree is comparatively easier to comprehend. The main principle is to find those attributes which are highly related with target values and set up the decision tree to predict outcomes. There are several ways to figure out how relevant each attribute is towards its category[11][12][13].

Conditional entropy H(Y|X): With the occurrence of condition A, the amount of uncertainty provided by condition B.

We assume that all data is discrete which means the occurrence of all events is independent, so the discrete conditional entropy formulation will be employed. Pr(Y|X) is the conditional probability.

$$H(Y|X=\nu) = \sum_{i=1}^{k} - \Pr(Y=y_i|X=\nu)\log_2 \Pr(Y=y_i|X=\nu) \quad (18)$$

$$H(Y|X) = \sum_{\text{vxaluesof } X} \Pr(X=\nu)H(Y|x=\nu) \quad (19)$$

Information gain manifests the certainty of X for the occurrence of the condition Y.

$$I(Y|X) = H(Y)-H(Y|X). \quad (20)$$

So, based on the information gain, we can figure out how much contribution each factor or condition has towards specific output we expect.

Similarly, we can apply gini index for the same purpose. The formula is shown as follow.

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2 , \quad (21)$$

where n is the number of unique values in a selected attribute and p_j denotes the probability of the i_{th} value in a sample space. Then, we place the attribute which has the highest gini index at the highest layer. Secondly, we need to find a split point to divide the whole value set of that attribute in a way that provides the highest purity. If it is a numerical feature, there will be two groups, below and above the split point. If it is categorical, the two groups will be “equal” and “not equal”. We determine the split point using the following formula:

$$Gini_{split}(T) = \sum \frac{N_i}{N} gini(T_i) , \quad (22)$$

where N_i/N is the probability of value i in the selected attribute. However, since sometimes k-ary tree is more effective in practice, deciding more split points can help to form each layer. For each subset, the same process is carried out until all attributes are exhausted. The biggest

concern for the decision tree classifier is over-fitting, since the algorithm can create an excessively complicated tree. Unfortunately, although a tree pruning process is part of the decision tree algorithm, the Sklearn package doesn't support it, so we minimize the over-fitting problem by tweaking the hyper parameters, including *max_depth, mini_sample_split*, etc.

Neural Network:

A Multi-layer Perceptron(MLP) is a class of feed forward artificial neural network. MLP is made up by neurons in different layers. It consists of at least an input layer and an output layer. The number of hidden layers is arbitrary and fully connected by the weights. The MLP is trained literately in two phases:

1. Forward phase: Each neuron starts from the input layer and works towards the output layer.

Its output is $\mathrm{x}_{\mathrm{j}}^{\mathrm{n}} = f(\sum_{i=1}^{m} w_{ji} x_i^{n-1} + b_j^{n-1})$, the function f is an activation function which is commonly either: Linear: f(x)=x, Non-linear: f(x)=tanh(x), or f(x)=1/(1+e-x). The MLP classifier activation 'logistic' is a Sigmoid function. The bias is the threshold θ for f(x).

w_{ji} is the weight connecting the current neuron with the i-th neuron, and x_i^{n-1} is the output of the i-th neuron in layer n-1.

Backward phase: The error is commonly computed by using the Euclidean distance. We have $h(n) = \sum_{i=0}^{\mathrm{p}} x_{i(n)} * w_i$, where n is the nth pattern. P is the number of input neurons.

Therefore, we get $E = \frac{1}{2}\sum_{n=1}^{m} (h_{(n)} - y_{(n)})^2$, where a neuron's output is $h_{(n)}$, $y_{(n)}$ is its actual value, and m is the sample number. Using a gradient descent method $\frac{\partial E}{\partial w_i} = \sum_{n=1}^{m} (h_{(n)} - y_{(n)}) * x_i(n)$, we find the minimum E and the corresponding weight vector in a high dimensional space. In this way, we compute Δw and update the weight between neurons in adjacent layers. The following formula shows how we update the weights.

$$w_i = w_i - a\frac{\partial E}{\partial w_i} = w_i - a\sum_{n=1}^{m} (h_{(n)} - y_{(n)}) x_{i(n)}, \qquad (23)$$

where a is the learning rate. Updating weight can be either synchronous, where all nodes update their activation simultaneously, or asynchronous, which all nodes update their

activation every pattern or every epoch. Finally, we have a fully connected neural network with optimal weights between them for making predictions[17[18][19][20].

SVM:

In the support vector machine (SVM) model, our aims to a separation hyper place with the largest margin between upper and lower planes determine by sample points on both sides, which we call support vectors. However, not all sample datasets with n-dimensional features are linearly separable, in these cases we map them into an x dimensional space using a kernel function, such as RBF, where it is possible to find a perfect hyper place[10][14][15][16].

For Binary classification problems, the linear SVM obtains the hyper place by learning: $w * x + b = 0$.

Distance between the upper and lower hyperplane is called the margin. The larger margin, the more reliable the separation plane is. Given two parallel hyper planes: $w_1 * x + c_1 = 1$ and $w_2 * x + c_2 = -1$, the distance is found according to the formula:

$$\frac{|c2-c1|}{\sqrt{w1^2 w2^2}} = \frac{2}{||w||}, \qquad (24)$$

And the optimal hyperplane is found by maximizing the distance. The second problem is to minimize the error rate. It is defined that all points above the upper hyperplane are in group 1, and all points below are in -1. Therefore, we can know that if we have the right prediction of an unknown sample point, the following inequality is always true, $y_i(w * x_i + b) \geq 1$, where y is the actual label, and xi is the unlabeled vector. So, in entirety, the SVM algorithm is addressing the following formula:

$$\begin{cases} \max \frac{2}{||w||} <=> \min ||w|| \\ y_i(w * x_i + b) - 1 \geq 0 \end{cases} \qquad (25)$$

By using the Lagrange function, the optimization problem can be transformed into finding the lowest value for the following formula:

$$L(w, b, a) = \frac{1}{2} || w ||^2 - \sum_{i=1}^{N} a_i [y_i (w * x_i + b) - 1]. \qquad (26)$$

We know that the minimum of value of L(w,b,a) and the corresponding xi is the point at which the slope of it's tangent plan is 0. By using partial derivation we can get the value of the value of the weight vector and bias as follows:

$$\frac{\partial L(w,b,a)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} a_i y_i x_i \tag{27}$$

$$\frac{\partial L(w,b,a)}{\partial \mathrm{b}} = 0 \Rightarrow \sum_{i=1}^{n} a_i y_i = 0 \tag{28}$$

Conclusion and Analysis:

By researching different machine learning algorithms in mathematical view, we can understand the principles for the above five mainstream techniques. Here is the pros and cons for the five we get by analyzing and some sample experiments. Different approaches are good fits to different fields.

	Key Advantage(s)	Disadvantage(s)	Applicable problem(s)
Gradient descent	1. Easy to approach the approximate global minimum point. 2. High accuracy	1. Comparatively time consuming. 2. Hard to find the exact global minimum point.	1. Regression problem
Bayes decision rules	1. Able to make the best choice decision according to probability	1. Loss matrix must be manually configured.	1. Classification problem
Decision tree	1. Make to best choice in the sample data.	1. over-fitting problem (high accuracy in training but low accuracy in testing) 2. Under-fitting problem.	1. Classification problem
SVM	1. Being able to deal with both linearly separable and non-linearly separable data because of	1. To reach the best performance, technicians must try different kernel functions.	1. Classification problem

	kernelization. 2. High performance.		
Neural network	1. Technicians do not have to set a lot of parameters to tune the network except the number of layers.	1. Comparatively time consuming because of back propagation.	1. Classification problem 2. Regression problem

Future work:

In the whole research, we can understand how these five artificial intelligent algorithms works. After delivering the website platform, we can derive data from users' online behaviors and give recommendations according to the "predicted" potential interests given by algorithms. The combination of different algorithms somehow can generate a stronger and more powerful classifier or regressions model which can provide better performance on our data format and attributes. In the future, we will try implementing those algorithms and tweaking the corresponding parameters to reach the highest precision based on users' data from our website.

Reference:

[1] K. Wei, J. Huang, and S. Fu, “A survey of e-commerce recommender systems," in *Service systems and service management, 2007 international conference on*. IEEE, 2007, pp. 1-5.

[2] R.-C. Chen and C.-H. Hsieh, “Web page classification based on a support vector machine using a weighted vote schema," *Expert Systems withApplications*, vol. 31, no. 2, pp. 427-435, 2006.

[3] A. Sun, E.-P. Lim, and W.-K. Ng, “Web classification using support vector machine," in *Proceedings of the 4th international workshop on Web information and data management*. ACM, 2002, pp. 96-99.

[4] N. S. Raghavan, “Data mining in e-commerce: A survey," *Sadhana*, vol. 30, no. 2-3, pp. 275-289, 2005.

[5] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J.-R. Wen, and X. Li, “Connecting social media to e-commerce: Cold-start product recommendation using microblogging information," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 5, pp. 1147-1159, 2016.

[6] J. L. Ticknor, “A bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5501-5506, 2013.

[7] H. Ogihara, N. Iizuka, and Y. Hamamoto, “Prediction of early recurrence of liver cancer by a novel discrete bayes decision rule for personalized medicine," *BioMed research international*, vol. 2016, 2016.

[8] Y. Uemura, “A study of fuzzy-bayes decision rule with reserved judgement," in *Uncertainty Modeling and Analysis, 1990. Proceedings., First International Symposium on*. IEEE, 1990, pp. 595-602.

[9] J. Li, X. Li, and L. Zhao, “Hyperspectral unmixing via projected minibatch gradient descent," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2017, pp. 1133-1136.

[10] B. Sugiarto, E. Prakasa, R. Wardoyo, R. Damayanti, L. M. Dewi, H. F. Pardede, Y. Rianto *et al.*, “Wood identification based on histogram of oriented gradient (hog) feature and support vector machine (svm) classifier," in *Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017 2nd International conferences on*. IEEE, 2017, pp. 337-341.

[11] C.-S. Rau, S.-C. Wu, P.-C. Chien, P.-J. Kuo, Y.-C. Chen, H.-Y. Hsieh, and C.-H. Hsieh, “Prediction of mortality in patients with isolated traumatic subarachnoid

hemorrhage using a decision tree classifier: A retrospective analysis based on a trauma registry system," *International journal of environmental research and public health*, vol. 14, no. 11, p. 1420, 2017.

[12] C. Verhofstede, K. Fransen, A. Van Den Heuvel, K. Van Laethem, J. Ruelle, E. Vancutsem, K. Stoffels, S. Van den Wijngaert, M.-L. Delforge, D. Vaira *et al.*, "Decision tree for accurate infection timing in individuals newly diagnosed with hiv-1 infection," *BMC infectious diseases*, vol. 17, no. 1, p. 738, 2017.

[13] M. Tayefi, M. Tajfard, S. Saffar, P. Hanachi, A. R. Amirabadizadeh, H. Esmaeily, A. Taghipour, G. A. Ferns, M. Moohebati, and M. Ghayour-Mobarhan, "hs-crp is strongly associated with coronary heart disease (chd): A data mining approach using decision tree algorithm," *Computer methods and programs in biomedicine*, vol. 141, pp. 105-109, 2017.

[14] D. Ahlawat and V. Nehra, "Improved svm classifier based on fodpso and gsa algorithms and its application in face recognition." *IUP Journal of Computer Sciences*, vol. 12, no. 2, 2018.

[15] L. Marlina, A. Roziqin, and A. Kusumorini, "Low-cost portable spectrometer for lard detection based on svm method."

[16] S. Asra and D. Shubhangi, "Human behavior recognition based on hand written cursives by svm classifier," in *Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017 International Conference on*. IEEE, 2017, pp. 260-268.

[17] T. Takase, S. Oyama, and M. Kurihara, "Effective neural network training with adaptive learning rate based on training loss," *Neural Networks*, vol. 101, pp. 68-78, 2018.

[18] M. Ghiassi, D. Zimbra, and S. Lee, "Targeted twitter sentiment analysis for brands using supervised feature engineering and the dynamic architecture for artificial neural networks," *Journal of Management Information Systems*, vol. 33, no. 4, pp. 1034-1058, 2016.

[19] Y. Wang and M. Kosinski, \Deep neural networks are more accurate than humans at detecting sexual orientation from facial images." 2017.

[20] A. Eguchi, J. B. Isbister, N. Ahmad, and S. Stringer, "The emergence of polychronization and feature binding in a spiking neural network model of the primate ventral visual system," *Psychological review*, 2018.