AUT Journal of Electrical Engineering

AUT J. Electr. Eng., 56(3) (2024) 495-502
DOI: 10.22060/eej.2024.22694.5557

# A New Physics-Inspired Discriminative Classifier

Mostafa Monemizadeh [1]*, Seyed Rouhollah Samareh Hashemi [2], Mohsen Sheikh-Hosseini [3], Hamed Fehri [4]

[1] Department of Electrical and Computer Engineering, University of Neyshabur, Neyshabur, Iran
[2] Fiber Optics Group, Institute of Science and High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran
[3] Department of Computer and Information Technology, Institute of Science and High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Irann
[4] Faculty of Engineering, University of Zabol, Zabol, Iran

**ABSTRACT:** Concepts and laws of physics have been a valuable source of inspiration for engineers to overcome human challenges and problems. Classification is an important example of such problems that play a major role in various fields of engineering sciences. It is shown that discriminative classifiers tend to outperform their generative counterparts, especially in the presence of sufficient labeled training data. In this paper, we present a new physics-inspired discriminative classification method using minimum potential lines. To do this, we first consider two groups of fixed point charges (as two classes of data) and a movable classifier line between them. Then, we find a stable position for the classifier line by minimizing the total potential integral on the classifier line due to the two groups of point charges. Surprisingly, it will be shown that the obtained classifier is actually an uncertainty-based classifier that minimizes the total uncertainty of the classifier line. Experimental results show the effectiveness of the proposed approach.

**Review History:**

Received: Sep. 22, 2023
Revised: May, 03, 2024
Accepted: May, 12, 2024
Available Online: Jul. 01, 2024

**Keywords:**

Classification
Discriminative
Machine learning
Potential

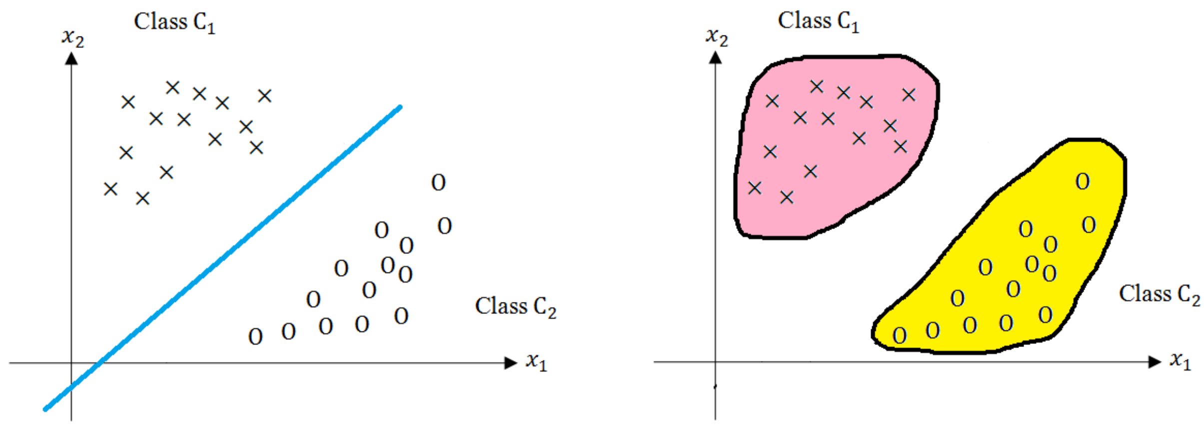
## 1- Introduction

Data classification as a supervised machine learning process involves building a classifier using a set of labeled data samples to predict the labels of new samples. Due to the many applications of the classification task in various and distinct fields, this problem has been heavily researched and many different classification methods have been developed that can be broadly classified into two main groups: generative and discriminative models [1-3]. In generative models (such as naive Bayes classifiers [1, 3, 4] and hidden Markov models [3]), classifying samples is the result of estimating a probability distribution for each class and using Bayes' rule to infer class labels. In contrast, in discriminative models, classifying samples results from directly learning a mapping from samples to class labels. Therefore, in discriminative models, the focus is on decision boundaries, and in generative models is on the data generation process (Fig. 1.). Logistic regression [5, 6], support vector machines (SVM) [7, 8], neural networks [9, 10], decision trees [11, 12], and k-nearest neighbor (KNN) classifiers [13-15] are some examples of widely used discriminative models. Discriminative classifiers often achieve better performance than generative ones, especially in the presence of sufficient labeled training data. When the labeled training data size is small, generative classifiers can outperform them, provided that appropriate models are chosen for the data [1, 3].

Nature has always been a valuable source of inspiration for engineers to overcome human challenges and problems, which has led to amazing results in various fields of engineering sciences. Optimization is one of the most widely used fields in which many nature-inspired algorithms have gained popularity due to their high efficiency. A number of good instances are genetic algorithm [16, 17], ant colony optimization [18, 19], and particle swarm optimization [20, 21].

Machine learning and nature also have long-standing strong links. For example, one such important link was created with the emergence of artificial neural networks [22]. Another such link is physics-inspired classification algorithms such as Coulomb classifiers [23], electrostatic field classifiers [24, 25], and gravitation-based classifiers [26, 27]. Coulomb classifiers [23] are a family of classifiers based on a physical analogy to an electrostatic system of charged conductors. These Coulomb classifiers are trained to minimize the Coulomb energy of three electrostatic systems: (i) uncoupled point charges, (ii) coupled point charges, and (iii) coupled point charges with battery. Electrostatic field Classifier works based on a direct analogy with the electrostatic field, treating all data samples as particles interacting with each other [24, 25]. Two well-performing gravitation-based classifiers are [26, 28, 29] and [27]. The data gravitation-based classification (DGC) method presented in [26, 28, 29] can effectively

*Corresponding author's email: monemi@neyshabur.ac.ir




(a) A discriminative classifier (b) A generative classifier

**Fig. 1. Discriminative models vs. generative models.**

classify a new sample by comparing the gravitational forces of different classes in the training dataset and choosing the class with the highest gravitational force. In contrast, the gravitation-based classification (GBC) algorithm presented in [27] can successfully perform the classification task by minimizing the gravitational potential energy of the classifier line due to two groups of fixed-point masses.

In this paper, a new physics-inspired discriminative classifier using a minimum potential line is presented. To achieve such a classifier, two groups of fixed point charges (as two classes of data) and a movable classifier line between them are first considered. By minimizing the total potential integral on the classifier line due to the two groups of point charges, a stable position for the classifier line is then found. Interestingly, it will be shown that the obtained classifier is actually an uncertainty-based classifier that minimizes the total uncertainty of the classifier line. Experimental results show the effectiveness of the proposed approach.

The remainder of the paper is organized as follows. The main results are presented in Section 2. Some experimental results indicating the effectiveness of the proposed method are given in Section 3. The paper is finally concluded in Section 4.

## 2- Main Results

In this section, we first calculate the integral of the potential on a line due to a point charge. Then, by considering a system consisting of two groups of fixed point charges (as two classes of data) and a movable line (as a classifier line) between them, we find a stable position for the classifier line by minimizing the total potential integral on the classifier line due to the two groups of point charges. Interestingly, we show that the obtained classifier is actually an uncertainty-based classifier that minimizes the total uncertainty of the classifier line. Note that the total potential integral typically represents a measure of the energy or cost associated with a system, and minimizing it can lead to optimal solutions that satisfy certain criteria or constraints. In the context of physics and engineering, there is a strong connection between minimizing the total potential integral, minimizing the total potential energy, and the stability of a system. This connection lies in the fact that minimizing potential integral or energy often leads to a stable equilibrium point where the system is at minimum energy (the principle of minimum total potential energy).

### 2- 1- Necessary preliminaries of electric potential

In a nutshell, the electric potential is the electric potential energy (or work) per unit charge. Practically, the electric potential is a continuous scalar function of position. For example, the electric potential caused by a point charge is continuous in all space except at the point charge location and is inversely proportional to the distance from the point charge.

In order to get the necessary mathematical background of the electric potential on which the proposed classifier is based, consider a fixed point charge $q$ at a distance $r$ from a line of length $L$, as shown in Fig. 2. The electric potential arising from the point charge $q$ at point $x$ on the line is $V_q(x)=\frac{Kq}{\sqrt{r^2+x^2}}$ where $K$ is Coulomb constant. Therefore, the integral of the potential on the line due to the charge $q$ is

$$V_q=\int_{-\frac{L}{2}}^{\frac{L}{2}} V_q(x)dx = 2Kq\int_0^{\frac{L}{2}}\frac{dx}{\sqrt{r^2+x^2}} = 2Kq\left(\ln\left(x+\sqrt{r^2+x^2}\right)\right)\Big|_0^{\frac{L}{2}} = 2Kq\ln\left(\frac{\frac{L}{2}+\sqrt{r^2+\left(\frac{L}{2}\right)^2}}{r}\right). \tag{1}$$

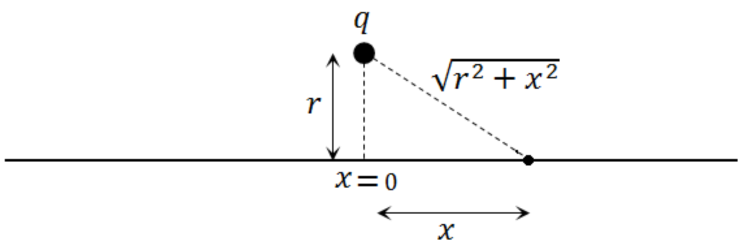



**Fig. 2. A fixed point charge $q$ at a distance $r$ from a line of length $L$ ($L \gg r$).**

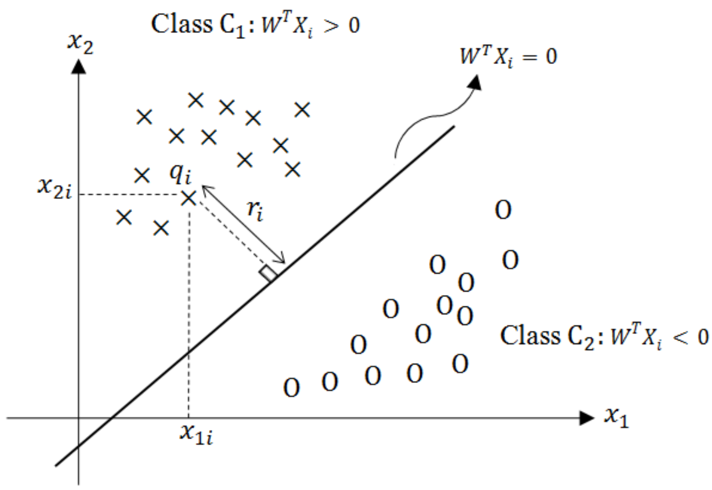



**Fig. 3. Physical model of a linear classifier in two-dimensional space.**

If $L \gg r$ then Eq. (1) reduces to

$$V_q \cong 2Kq \ln\left(\frac{L}{r}\right). \tag{2}$$

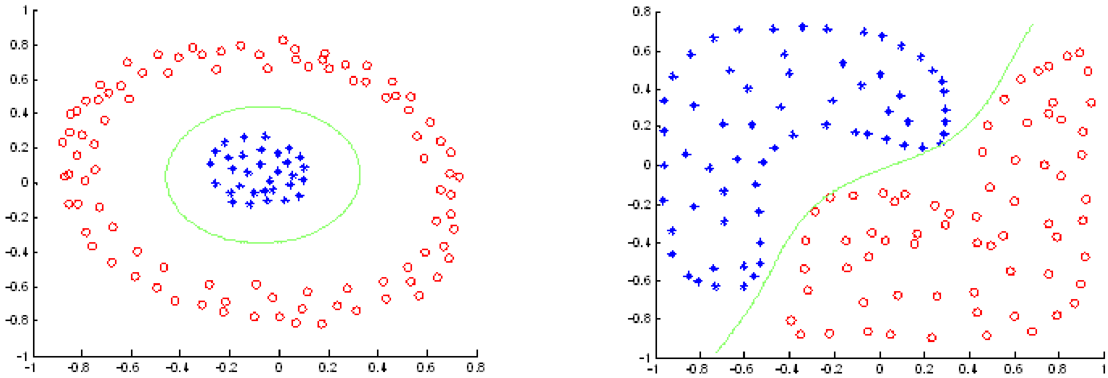
## 2- 2- Classifier design based on electric potential

Consider two groups of fixed point charges (as two data classes $C_1$ and $C_2$) with known coordinates (features) in the Euclidean two-dimensional plane (as a two-dimensional feature space) and a movable line between them (as a classifier line), as shown in Fig. 3. To design a potential-based classifier, we need to find a stable position for the classifier line where the total potential integral over the line due to point charges is minimal (compared to other possible positions). Note that to design the classifier, we ignore the mutual effects of the fixed point charges and only consider the electrical effects of the fixed point charges on the classifier line. Considering (2), the objective function is defined as the total potential integral on the line due to the two groups of fixed point charges (denoted by $V_t$):

$$V_t = \sum_{i=1}^{n_1+n_2} V_{qi} = 2K \sum_{i=1}^{n_1+n_2} q_i \ln\left(\frac{L}{r_i}\right) \tag{3}$$

in which the point charges $\{q_1, \cdots, q_{n_1}\}$ belong to the class $C_1$ and the point charges $\{q_{n_1+1}, \cdots, q_{n_1+n_2}\}$ belong to the class $C_2$. The distance of the point charge $q_i$ from the classifier line, denoted by $r_i$, is

$$r_i = \frac{|W^T X_i|}{\| W' \|} \tag{4}$$



(a) (b)

**Fig. 4. Decision boundary for two synthetically generated datasets.**

in which $X_i = [1, x_{1i}, x_{2i}]^T$, $W = [w_0, w_1, w_2]^T$, and $W' = [w_1, w_2]^T$. Considering (3) and (4), the optimal vector $W^\circ$ that leads to a stable position for the classifier line is obtained by solving the constrained optimization problem

$$\begin{cases} W^\circ = \arg\min_W V_t = \arg\min_W \sum_{i=1}^{n_1+n_2} q_i \ln\left(\frac{L}{r_i}\right) \\ \quad = \arg\min_W \sum_{i=1}^{n_1+n_2} q_i \ln\left(\frac{L\,\|W'\|}{|W^T X_i|}\right) \\ \text{subject to} \begin{cases} W^T X_i > 0, \ i = 1, \ldots, n_1 \\ W^T X_i < 0, \ i = n_1+1, \ldots, n_1+n_2 \end{cases} \end{cases} \quad (5)$$

in which the added constraints guarantee the placement of the classifier line between two groups of point charges.

***Remark 1:*** By defining the probability of belonging of the point charge $q_i$ to its class $p_i = \frac{r_i}{L}$, Eq. (5) states that the classifier line in its stable position has the minimum total (weighted) uncertainty. It is worth mentioning that the uncertainty (information) of the charge $q_i$ at a distance $r_i$ of the classifier line is defined $I_i = -\ln(p_i) = \ln\left(\frac{L}{r_i}\right)$ and it shows that the point charge closer to the classifier line charge has more uncertainty. Moreover, the coefficient $q_i$ in (5) shows the weight (importance) of the charge $q_i$ in designing the classifier line.

## 3- Experimental Results

In this section, we evaluate the performance of the proposed method by testing it on both synthetic and real datasets.

### 3- 1- Results for synthetic dataset

Two synthetic datasets shown in Fig. 4 are constructed to visually demonstrate the effectiveness of the proposed method. As seen in Fig. 4, the experimental results on these synthetic datasets show that the proposed method performs well. It is worth mentioning that synthetic data is often used to visually illustrate the efficacy of a classification method, especially in 2-D or 3-D spaces. These visualizations help researchers and practitioners understand how the algorithm separates different classes in the feature space and assess its performance in discriminating between them. However, there are common metrics in machine learning for providing quantitative results about the performance of a method on synthetic data. Accuracy, as one of the most common metrics, measures the proportion of correctly classified instances out of the total instances in the dataset. Considering this, the accuracy of our method for the synthetic datasets presented in Fig. 4 is 100%. Moreover, Note that although a linear classifier is designed above, it can be extended to separate non-linear separable data by mapping the input data into a higher-dimensional feature space (HDS) where they become linearly separable. Figure 5 shows the proposed method in HDS.

Considering Fig. 5 and (5), the proposed method in HDS is formulated as

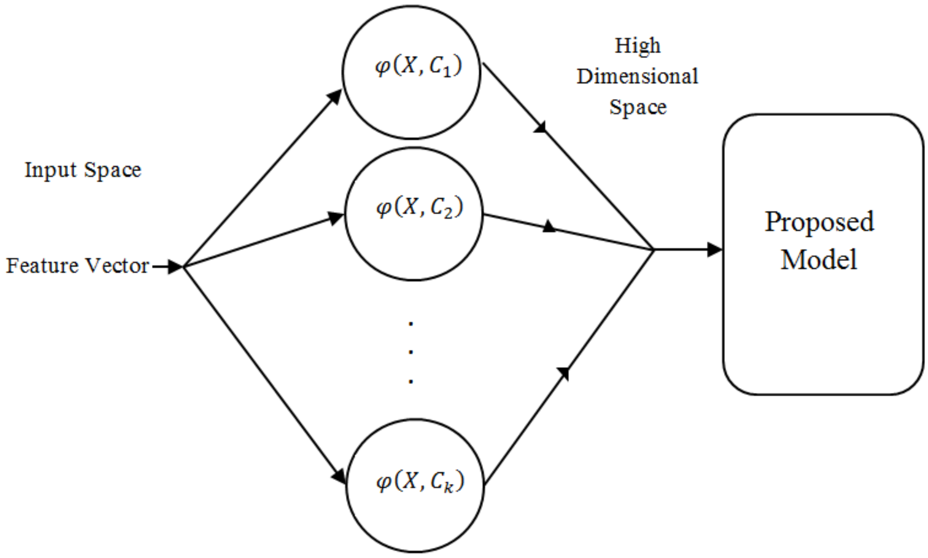



**Fig. 5. The proposed method in HDS.**

$$\begin{cases} W^{\circ} = \arg\min_{W} \sum_{i=1}^{n_1+n_2} q_i \ln\left(\dfrac{L\,\|W'\|}{\left|W^T\varphi(X_i)\right|}\right) \\ \text{subject to} \begin{cases} W^T\varphi(X_i) > 0,\ i=1,\ldots,n_1 \\ W^T\varphi(X_i) < 0,\ i=n_1+1,\ldots,n_1+n_2 \end{cases} \end{cases} \quad (6)$$

Here, for every feature vector $X_i$ in the input space, $\varphi(X_i)=\left[\varphi(X_i,C_1),\varphi(X_i,C_2),\cdots,\varphi(X_i,C_k)\right]^T$ represents the new feature vector in a HDS that is applied to the proposed model as new input data, $W=\left[w_0,w_1,w_2,\cdots,w_k\right]^T$ and $W'=\left[w_1,w_2,\cdots,w_k\right]^T$. In this paper, input non-linear separable data are mapped into a HDS through the Gaussian Radial Basis Function (RBF) kernel. The Gaussian RBF kernel is a popular kernel, calculating the similarity or distance between data points in the transformed space, which is defined as

$$\varphi(X,C_j) = \exp\left(\frac{-\left\|X-C_j\right\|^2}{\sigma}\right) \quad (7)$$

where $\sigma \in R$ is the width of the Gaussian RBF function. This kernel function allows for capturing complex patterns and relationships in the data, making it a versatile and powerful tool in machine learning applications. Figure 4 shows two examples of the proposed method in HDS, where two non-linear separable classes can be effectively separated by mapping into a HDS through the Gaussian RBF kernel. Note that the distance $r_i = \dfrac{\left|W^T\varphi(X_i)\right|}{\|W'\|}$ in (6) is the distance of the point charge $q_i$ from the decision boundary which is a hyper-plane in a HDS.

### 3- 2- Results for real dataset

Some real-world benchmark datasets were selected from the well-known UCI machine learning repository (available at: http://www.ics.uci.edu/~mlearn/databases/) which their main characteristics are depicted in Table 1. In this Table, it is evident that some datasets contain more than two classes, while the proposed method was originally designed for a two-class problem. To address this, the method can be extended to multi-class classification using a "*one-vs-one*" strategy, where each multi-class classification is broken down into individual two-class classification problems for each pair of classes. Consequently, a $K$ -class classification transforms into $K(K-1)/2$ two-class problems. Note that to avoid the dependence of the results on the values of each feature range, first, all features in the datasets are normalized to fall within the common range [0, 1] by using the min–max normalization method. In this method, the normalized feature value $f_{norm}$ for an original feature value $f$ is scaled as follows:

$$f_{norm} = \frac{f - f_{min}}{f_{max} - f_{min}} \quad (8)$$

in which $f_{min}$ and $f_{max}$ are the minimum and maximum values of that feature in the dataset, respectively.

Moreover, 10-fold cross-validation is used to evaluate

**Table 1. Benchmark datasets' summary**

| Dataset name | No. of classes | No. of features | No. of samples |
|---|---|---|---|
| Ecoli | 8 | 7 | 336 |
| Glass | 7 | 9 | 214 |
| Ionosphere | 2 | 34 | 351 |
| Iris | 3 | 4 | 150 |
| Pima | 2 | 8 | 768 |
| Vehicle | 4 | 18 | 846 |
| WBCD | 2 | 10 | 683 |
| Wine | 3 | 13 | 178 |

**Table 2. Accuracy results from 10-fold cross-validation test using the proposed method (in percentage).**

| Dataset name | Accuracy results in Fold # | | | | | | | | | | Best | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fold #1 | Fold #2 | Fold #3 | Fold #4 | Fold #5 | Fold #6 | Fold #7 | Fold #8 | Fold #9 | Fold #10 | | |
| Ecoli | 97.20 | 94.10 | 93.88 | 94.15 | 93.70 | 95.06 | 95.06 | 94.32 | 93.46 | 95.64 | 97.20 | 95.15 |
| Glass | 86.72 | 87.06 | 88.14 | 86.55 | 90.48 | 88.06 | 86.25 | 90.22 | 89.80 | 88.14 | 90.48 | 89.14 |
| Ionosphere | 98.29 | 93.14 | 92.29 | 96.29 | 95.43 | 84.57 | 96.29 | 91.71 | 93.61 | 90.88 | 98.29 | 93.25 |
| Iris | 86.67 | 100 | 100 | 80 | 100 | 100 | 93.33 | 93.33 | 100 | 100 | 100 | 95.33 |
| Pima | 76.62 | 72.73 | 69.74 | 68.83 | 79.22 | 75.33 | 76.62 | 72.73 | 76.32 | 74.03 | 79.22 | 74.22 |
| Vehicle | 80.95 | 75.29 | 74.12 | 69.05 | 83.53 | 83.33 | 71.43 | 64.71 | 75.29 | 85.88 | 85.88 | 76.36 |
| WBCD | 99.12 | 97.10 | 95.74 | 96.32 | 87.35 | 99.27 | 96.91 | 97.25 | 94.41 | 96.96 | 99.27 | 96.04 |
| Wine | 94.12 | 100 | 100 | 94.44 | 94.44 | 100 | 100 | 100 | 88.89 | 100 | 100 | 97.19 |

the proposed model. In 10-fold cross-validation, each dataset is randomly partitioned into 10 approximately equal-sized subsets (or folds); one subset is retained as the test data for evaluating the model, and the remaining nine subsets are used to train the model. Then, this process is repeated 10 times, and each time one of the 10 subsets is used as the test data. Table 2 shows the accuracy results of each fold as well as the best and mean accuracy results from the 10-fold cross-validation test for the different datasets using the proposed method.

Table 3 compares the classification performance of the proposed method with six well-known classification methods, including five discriminative models (SVM, KNN, GBC, DGC+ and FFW-DGC) and one generative model (naive Bayes), on 8 standard datasets shown in Table 1.

As seen in Table 3, the proposed method outperforms the other methods in 3 datasets and obtains competitive accuracy results in the other datasets. To justify this superior performance, we take a closer look at Eq. (5) where the optimal vector $W^{\circ}$ that leads to a stable position for the classifier depends on three factors: (i) the amount of charge "$q_i$", (ii) the distance "$r_i$", and (iii) the "$\log(.)$" function. The weighting factor $q_i$ shows the importance of each data and can be adjusted in such a way that it takes into account a much lower weighting effect for outliers. The distance "$r_i$" has an inverse impact such that the samples that are too far from the other samples or the decision boundary such as outliers, have less importance. The logarithm transformation is a valuable tool in data science, as it has the intriguing capability to

**Table 3. Comparison of mean accuracy results on the standard datasets (in percentage).**

| Dataset name | The proposed method | SVM | KNN | Naïve Bayes | GBC [27] | DGC+ [29] | FFW-DGC [28] |
|---|---|---|---|---|---|---|---|
| Ecoli | 95.15 | 92.30 | 91.11 | 88.34 | **95.50** | 88.98 | 94.36 |
| Glass | **89.14** | 71.50 | 70.11 | 74.65 | 82.30 | 80.72 | 87.38 |
| Ionosphere | **93.25** | 88.00 | 86.60 | 82.40 | 92.28 | 93.11 | 89.58 |
| Iris | 95.33 | 97.33 | 94.00 | 89.43 | 97.2 | 95.33 | **97.38** |
| Pima | 74.22 | 73.87 | 74.20 | 67.88 | 74.03 | **74.51** | 74.20 |
| Vehicle | **76.36** | 75.11 | 68.50 | 72.34 | 71.28 | 71.16 | 74.45 |
| WBCD | 96.04 | 88.00 | **96.80** | 94.76 | 96.01 | 96.19 | 94.76 |
| Wine | 97.19 | **99.33** | 96.07 | 88.19 | 98.30 | 97.31 | 97.45 |
| Average | **89.59** | 85.68 | 84.67 | 82.25 | 88.36 | 87.16 | 88.70 |

reduce the impact of samples located in proximity to or too far from the decision boundary. The combination of these factors in the proposed classifier, which employs the principle of minimum potential energy to establish a stable position, leads to improved robustness, particularly in handling outliers. It is noteworthy that the FFW-DGC method yields results that are nearly comparable to our method (slightly inferior overall). This similarity may stem from the common utilization of the "$\log(.)$" function in both methods, as FFW-DGC also incorporates a mutual information metric. Moreover, as seen in Table 3, in the presence of sufficient labeled training data, discriminative classifiers outperform generative classifiers (here, naive Bayes).

An important factor to consider when selecting a classifier for a particular application is its run time. The run time of a classification algorithm consists of both the training and testing phases. In classifiers such as SVM, a significant portion of the run time is devoted to the training phase, while the testing phase usually occupies a much smaller fraction of the total algorithm run time. Such classifiers are suitable for offline applications because the lengthy training phase does not impact the algorithm's performance. Conversely, in algorithms like KNN, the majority of the run time is consumed after receiving the test data. The FFW-DGC and DGC+ methods are enhanced versions of the nearest-neighbor concept. Despite incorporating some time-saving modifications, these classifiers defer the main computations of the classification phase until receiving the test data, which may be very time-consuming and reduce the efficiency of these methods, akin to the KNN method. In the proposed method and GBC, similar to SVM, the training phase constitutes a very small fraction of the total runtime, while the training phase consumes a significant portion of the runtime. Consequently, owing to its high classification accuracy, our proposed classifier is well-suited for applications where the training phase is conducted offline.

## 4- Conclusion

In this paper, a potential-based discriminative classification method was presented, where the two-class classification problem was modeled as the problem of placing a classifier line between two groups of fixed point charges and finding an equilibrium point for the classifier line. To determine the equilibrium point parameters, we minimized the total potential of the classifier line due to two groups of point charges. Interestingly, we saw that minimizing potential is equivalent to minimizing uncertainty. The effectiveness of the proposed method was validated by some experiments on both synthetic and real datasets.

## References

[1] A. Ng, M. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, Advances in neural information processing systems, 14 (2001).

[2] T. Jebara, Machine learning: discriminative and generative, Springer Science & Business Media, 2012.

[3] C.M. Bishop, N.M. Nasrabadi, Pattern recognition and machine learning, Springer, 2006.

[4] D. Berrar, Bayes' theorem and naive Bayes classifier,

Encyclopedia of bioinformatics and computational biology: ABC of bioinformatics, 403 (2018) 412.

[5] S. Dreiseitl, L. Ohno-Machado, Logistic regression and artificial neural network classification models: a methodology review, Journal of biomedical informatics, 35(5-6) (2002) 352-359.

[6] D.W. Hosmer Jr, S. Lemeshow, R.X. Sturdivant, Applied logistic regression, John Wiley & Sons, 2013.

[7] C. Cortes, V. Vapnik, Support-vector networks, Machine learning, 20 (1995) 273-297.

[8] Z. Akram-Ali-Hammouri, M. Fernández-Delgado, E. Cernadas, S. Barro, Fast support vector classification for large-scale problems, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10) (2021) 6184-6195.

[9] A. Ghods, D.J. Cook, A survey of deep network techniques all classifiers can adopt, Data mining and knowledge discovery, 35 (2021) 46-87.

[10] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: analysis, applications, and prospects, IEEE transactions on neural networks and learning systems, (2021).

[11] Z. He, Z. Wu, G. Xu, Y. Liu, Q. Zou, Decision tree for Sequences, IEEE transactions on Knowledge and Data Engineering, (2021).

[12] S. Tsang, B. Kao, K.Y. Yip, W.-S. Ho, S.D. Lee, Decision trees for uncertain data, IEEE transactions on knowledge and data engineering, 23(1) (2009) 64-78.

[13] Z. Yu, H. Chen, J. Liu, J. You, H. Leung, G. Han, Hybrid $ k $-nearest neighbor classifier, IEEE transactions on cybernetics, 46(6) (2015) 1263-1275.

[14] T. Liao, Z. Lei, T. Zhu, S. Zeng, Y. Li, C. Yuan, Deep metric learning for k nearest neighbor classification, IEEE Transactions on Knowledge and Data Engineering, 35(1) (2021) 264-275.

[15] M. Bansal, A. Goyal, A. Choudhary, A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning, Decision Analytics Journal, 3 (2022) 100071.

[16] D.E. Goldberg, Genetic algorithms, pearson education India, 2013.

[17] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimedia tools and applications, 80 (2021) 8091-8126.

[18] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE computational intelligence magazine, 1(4) (2006) 28-39.

[19] Y. Liu, B. Cao, A novel ant colony optimization algorithm with Levy flight, Ieee Access, 8 (2020) 67205-67213.

[20] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, IEEE, 1995, pp. 1942-1948.

[21] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y.-L. Zhang, Z.-H. Zhan, Triple archives particle swarm optimization, IEEE transactions on cybernetics, 50(12) (2019) 4862-4875.

[22] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the national academy of sciences, 79(8) (1982) 2554-2558.

[23] S. Hochreiter, M.C. Mozer, K. Obermayer, Coulomb classifiers: Generalizing support vector machines via an analogy to electrostatic systems, Advances in neural information processing systems, 15 (2002).

[24] D. Ruta, B. Gabrys, A framework for machine learning based on dynamic physical fields, Natural Computing, 8 (2009) 219-237.

[25] M. Budka, B. Gabrys, Electrostatic field framework for supervised and semi-supervised learning from incomplete data, Natural Computing, 10 (2011) 921-945.

[26] L. Peng, B. Yang, Y. Chen, A. Abraham, Data gravitation based classification, Information Sciences, 179(6) (2009) 809-819.

[27] P. Shafigh, S.Y. Hadi, E. Sohrab, Gravitation based classification, Information Sciences, 220 (2013) 319-330.

[28] L. Peng, H. Zhang, H. Zhang, B. Yang, A fast feature weighting algorithm of data gravitation classification, Information Sciences, 375 (2017) 54-78.

[29] A. Cano, A. Zafra, S. Ventura, Weighted data gravitation classification for standard and imbalanced data, IEEE transactions on cybernetics, 43(6) (2013) 1672-1687.

**HOW TO CITE THIS ARTICLE**

*M. Monemizadeh, S. R. Samareh Hashemi, M. Sheikh-Hosseini, H. Fehri. A New Physics-Inspired Discriminative Classifier. AUT J Electr Eng, 56(3) (2024) 495-502.*

**DOI:** 10.22060/eej.2024.22694.5557