Automatic parametrization of Support Vector Machines for short texts polarity detection

Aurelio Sanabria Rodríguez

Instituto Tecnológico de Costa Rica, Escuela de ingeniería en computación, San José, Costa Rica, 159-7090 *ausanabria@itcr.ac.cr*

and

Edgar Casasola Murillo

Universidad de Costa Rica, Escuela en Ciencias de Computación e Informática, San José, Costa Rica, 10501 *edgar.casasola@ucr.ac.cr*

Abstract

The information from social media is emerging as a valuable source in decision-making, unfortunately the tools to turn these data into useful information still need some work. Using Support Vector Machines for polarity detection in short texts are popular among researchers for their good results, but parameter optimization to train classification models is a complex and costly process. This article compares two algorithms for automated parameter optimization in the process of creating classification models for polarity detection: the recently created Grey Wolf Optimizer and the Grid Search, using accuracy and f-score metrics.

Keywords: Support vector machines, Polarity detection, Grid search, Grey wolf optimizer.

1. Introduction

Since the creation of the World Wide Web in the 90's the humanity has changed the way it collaborates, creates and shares information. The point of view of the collectivity of people often has more impact on the others than expert's opinion [1].

According to Cambria et al:

"Today millions of web-users express their opinions about many topics through blogs, wikis, fora, chats and social networks. For sectors such as e-commerce and e-tourism, it is very useful to automatically analyze the huge amount of social information available on the Web, but the extremely unstructured nature of these contents makes it a difficult task." [1]

Automatic analysis of social text requires the combination of techniques from the fields of Natural Lan-



Figura 1: Differences between Grey Wolf Optimizer and Grid search when exploring parameter space. Light grey spot mark the result of the algorithms

guage Processing (NLP) and Machine Learning (ML) to identify, process and classify opinions according to different criteria. In particular, to classify text according to its polarity (negative or positive), Support Vector Machines (SVM) are popular among researchers [2, 3]. It is noteworthy, SVM performance greatly depends upon the parameter optimization [4]. That process is known as *model selection* [5] which uses one of three different mechanisms: default values, expert criteria and automated selection [6].

This research compares two algorithms for automating parameter optimization on SVM using RBF kernel to identify polarity of short texts (coming from the Twitter social network) in Spanish. Those algorithms are: grid search and the recently created Grey Wolf Optimizer (GWO). Differences in the way both algorithms explore the parameter space can be better understood when displayed in a chart (see figure 1).

2. Related work

2.1. Sentiment analysis

Sentiment analysis, often known as opinion mining, is a research field which works with analysing the opinions in texts [7]. What others think of a person or product becomes important to us. We ask our friends for opinions and recommendations about products, politicians, and even home appliances. In this field, as in many others the internet through blogs, forums, discussions groups and social media has dramatically changed the way we express our opinions. We are no longer limited to ask family and friends for their opinion about an specific item, even companies no longer need big focus groups or external consulting to obtain and understand the users opinion about marketed product [7]. However, using these publicly available opinions is a complex process [8] because manually extracting, processing and visualizing this immense amount of

data becomes prohibitively expensive making this a challenging problem for the fields of Natural Language Processing (NLP) and Machine Learning (ML) [7].

It's important to notice that opinion is a broad concept divided in two types:

- **Direct opinion:** can be defined as a quintuple $(e_i, a_{ik}, so_{ijkl}, h_i, t_l)$ where:
 - e_j is the entity upon which the opinion is given.
 - a_{jk} is the aspect or specific feature of the entity.
 - *so_{ijkl}* is the orientation or polarity of the opinion on the specified feature. This can be positive, negative or neutral with different opinion strengths.
 - h_i is the opinion holder.
 - t_l is the emission date.
- Comparative opinion: Expresses similarities or differences between two or more object. This type is not used in this article

[8]

Being a little less formal, we can say that sentiment analysis can be divided in three different subtasks: opinion extraction, polarity detection and opinion subject relationship. Polarity detection presents particular challenges that need to be addressed [9].

2.2. Polarity detection

Polarity detection is a open research problem considered a subtask of the sentiment analysis field focused in evaluating texts and identifying if those texts contains a negative, positive or neutral orientation[10]. This orientation identified within a large document collection can be useful to competitive analysis, brand management, market analysis, risk management and public opinion analysis for politicians [11].

Notably, from the Machine Learning field, three algorithms used for texts classification receive are applied to identify the polarity value of text. This algorithms are:

- Naïve bayes: probabilistic classifier, characterized by the assumption that the absence or presence of a particular feature isn't related to the other feature.
- Maximum entropy: predicts the class of an instance based in function of the independent variables.
- Support Vector Machines: Given a geometric space, the algorithm tries to find the best possible hyperplane to separate the different classes.

2.3. Support Vector Machines

Current formulation of the Support Vector Machines (SVM), also known as soft margin support vector machines are supervised learning models[12]. Created in 1995 [13], used for classification and regression analysis. Above all, in polarity detection SVM is used for its capabilities as a text classifier where the SVM divides the dataset in two different categories by assigning them a tag y_i (see Equation 1) to a feature vector x.

$$\exists y_i \mid y_i \in \{-1, +1\} \tag{1}$$

The class separation is given by an unknown hyper-plane which is approximated based on a training dataset (see Equation 2)

$$h(x) = w^T x + b \tag{2}$$

Where w is a weight vector, b a threshold value and x an instances of the training dataset (see Equation 3).

$$\exists x \in X | x = (x_i, y_i) \tag{3}$$

Given the approximate hyperplane function, the tag for the instance can be obtain from the sign of the result (see Equation 4).

$$y_i \approx sign(w^T x + b) \tag{4}$$

However, is impossible to work with non-linear separable datasets for linear classifiers, this is why kernel functions needs to be incorporated in order to allow the classifier to be really useful. Some of the most popular kernel functions are mentioned in the next section, nonetheless only RBF kernel function is used in this research.

2.4. Kernel functions

Usually to apply linear classifiers to non-linear data existing on an arbitrary \mathcal{X} dimensional space, the data needs to be transformed to a higher dimensional space \mathcal{Z} where the data can be classified using a linear classifier [14, 15]. Even so, the transformation process is quite expensive in terms of computational resources. Alternatively a *kernel trick* can be applied. The *kernel trick* or *kernel function* [16, 17], executes the essential classifier operation (a dot product) in the \mathcal{Z} dimensional space without transforming the instances from one dimensional space to another.

All kernel function must meet with the Mercer's condition [18], to be valid. In the case that a kernel function does not meet the Mercer's condition there is a risk of not finding a solution[19].

Common kernel functions are[20]:

Lineal

$$K(u,v) = u \cdot v^T \tag{5}$$

Polynomial

$$K(u,v) = (1+u \cdot v^T)^d \tag{6}$$

Where d is the degree of the kernel function.

Radial Basis Functions (RBF)

$$K(u, v) = e^{-\gamma (\|u - v^T\|)^2}$$
(7)

Sigmoid function

$$K(u,v) = tanh(\gamma \cdot u \cdot v^{T} + r)$$
(8)

Assuming u and v feature vectors belonging to a dimensional space \mathcal{X} .

Furthermore, even when the kernel functions result in SVM capable of classifying non-linear data, they introduce the difficulty of parameter optimization for the kernel functions.[21], a problem also known as *model selection* [5] directly impacts the performance of the classifier [21].

For SVM with RBF kernel two configuration parameters are needed: \mathbf{C} which will control the influence of the outlier instances in the final classifier and it's part of the SVM formulation, and Gamma, only RBF Kernel parameter, which will change the way the classes are separated by controlling the dimension of the hyperplane.

2.5. Parameter optimization

For parameter optimization different algorithms are used, grid search [22] is the easiest to implement but it consumes with a high amount of resources[21]. For this reason algorithms with different approximations are constantly tested. For example: genetic algorithms [23] and Simulated annealing [24]. Other algorithms used for parameter selection can be seen in Table 1.

2.6. Grid search

The Grid search algorithms is widely used by the researchers [38, 34, 21, 22, 24, 31, 35], considered the default approach for tuning the parameters SVM [38] is characterized by being exhaustive and providing a high precision but at a cost of time and computing resources. It consist of generating a matrix $A_{m\times n}$ where $a_{i,j} = (C_i, \gamma_j)$ determine the accuracy value by 10-fold cross validation for each $a_{i,j}$. The pair with the biggest accuracy value will be selected as the best parameters.

Considering

Cuadro 1: Algorithms used in parameter optimization

- * Leave One Out [25].
- * Xi Alpha (ξ/α) bound [25]
- * VC bound [25]
- * Sequential Minimal Optimization (SMO) [26, 27], [28].
- * Genetic algorithms [23].
- * Particle Swarm Optimizer[29, 30].
- * Gradient descent [31].
- * Simulated annealing. [24].
- * Evolutionary multi-objective Optimization (EMO) [32].
- * Grid Search [22], [31], [33], [34], [35], [21], [36].
- * Automatic parameter selection[36].
- * Distance between classes [37].

 $C_{k+1} = C_k + d_1$

Where $d_1 \in \mathbb{R}$ chosen by the researcher and $C_k \in [a, b]$ such as a and b are chosen by the researcher.

 $\gamma_{k+1} = \gamma_k + d_2$

Where $d_2 \in \mathbb{R}$ chosen by the researcher and $\gamma_k \in [a, b]$ such as a and b chosen by the researcher.

It is important to notice that the values for d_1 and d_2 must guarantee a low dimensionality of the matrix to avoid a rise in the time needed to complete the calculations but maintaining the result precision unaffected.

Acording to [22] grid search presents the following advantages:

- Allows parallel execution.
- Has complexity of $O(n^2)$
- High precision

With the disadvantage of needing a significant amount of calculation implying a valuable time and resources.

3. Swarm intelligence

Swarm intelligence, according to [39] is inspired by animal behavior when in herds, flocks or colonies. Two fundamental concepts identify algorithms belonging to this category: Self organization and task separation. Self organization refers to individuals capacity to evolve within a system without external stimulus. On the other side, tasks separation corresponds to simultaneous simple task execution by different individuals. Usually these algorithms don't follow leaders command, or a global strategy plan, instead, their global behavior is determined by the agents tasks.

CLEI ELECTRONIC JOURNAL, VOLUME 20, NUMBER 1, PAPER 6, APRIL 2017

Based in the literature review by [40] and [41] belongs to this category the following algorithms:

- Genetic algorithms (GA)
- Ant colony optimizer(ACO)
- Particle swarm optimizer(PSO)
- Differential evolution (DE)
- Artificial Bee Colony(ABC)
- Glow-worms optimizer GSO)
- Cuckoo search algorithm (CSA)
- Genetic programming (GP)
- Evolutionary strategy (ES)
- Evolutionary programming (EP)
- Firefly algorithm(FA)
- Bat algorithm (BA)
- Grey wolf optimizer. (GSO)
- Honey bee mating algorithm (MBO)
- Artificial Fish Swarm Algorithm (AFSA)
- Termite algorithm.
- Wasp swarm algorithm.
- Monkey search.
- Bee Collecting Pollen Algorithm (BCPA)
- Dolphin partner optimizer (DPO)
- Bird Mating Algorithm (BMO)
- Krill Herd (KH)
- Fruit Fly Optimization Algorithm (FOA)

Even when the application of these algorithms to the parameter optimization problem is straightforward, only a few has been applied on Support vector machines [42, 30, 29]. Notably, one of the newest algorithms, the Grey Wolf Optimizer [41], which will explained with some detail in the next section, has an interesting characteristics set [43]:

- Simplicity.
- Flexibility.
- Good local optima avoidance.
- Simple implementation.
- Only two parameters to adjust behavior.
- Fast convergence.

3.1. Grey wolf optimizer

Grey Wolf Optimizer (GWO) [41] and Ant Lion Optimizer [44], both belonging to the newest swarm intelligence algorithms, has already been used to explore solutions to multiple problems [45, 43, 46, 47]. GWO has only two parameters: wolves who will work as search agents and a number of iterations, which will indicate to the algorithms when to stop. Usage of this parameters can be appreciated in the pseudo-code (see figure 4).

The concept of this algorithm is based in the hierarchical behavior of the grey wolves (*Canis lupus lupus*) when hunting [48]. The hierarchy is made up individuals located in different levels that meet specific functions within the pack [49]. The main wolf pack hierarchy levels used by the algorithm can be summarized as follows:

- Alfa wolfa: Current best parameter optimization.
- **Beta wolf** *β*: Second best parameter optimization.
- **Delta wolf** δ : Third best parameter optimization.
- **Omega woves** ω : Work as search agents exploring the parameter space.

Wolf pack hunting process is described in [48], this process includes several actions performed by the wolves. In brief this actions can be separate in three different phases:

- 1. Tracking and approaching the prey.
- 2. Pursuing and encircling.
- 3. Attack towards the prey.

This phases are mathematically model by the algorithm and briefly explained as follows:

1. Encircling the prey



Figura 2: Hunting process, image taken from [41]

To emulate wolf behavior when encircling the prey the following equations were defined:

$$\vec{D} = |\vec{C} \cdot \vec{X_p}(t) - \vec{X}(t)| \tag{9}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \cdot \vec{D} \tag{10}$$

Where t indicates the current version, \vec{A} and \vec{C} are coefficient vector used to balance exploration and exploitation, process that can be seen in figure 2.

2. Hunting:

In nature, the alpha wolf is the first to approach the prey, followed by the beta. Nonetheless, for the algorithm the prey's position is unknown so it needs to be approximated by using the positions of the alpha, beta and delta wolves as describe by the Equations 11 and 12.

$$\overrightarrow{D_{wolf}} = |\overrightarrow{C_1} \cdot \overrightarrow{X_{wolf}} - \overrightarrow{X}| \tag{11} \qquad \overrightarrow{X_1} = \overrightarrow{X_{wolf}} - \overrightarrow{A_1} \cdot (\overrightarrow{D_{wolf}}) \tag{12}$$

Where wolf is one of alpha, beta or delta.

$$\overrightarrow{X_{t+1}} = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_2}}{3}$$
(13)

Figure 3 shows the effect of the equations in the position of the search agents or omega wolves.



Figura 3: position changing during the execution of the algorithm, taken from [41]

3. Search for prey (exploration):

The search begin with the creation of a randomly place omega wolves population (search agents). After every iteration of the algorithm alpha, beta and delta wolves are move to the best possible positions found by the omegas. Initially the exploration is driven by the \vec{A} values provoking wolves dispersion supporting exploration and avoiding local optima.

Notably, \vec{C} (see Equation 12) is not linearly decrease as \vec{A} , instead it always provides random values encouraging exploration.

4. Attack (Exploitation):

The attack toward the prey is model by the vector \vec{A} (present in Equation 12), which constantly decreases during the iterations of the algorithm. When $|\vec{A}| < 1$ the wolf will move closer to the approximated position of the prey; otherwise if $|\vec{A}| > 1$ the search agent will explore other areas away from the prey.

Some examples of successful GWO implementation are the two-stage assembly flow shop scheduling [45] which compares GWO to particle swarm optimizer (PSO) and Cloud theory-based Simulated Annealing (CSA), results of this research shows that GWO got an slightly lower precision but a better performance

Initialize the grey wolf population X_i (i = 1, 2, ..., n)Initialize a, A, and C Calculate the fitness of each search agent X_a =the best search agent X_{β} =the second best search agent X_{δ} =the third best search agent *while* (*t* < *Max number of iterations*) for each search agent Update the position of the current search agent end for Update a, A, and C Calculate the fitness of all search agents Update X_{α} , X_{β} , and X_{δ} t=t+1end while return X_a

Figura 4: GWO Algorithm pseudo-code, adapted from figure 6 in [41]

than the other algorithms. Also in [43], GWO is used for parameter estimation to calculate dispersion curves in surface waves. Good results were obtain thanks to its characteristics, for instance the balance between exploration, fast convergence and the low amount of configuration parameters of the algorithm finally Mirjalili also uses the algorithm to train multi-layer perceptrons (MLP) comparing its performance with PSO, genetic algorithms (GA), Ant colony optimization (ACO), evolutionary strategy (ES) and population based incremental learning (PBIL).[47]

4. Methodology

4.1. Corpus

To compare both parameter optimization algorithms we base our dataset in the TASS corpus. This corpus contains 7220 Tweets messages written in Spanish. Each message is tagged with its global polarity, indicating whether the text expresses one of 6 different possible polarities (categories) : strong positive (P+), positive (P), neutral (NEU), negative (N), strong negative (N+) and no sentiment (NONE). In addition, there is also an indication of agreement or disagreement between annotators. The corpus were pre-processed, merging both positive classes P and P+ into a single category P, same for the negative categories N and N+ which merged into single category N. After that, instances with classed NONE and NEU were deleted and finally all remaining tweets (5068) were turned into vectors of numbers by using a polarity dictionary.

Sixty sets were created using the pre-processed corpus, every set contains two subsets (see figure 5). Train: with 3040 records (60%), for parameter optimization using grid search and GWO and model generation and test: containing 2028 records (40%), for model validation. Subset size selection was made to ensure metrics get a better chance to catch overfitting problems. Then text classifiers models were generated as explained



Figura 5: Randomize subset creation for training and evaluation purposes

in the next section.

4.2. Classifiers generation

	Algorithm	Setting description					
	Default	Models trained using LIBSVM default parameters					
	Grid Search	Models trained using grid search					
	GWO 56	Models trained using GWO with 4 wolves and 14 iterations.					
	GWO 112	Models trained using GWO with 4 wolves and 28 iterations.					
	GWO 168	Models trained using GWO with 4 wolves and 42 iterations.					

Cuadro 2: Different treatment settings for algorithms used. In parenthesis the number of SVM trainings realized by the algorithm

To generate the classifiers and evaluate them, all 60 sets were through a 3 step process:

- 1. Step 1: Parameter optimization, *Train* subsets were used by all different algorithms treatments (see table 2), using the accuracy value and 10-fold cross validation, resulting in 60 (C, γ) pairs, one for each train subset.
- 2. Step 2: Model generation, Using parameter pairs selected in step 1, a classification model is generated using accuracy and 10-fold cross validation.
- 3. Step 3: Model validation, All classifiers are evaluated using the *test* subsets. Accuracy and F_1 -measure metrics were calculated in this step.

4.3. Evaluation metrics

All classifiers were evaluated using the Accuracy (Equation 14) and F_1 score (Equation 15)

$$\operatorname{accuracy} = \frac{T_p + T_n}{T_p + F_p + F_n + T_n}$$
(14)

where T_p are true positives, F_p false positives, F_n false negatives y T_n true negatives.

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{15}$$

5. Results

Results were summarized using box plots created with $GNU \ Octave^1$ (see figures 6 and 7) for visual comparison. Each box represents an algorithm treatment with the top side of the box representing 75th percentile and bottom side 25th percentile, lines coming out of the boxes correspond to maximum (top) and minimum (bottom) values, also the line within represents the average value. See tables 3 and 4 for exact values.

Cuadro 3: F-Score: minimum, maximum, average and standard deviation values

	Default	Grid Search	GWO 56	GWO 112	GWO 168
Minimum	0,5753	0,5894	0,5911	0,5853	0,5909
Maximum	$0,\!6350$	$0,\!6382$	0,6428	0,6380	0,6409
Average	$0,\!6005$	0,6159	0,6173	0,6171	$0,\!6171$
Standard deviation	0,0118	0,0126	0,0108	0,0123	0,0114

 F_1 -measure values were consistently between 0.57 and 0.64 with very small standard deviation indicating that no treatments was considerably better than the others. Also accuracy values behave in a similar way with values between 67.6 and 72.23 also with small standard deviation values.

¹https://www.gnu.org/software/octave/

Cuadro 4: Accuracy: minimum, maximum, average and standard deviation values

	Default	Grid Search	GWO 56	GWO 112	GWO 168
Minimum	67,5542	68,3925	68,4911	68,4911	68,2939
Maximum	71,5483	71,9428	$72,\!2387$	71,9428	71,7949
Average	69,2365	69,9030	$69,\!8997$	69,8948	69,8882
Standard deviation	0,8322	0,9038	0,8810	0,8931	0,8620



Figura 6: Summarize Accuracy results for all 60 test datasets



Figura 7: Summarize F-score results for all 60 test datasets

6. Discussion and Conclusions

All algorithms perform within the same value range, even the models trained using *LIBSVM* default values, for both metrics. We tend to think that this may happened due to several reasons: feature extraction used may result in records dificult to classify or a possible class superposition. Using corpus from different fields like medicine or natural sciences and exploring different feature extraction algorithms may help to clarify this issue.

Notably, the GWO 56 treatment used only half the trainings used by grid search. The other two GWO treatments (GWO 112 and GWO 168) seems to indicate that after a certain amount of iterations there is little or none difference in the results, but further experimentation is needed, specially a parameter sensibility analysis for the grey wolf optimizer is recommended to explore GWO parameters on the results.

Even when our research is focused in global polarity detection for sentiment analysis problems the methodology developed in this experiment can be applied to any field using support vector machines. Finally, automatic parameter optimization is a time consuming and CPU intensive process, for example this experiment took 144 hours, which make it a problem worth considering in future research.

Acknowledgment

This work has been supported in part by the Universidad de Costa Rica (UCR), the Tecnológico de Costa Rica (TEC) and the Ministerio de Ciencia, Tecnología y Telecomunicaciones (MICITT) de la República de Costa Rica. Special thanks to Mario Solís and his work creating a polarity dictionary which was extremely helpful.

Referencias

- E. Cambria, F. K. Uk, and R. Speer, "SenticNet : A Publicly Available Semantic Resource for Opinion Mining," in Artificial Intelligence, 2010, pp. 14–18.
- [2] A. L. F. Alves, C. de Baptista, A. A. Firmino, M. G. D. Oliveira, and A. C. D. Paiva, "A Comparison of SVM Versus Naive-Bayes Techniques for Sentiment Analysis in Tweets : A Case Study with the 2013 FIFA Confederations Cup Categories and Subject Descriptors," WebMedia '14 Proceedings of the 20th Simpósio Brasileiro de Sistemas Multimídia e Web, pp. 123–130, 2014. [Online]. Available: http://dx.doi.org/10.1145/2664551.2664561
- M. Kaya, G. Fidan, and I. H. Toroslu, "Sentiment analysis of Turkish political news," in *Proceedings 2012 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2012.* Ieee, dec 2012, pp. 174–180. [Online]. Available: http://dx.doi.org/10.1109/WI-IAT.2012.115

- [4] B. Sherin and M. Supriya, "Selection and parameter optimization of SVM kernel function for underwater target classification," 2015 IEEE Underwater Technology (UT), pp. 1–5, 2015. [Online]. Available: http://dx.doi.org/10.1109/UT.2015.7108260
- [5] O. Chapelle and V. Vapnik, "Model selection for support vector machines," Advances in neural information processing systems, vol. 12, no. 1, pp. 230–236, 1999. [Online]. Available: http://dx.doi.org/10.1016/j.csda.2007.02.013
- [6] C. Thornton, F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. L.-B. Chris Thornton, Frank Hutter, Holger H. Hoos, "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, aug 2013, p. 847. [Online]. Available: http://dx.doi.org/10.1145/2487575.2487629
- B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and trends in information retrieval, vol. 2, no. 1-2, pp. 1–135, 2008. [Online]. Available: http://dx.doi.org/10.1561/1500000011
- [8] B. Liu, "Sentiment Analysis and Subjectivity," Handbook of Natural Language Processing, vol. 5, no. 1, pp. 1–38, 2010. [Online]. Available: http://dx.doi.org/10.1145/1772690.1772756
- [9] S.-m. Kim and E. Hovy, "Determining the sentiment of opinions," in Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004, p. 1367. [Online]. Available: http://dx.doi.org/10.3115/1220355.1220555
- [10] G. G. Isguder-Sahin, H. R. Zafer, and E. Adah, "Polarity detection of turkish comments on technology companies," in Asian Language Processing (IALP), 2014 International Conference on. IEEE, 2014, pp. 136–139. [Online]. Available: http://dx.doi.org/10.1109/ialp.2014.6973514
- T. Nasukawa, T. Nasukawa, J. Yi, and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," *Proceedings of the 2nd international conference on Knowledge capture*, pp. 70–77, 2003. [Online]. Available: http://dx.doi.org/10.1145/945645.945658
- [12] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 27:1–27:27, May 2011, software available at http: //www.csie.ntu.edu.tw/~cjlin/libsvm. [Online]. Available: http://dx.doi.org/10.1145/1961189.1961199
- C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018
- [14] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, Learning from data. AMLBook, 2012.
- B. Schölkopf, "Learning with kernels," *Journal of the Electrochemical Society*, vol. 129, no. November, p. 2865, 2002.

- [16] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," The annals of statistics, pp. 1171–1220, 2008.
- B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in Proceedings of the fifth annual workshop on Computational learning theory. ACM, 1992, pp. 144–152. [Online]. Available: http://dx.doi.org/10.1145/130385.130401
- [18] V. Vapnik, The nature of statistical learning theory. Springer Science & Business Media, 2013.
- [19] C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121–167, 1998. [Online]. Available: http://dx.doi.org/10.1023/A: 1009715923555
- [20] A. Patle and D. Chouhan, "Svm kernel functions for classification," in Advances in Technology and Engineering (ICATE), 2013 International Conference on. Ieee, Jan 2013, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/ICAdTE.2013.6524743
- [21] C.-H. L. C.-H. Li, C.-T. L. C.-T. Lin, B.-C. K. B.-C. Kuo, and H.-H. H. H.-H. Ho, "An Automatic Method for Selecting the Parameter of the Normalized Kernel Function to Support Vector Machines," 2010 International Conference on Technologies and Applications of Artificial Intelligence, pp. 226–232, 2010. [Online]. Available: http://dx.doi.org/10.1109/TAAI.2010.46
- [22] S.-l. Lin and Z. Liu, "Parameter selection in SVM with RBF kernel function," Journal-Zhejiang University of Technology, vol. 35, no. 2, p. 163, 2007.
- [23] H.-J. Liu, Y.-N. Wang, and X.-F. Lu, "A method to choose kernel function and its parameters for support vector machines," in *Machine Learning and Cybernetics*, 2005. Proceedings of 2005 International Conference on, vol. 7, no. August. IEEE, 2005, pp. 4277–4280.
- [24] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng, "Parameter determination of support vector machine and feature selection using simulated annealing approach," *Applied Soft Computing*, vol. 8, no. 4, pp. 1505–1512, sep 2008. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2007.10.012
- [25] K. Duan, S. S. Keerthi, and A. N. Poo, "Evaluation of simple performance measures for tuning svm hyperparameters," *Neurocomputing*, vol. 51, pp. 41–59, Apr. 2003. [Online]. Available: http://dx.doi.org/10.1016/S0925-2312(02)00601-X
- [26] V. D. Sánchez A, "Advanced support vector machines and kernel methods," Neurocomputing, vol. 55, no. 1–2, pp. 5 – 20, 2003, support Vector Machines. [Online]. Available: http: //dx.doi.org/10.1016/S0925-2312(03)00373-4
- [27] T. Yao and L. Li, "A kernel-based sentiment classification approach for chinese sentences," in Computer Science and Information Engineering, 2009 WRI World Congress on, vol. 5. Ieee, March 2009, pp. 513–518. [Online]. Available: http://dx.doi.org/10.1109/CSIE.2009.117

- [28] M. Davenport, R. G. Baraniuk, C. D. Scott et al., "Tuning support vector machines for minimax and neyman-pearson classification," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 10, pp. 1888–1898, 2010. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.29
- [29] Q. Zong, W. Liu, and L. Dou, "Parameters selection for svr based on pso," in Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, vol. 1, no. 1. Ieee, 2006, pp. 2811–2814. [Online]. Available: http://dx.doi.org/10.1109/WCICA.2006.1712877
- [30] S. Qiu, Z. Li, L. Zhang, Y. Sun, and D. Wang, "Modified particle swarm optimization for multi-scale kernel function in svm," in *Information Engineering and Computer Science (ICIECS)*, 2010 2nd International Conference on. IEEE, 2010, pp. 1–5.
- [31] T. Mu and A. K. Nandi, "Breast cancer detection from FNA using SVM with different parameter tuning systems and SOM-RBF classifier," *Journal of the Franklin Institute*, vol. 344, no. 3-4, pp. 285–311, may 2007. [Online]. Available: http://dx.doi.org/10.1016/j.jfranklin.2006.09.005
- [32] S. Watanabe and Y. Kimura, "A methodology using emo for parameter estimation of svm kernel function," in Soft Computing in Industrial Applications, 2008. SMCia'08. IEEE Conference on. IEEE, 2008, pp. 211–216. [Online]. Available: http://dx.doi.org/10.1109/smcia.2008.5045962
- [33] E.-S. Mahmoud and D. Calvert, "Auto-calibration of support vector machines for detecting disease outbreaks," in *Science and Technology for Humanity (TIC-STH)*, 2009 IEEE Toronto International Conference. Ieee, Sept 2009, pp. 112–117. [Online]. Available: http://dx.doi.org/10.1109/TIC-STH. 2009.5444523
- [34] O. Devos, C. Ruckebusch, A. Durand, L. Duponchel, and J. P. Huvenne, "Support vector machines (SVM) in near infrared (NIR) spectroscopy: Focus on parameters optimization and model interpretation," *Chemometrics and Intelligent Laboratory Systems*, vol. 96, no. 1, pp. 27–33, mar 2009. [Online]. Available: http://dx.doi.org/10.1016/j.chemolab.2008.11.005
- [35] H. Wei-chih and T.-y. Yu, "E-mail Spam Filtering Using Support Vector Machines with Selection of Kernel," *Information and Control*, pp. 764–767, 2009.
- [36] C.-T. Lin, E. Control, and N. Chiao, "AN AUTOMATIC METHOD FOR SELECTING THE PARAMETER OF THE RBF KERNEL FUNCTION TO SUPPORT VECTOR MACHINES," *Measurement*, pp. 836–839, jul 2010. [Online]. Available: http://dx.doi.org/10.1109/IGARSS.2010. 5649251
- [37] J. Sun, C. Zheng, X. Li, and Y. Zhou, "Analysis of the distance between two classes for tuning svm hyperparameters," *Neural Networks, IEEE Transactions on*, vol. 21, no. 2, pp. 305–318, Feb 2010.
 [Online]. Available: http://dx.doi.org/10.1109/TNN.2009.2036999

- [38] E. Carrizosa, B. Martín-Barragán, and D. Romero Morales, "A nested heuristic for parameter tuning in Support Vector Machines," *Computers and Operations Research*, vol. 43, pp. 328–334, mar 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2013.10.002
- [39] Y. F. Zhu and X. M. Tang, "Overview of swarm intelligence," in ICCASM 2010 2010 International Conference on Computer Application and System Modeling, Proceedings, vol. 9, no. Iccasm, 2010, pp. 400–403. [Online]. Available: http://dx.doi.org/10.1109/ICCASM.2010.5623005
- [40] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, "A Comprehensive Review of Swarm Optimization Algorithms," *Plos One*, vol. 10, no. 5, p. e0122827, 2015. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0122827
- [41] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.advengsoft.2013.12.007
- [42] M.-Z. Lu, C. P. Chen, and J.-B. Huo, "Optimization of combined kernel function for svm by particle swarm optimization," in *Machine Learning and Cybernetics*, 2009 International Conference on, vol. 2, no. July. IEEE, 2009, pp. 1160–1166.
- [43] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, and W. Cai, "Grey Wolf Optimizer for parameter estimation in surface waves," *Soil Dynamics and Earthquake Engineering*, vol. 75, pp. 147–157, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.soildyn.2015.04.004
- [44] S. Mirjalili, "The Ant Lion Optimizer," Advances in Engineering Software, vol. 83, no. 0, pp. 80–98, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.advengsoft.2015.01.010
- [45] G. Komaki and V. Kayvanfar, "Grey Wolf Optimizer Algorithm for the Two-stage Assembly Flowshop Scheduling Problem with Release Time," *Journal of Computational Science*, vol. 8, pp. 109–120, 2015.
 [Online]. Available: http://dx.doi.org/10.1016/j.jocs.2015.03.011
- [46] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, no. 30, pp. 1–7, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00521-014-1806-7
- [47] S. Mirjalili, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons," Applied Intelligence, pp. 150–161, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10489-014-0645-7
- [48] C. Muro, R. Escobedo, L. Spector, and R. Coppinger, "Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behavioural processes*, vol. 88, no. 3, pp. 192–197, 2011.
- [49] L. D. Mech, "Alpha status, dominance, and division of labor in wolf packs," Canadian Journal of Zoology, vol. 77, no. 8, pp. 1196–1203, 1999.