The LapRLS Algorithm as a Simple Approach to Sentiment Analysis Lexicon Construction

Brian Keith Norambuena
Department of Computing and Systems Engineering
Universidad Católica del Norte
Av. Angamos 0610, Antofagasta, Chile
brian.keith@ucn.cl

Iván Jirón Araya
Department of Mathematics
Universidad Católica del Norte
Av. Angamos 0610, Antofagasta, Chile
ijiron@ucn.cl

ABSTRACT

This article presents the results and analysis performed on the application of the Laplacian Regularized Least Squares (LapRLS) algorithm on a set of sentiment analysis data. Initially, a description of the method, its parameters, and its characteristics are presented. Then, a case study and some necessary concepts about text representation are detailed. Next, the obtained results in the task of classifying words are shown and analyzed in the context of semi-supervised learning of opinion dictionaries. The analysis of results suggests that the method has shown potential for the construction of opinion dictionaries, obtaining an accuracy of 67.74% on the data set. It should be noted that these results could possibly be improved by applying an extensive optimization of the parameters of the LapRLS algorithm.

Keywords: Sentiment analysis, Lexicon construction, LapRLS algorithm

INTRODUCTION

The aim of this research work is to present an application of the spectral graph theory in the field of neural networks, specifically the Laplacian Regularized Least Squares (LapRLS) algorithm. Initially, in order to do this, a description of the LapRLS algorithm is presented and then the algorithm is applied in a case study in the field of Sentiment Analysis.

Sentiment analysis can be defined informally as the process by which it is determined whether a phrase or act of speech contains an opinion, positive or negative, about a specific entity or concept. Sentiment analysis is a complex process because it attempts to recognize a pattern in a text expressed in natural language, usually in a specific domain [1].

Sentiment analysis and opinion mining can provide support and help in the following tasks and areas: market intelligence [2], observing the attitudes of the populace towards political movements [3], box office prediction for feature films [4], determining the level of consumer satisfaction with a product or service [5], among others [1, 6].

The effective development of opinion mining systems has a large number of challenges. First, it is necessary to identify the contents in a text, this task is not trivial due to the nature of language, which has a large number of semantic subtleties that are not present in other types of data. Second, sentiments must be classified in some way and thus determine their orientation. There are different ways of approaching this problem [7].

Creating sentiment lexicons for specific domains is an important task in sentiment analysis [8]. These lexicons allow researchers and practitioners to analyze key subjective properties of texts [9]. There are two main schemes to obtain a sentiment lexicon: corpus-based techniques and dictionary-based techniques [8]. In particular, corpus-based approaches require seed words and use pattern recognition techniques to induce sentiment lexicons [10, 11]. Dictionary-based approaches use manually crafted resources [12], such as ontologies as WordNet [13].

One of the possible approaches used in corpus-based techniques is to use lexical graphs using word co-occurrences and then propagate labels [14, 15]. In the work of Hamilton et al. [8], the authors combine domain-specific word embeddings and apply a label propagation framework in order to induce an accurate sentiment lexicon for this domain, starting from a small set of seed words.

Our proposal uses a similar approach, using a graph-based approach (through the application of the LapRLS algorithm) based on word embeddings obtained with word2vec.

LAPRLS ALGORITHM

The following section is based on the description given by Haykin [15]. Given a set of training examples $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ a weighted and non-directed graph G is constructed. Each example will be represented by a vertex (that is, the example $\mathbf{x}_i$ will be associated with the vertex i). To determine if two vertices are adjacent, the following criterion of adjacency based on the Euclidean norm will be used, as defined in Formula (1).

$$i \sim j \Leftrightarrow \|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \varepsilon \quad (1)$$

Taking this adjacency criterion into account, the weights of each edge are defined. To determine the weights w_{ij}, two cases must be considered. The first corresponds to vertices *i* and *j* being adjacent. In this case, the weight of the corresponding edge is defined by Formula (2).

$$w_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (2)$$

The second case occurs when the vertices *i* and *j* are not adjacent. In this situation, it is defined that $w_{ij} = 0$ to satisfy the connectivity condition [15] (i.e. there is no edge between *i* and *j*).

The Regularized Least Squares algorithm is taken as a base and the new intrinsic regularization term is added. The error function is written in matrix notation as

$$\mathcal{E}_\lambda(\mathbf{a}) = \frac{1}{2}(\mathbf{d} - \mathbf{JKa})^T(\boldsymbol{d} - \boldsymbol{JKa}) + \frac{1}{2}\lambda_A \mathbf{a}^T\mathbf{Ka} + \frac{1}{2}\lambda_I \mathbf{a}^T\mathbf{KLKa}, \quad (3)$$

where the following notation has been used:

- Vector of expected responses $\mathbf{d} = [d_1, d_2, \ldots, d_l]^T$ of order $N \times 1$. Note that the last entries are filled with zeroes, this is done in order to ensure that the matrix operations have consistent dimensiones. Filling these entries with zeroes does not affect the results because the matrix makes sure that unlabeled entries do not influence this term of the error, in fact, any other value could be used as a placeholder for these entries.
- Vector of expansion coefficients $\mathbf{a} = [a_1, a_2, \ldots, a_N]^T$ of order $N \times 1$.
- Diagonal matrix partially filled with l unit entries, $\mathbf{J} = \text{diag}[1,1,\ldots,1,0,0,\ldots,0]^T$ of order $N \times N$, that allows to distinguish which elements have a known response and which do not (allows separating the supervised from the unsupervised part). Note that **J** is symmetric because it is diagonal.
- Kernel or Gram matrix **K** of order $N \times N$, the entries are defined as $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ where $k(\cdot,\cdot)$ is the kernel function.
- Laplacian Matrix **L** of the constructed graph of order $N \times N$.

The proposed model has the following parameters:

- Regularization parameters: λ_A and λ_I.
- Graph parameters: ε, σ^2
- If $\lambda_I = 0$ it is reduced to the case of classic regularization

The kernel function to be used corresponds to a Gaussian kernel, defined in Formula (4) as

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}. \quad (4)$$

The penalty term $\mathbf{a}^T\mathbf{Ka}$, under the control of the ambient regularization parameter λ_A reflects the complexity of the approximation function F in the ambient space. In particular, this penalty term can be expressed in terms of a reproducing kernel Hilbert space (RKHS) [15], hence the K subscript. The penalty term $\mathbf{a}^T\mathbf{KLKa}$, under the control of the intrinsic regularization parameter λ_I reflects the underlying geometric structure of the input space, which is why subscript I [16] is used. This underlying geometric structure can be modeled in different ways, in the case of the LapRLS algorithm an undirected weighted graph is used [17].

To find the optimum, the total error is derived with respect to **a**, so that the optimal value is given by

$$\mathbf{a}^* = (\mathbf{JK} + \lambda_A \mathbf{I} + \lambda_I \mathbf{LK})^{-1} \mathbf{J}^T \mathbf{d}.$$

Having found the optimal value of the representation coefficients, the LapRLS method is formally stated in Algorithm 1.

Algorithm 1 LapRLS Algorithm

Input: Dataset X with l labeled data points and $N - l$ unlabeled data points; label vector **d**; parameter ε for the adjacency criterion; parameter σ^2 for weight computation; regularization parameters λ_A and λ_I.
Output: Optimal coefficients $\boldsymbol{a}$

function LapRLS
1. G = build-graph(X, ε, σ^2)
2. **K** = calculate-gram(X)
3. **L** = calculate-laplacian(G)
4. $\boldsymbol{J}$ = get-label-matrix(X, **d**)
5. $\mathbf{a}^* = (\mathbf{JK} + \lambda_A \mathbf{I} + \lambda_I \mathbf{LK})^{-1} \mathbf{J}^T \mathbf{d}$
6. return $\mathbf{a}^*$

end function

Each part of the algorithm is briefly described:

1. *Build-graph* function: constructs the undirected weighted graph G of N nodes using the adjacency criterion and the function for the weights.
2. *Calculate-gram* function: calculates the Gramian matrix of the kernel for the data set.
3. *Calculate-laplacian* function: calculates the Laplacian matrix of graph G.
4. *Get-label-matrix* function: obtains the matrix **J** that indicates the labeled and unlabeled elements.
5. The optimal coefficient vector is calculated and returned.

The vector of optimal coefficients is used in conjunction with the generalized representation theorem to approximate the function $F_\lambda^*(\mathbf{x})$. Then the approximation of the function F at the point **x** is obtained by the formula given by

$$F_\lambda^*(\mathbf{x}) = \sum_{i=1}^{N} a_i^* k(\mathbf{x}, \mathbf{x}_i). \quad (6)$$

CASE STUDY

The problem addressed in this section corresponds to a subproblem within the field of sentiment analysis that is relatively simple but illustrative. Specifically, the objective of this case study is to show the applicability of the LapRLS algorithm to determine the semantic orientation of a word (i.e. to determine if it is a positive or negative word).

The development of this case study requires elements of linear algebra since words can be represented by vectors of real numbers, these vectors are obtained from the frequency analysis on several text sets. The text representation model that will be used in this work is word2vec as implemented in the open source genism library for Python [18], as detailed in the following paragraphs.

Figure 1 provides an overview of the steps that will be followed in this case study. In particular, a list of words will be converted by word2vec to their respective vector representation. Then, the graph required by the LapRLS algorithm will be constructed. Finally, LapRLS will be applied to obtain the semantic orientation of each word.

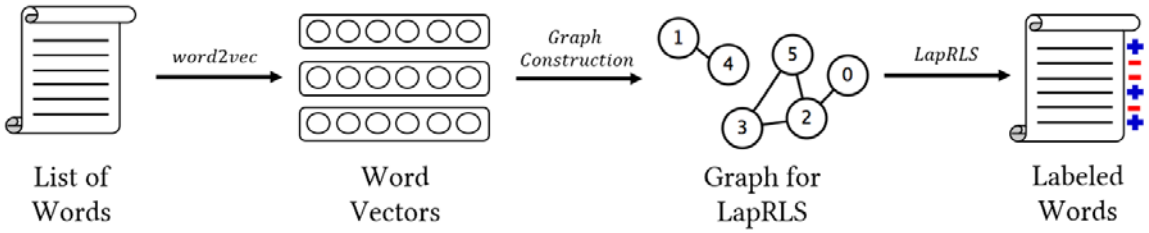

Figure 1. Overview of the Case Study.

Some of the words will be previously labeled with their orientation. The data set used has 93 words, of which 20 are labeled (10 positives and 10 negatives). For the purposes of this case study, only positive and negative cases will be considered, the case of neutral or objective words will be excluded. Neutral cases have been omitted because sentiment analysis with a neutral class is a harder problem and many sentiment analysis works omit multiclass classification [1]. It should be noted that it would be possible to consider varying degrees of semantic orientation, turning this into either a multiclass classification problem or a regression problem depending on the nature of these degrees of semantic orientation (discrete vs continuous). However, for simplicity only a binary classification case has been addressed in this work, considering potential extensions to the other cases as future work.

Next, the structure of the data set will be explained by an example, consider the following list of words in Table 1.

Table 1. List of example words, N/A expresses that a class has not been assigned to it.

Word	**Vector**	**Orientation**
Good	(0.56, 0.78)	+1
Bad	(-0.25, -1.23)	-1
Adequate	(0.43, 0.28)	N/A
Adverse	(-0.54, -0.76)	N/A
Perfect	(1.12, 0.83)	N/A

Using the words that have labels as a base, the semi-supervised LapRLS algorithm must be able to predict the semantic orientation of the other words without labels. Once the words have been represented as real n-dimensional vectors, the algorithm will work in the same way as presented in the previous chapter.

Although in practical situations the labels of all the data would not be available, in order to illustrate the correct functioning of the algorithm, the results are evaluated by comparing the semantic orientation assigned to the word with respect to the true orientation of the term. The evaluation is done by calculating the *accuracy* (acc) metric, defined as $acc = c/n$, where c represents the number of correctly evaluated words and n the total number of words evaluated. The final result will be expressed in terms of percentages. The result will be considered successful if it exceeds 50% accuracy, because that means that it is better than making a random choice between both classes.

TEXT REPRESENTATION

In this section, models to represent the text by means of numerical vectors will be discussed. The space formed by these vectors is called a semantic vector space. There is an underlying hypothesis shared by all models of semantic vector spaces, called the statistical semantics hypothesis: "Statistical patterns of word usage can be used to deduce what people are trying to communicate." That is, in vector terms, if the text units have similar vectors (where the similarity is expressed in terms of their proximity as vectors in the corresponding space), then they tend to have similar meanings [18].

The simplest vector representation corresponds to bags of words, in which each word of the dictionary is associated with a position in the vector. The element in this position will take the value one if it represents that word and zero otherwise. This generates a representation of the words in the space $\mathbb{R}^{|V|\times 1}$, where $|V|$ is the size of the dictionary. It can be seen that this representation is not efficient, since it does not capture the relationship between words and also generates a sparse representation of the text [1]. To address these problems, the conventional approach is to perform a singular value decomposition on a co-occurrence matrix X [6].

Although these methods provide useful vectors for encoding the semantic-syntactic information of words, they are associated with a series of problems. Mainly, the high dimensionality of the space, the cost of training, the fact that the matrix is sparsely populated, the sensitivity with respect to the introduction of new words in the dictionary and changes in the size of the set of documents [18]. A proposal to solve these problems is to create a model that learns in an iterative way and that allows coding the probability of occurrence of a word given its context (understanding its context as the set of C words that surround it). The learning of this model is based on *backpropagation*, in which the error is calculated in each iteration and the model is corrected accordingly [19].

The *word2vec* method is based on a two-layer neural network to process the text. Its input is the text data set and its output is the set of vectors that represent the words in a semantic vector space [19]. The main purpose and utility of *word2vec* are that it allows to group vectors of similar words (semantically and syntactically) in the vector space, that is, it detects the similarities mathematically. One of the most interesting results of the representation in *word2vec* is that it is very appropriate to encode different dimensions of similarity. The *word2vec* method encodes each word in a vector and the model is trained with respect to the words that surround it (i.e. its context).

RESULTS AND DISCUSSION

The parameters used for the LapRLS algorithm are the following: $\sigma = 0.005$, $\lambda_A = 1.0$, $\lambda_I = 0.5$ and $\varepsilon = 0.01$. A simplified version of *word2vec* has been used. Specifically, the dimensionality of vectors has been reduced by principal component analysis (PCA) [20]. Each word is then represented by a two-dimensional vector.

In Table 2 it can be seen that positive words are classified better (the algorithm only failed 9 times out of 50), while with negative words the algorithm presents more problems, although mostly it is still more accurate than a random choice. Using the information in the table, the accuracy can be calculated, obtaining that $acc = 67.74\%$. According to the criterion defined above, this result can be considered as successful.

Table 2. Confusion matrix with the results of the case study.

Confusion Matrix		Real class		
		Positive	Negative	**Total**
Predicted class	Positive	41	21	62
	Negative	9	22	31
	Total	50	43	93

From Table 2 additional performance metrics can be obtained, such as precision and recall. In particular, precision is equal to 66.13% and recall is equal to 82.00% for the positive class, while precision is equal to 70.97% and recall is equal to 51.16% for the negative cases. For positive words this approach has given relatively good results, however, for negative words the algorithm has a harder time, specifically, it assigns some positive words as negative more often than expected. This could be due to the small size of the dataset, which does not allow for an adequate generalization.

It should be noted that if the Regularized Least Squares algorithm is used without the Laplacian component (that is, $\lambda_I = 0$), the algorithm is unable to distinguish between the two classes. In this case, it can only correctly classify the previously labeled examples. All instances without labels are assigned a zero value. This is to be expected given the supervised nature of the original algorithm, but it also highlights the importance of the internal geometric structure of the data, since adding this information to a relatively simple algorithm increases its learning capacity.

It is possible to make some interesting observations about the obtained graph. First, the graph is connected, since it has an algebraic connectivity of 0.0096, which is greater than zero [21]. It should be noted that by reducing the value of the parameter ε that defines the adjacency criterion, it is possible to construct a graph with several connected components. Second, it is also possible to calculate the energy of the graph associated with the data, obtaining $E(G) = 1.6844$. Reducing the value of the parameter ε also results in a reduction in energy, due to its relation to the number of connected components.

One of the possible applications of these results is the construction of a dictionary of opinion words. That is a list indicating the semantic orientation of each term. This list can be used by more advanced methods of sentiment analysis that allow determining the semantic orientation of complete documents.

Considering the difficulty of the problem of sentiment analysis, the following observations should be made:

- It is possible that, with a larger data set, the results obtained might be better, because the algorithm could extract more information and detect more patterns.
- The reduction in dimensionality made by analyzing principal components may have impaired the classification performance. However, the use of this tool facilitates the visualization of the word vectors.
- It is possible that there might be a different parameterization that allows obtaining better classification results. This could be improved by an iterative search on the parameter space. However, as mentioned above, the search for optimal parameterization has been considered outside the scope of this work.

Even taking into account these observations and that the results are possibly improvable, the case study fulfills its objective. It has been shown that the LapRLS algorithm is able to correctly classify the semantic orientation (for the most part) of a set of words. The LapRLS algorithm, based on concepts of regularization theory, the problem of least squares and spectral graph theory can be applied in the training of a neural network for the construction of opinion lexicons.

The search for the optimal parameterization of the algorithm is considered, in general, a complex problem in the field of machine learning. In this particular case, the search for optimal parameterization has been considered outside the scope of this work; on the other hand, the illustration of the method and its possible practical applications have been emphasized. However, the search process of optimal parameters is briefly discussed.

Although there are some heuristics for the regularization parameters, in general, this process is performed empirically [15]. Specifically, a possible strategy to perform this task is a brute force approach in which each possible combination of parameters is evaluated, given upper and lower limits for each one, as well as an increment for each parameter that together would determine the number of total tests to be performed (i.e., a grid search).

Finally, it is also necessary to comment on the limitations of the LapRLS algorithm. Although it has been tested on several real data sets, the predictability of this algorithm comes at the cost of a high computational complexity. This is due to the inclusion of the intrinsic regularization term, whose calculation has an associated order of magnitude $O(N^3)$, where N is the size of the data set (considering labeled and unlabeled elements). The high computational complexity of this method makes it difficult to apply it on real data sets that have a large amount of data; the development of semi-supervised machine learning methods is an area of active research [15].

CONCLUSIONS

The present work has explored one of the applications of the spectral graph theory on the field of neural networks. Specifically, we have studied the LapRLS algorithm, a generalization of the LMS algorithm that uses as a foundation the Laplacian matrix of a graph to find the synaptic weights of a neural network. Furthermore, this method has been applied in a case study within the field of sentiment analysis, where a simplified version of the problems faced in that area has been resolved.

As mentioned during the development of this work, opinions are central to almost all human activities, which is why sentiment analysis is a broad field in which there are numerous applications. A subproblem within this field has been explored by applying the LapRLS algorithm. Specifically, the classification of words into positive or negative was explored. Although this problem reflects only part of the complexity associated with sentiment analysis, it has been useful to illustrate the operation of the LapRLS algorithm.

Finally, in future work, the search for optimal parameters and a detailed parameter tuning guide for the algorithm in this setting would prov ide the first line of research. Furthermore, applying the LapRLS algorithm on different and more varied data sets would provide a second line of research.

Acknowledgments: The authors would like to acknowledge the financial support of VRIDT/UCN N° 84/2018.

REFERENCES

1. B. Liu, Web data mining: exploring hyperlinks, contents, and usage data. Springer Science & Business Media, 2011.

2. Li, Y.M., Li, T.Y.: Deriving market intelligence from microblogs. Decision Support Systems 55(1) (2013) 206–217
3. Tumasjan, A., Sprenger, T.O., Sandner, P.G., Welpe, I.M.: Predicting elections with Twitter: What 140 characters reveal about political sentiment. ICWSM 10(1) (2010) 178–185
4. Nagamma, P., Pruthvi, H., Nisha, K., Shwetha, N.: An improved sentiment analysis of online movie reviews based on clustering for box-office prediction. In: Computing, Communication & Automation (ICCCA), 2015 International Conference on, IEEE (2015) 933–937
5. Ren, F., Quan, C.: Linguistic-based emotion analysis and recognition for measuring consumer satisfaction: an application of affective computing. Information Technology and Management 13(4) (2012) 321–332
6. Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowledge-Based Systems 89 (2015) 14–46B. Pang and L. Lee, "Opinion mining and sentiment analysis," Found. Trends Inf. Retr., vol. 2, no. 1-2, pp. 1–135, Jan. 2008.
7. S. S. Haykin, Neural networks and learning machines. Pearson Upper Saddle River, NJ, USA, 2009, vol. 3.
8. Hamilton, W. L., Clark, K., Leskovec, J., & Jurafsky, D. (2016, November). Inducing domain-specific sentiment lexicons from unlabeled corpora. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing (Vol. 2016, p. 595). NIH Public Access.
9. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2), 267-307.
10. Widdows, D., & Dorow, B. (2002, August). A graph model for unsupervised lexical acquisition. In Proceedings of the 19th international conference on Computational Linguistics - Volume 1 (pp. 1-7). Association for Computational Linguistics.
11. Riloff, E., & Shepherd, J. (1997). A corpus-based approach for building semantic lexicons. arXiv preprint cmp-lg/9706013.
12. San Vicente, I., Agerri, R., & Rigau, G. (2014). Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (pp. 88-97).
13. Fellbaum, C. (1998). A semantic network of English verbs. WordNet: An electronic lexical database, 3, 153-178.
14. Huang, S., Niu, Z., & Shi, C. (2014). Automatic construction of domain-specific sentiment lexicon based on constrained label propagation. Knowledge-Based Systems, 56, 191-200.
15. Velikovich, L., Blair-Goldensohn, S., Hannan, K., & McDonald, R. (2010, June). The viability of web-derived polarity lexicons. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 777-785). Association for Computational Linguistics.
16. Sindhwani, M. Belkin, and P. Niyogi, The Geometric Basis of semi-supervised Learning. Citeseer, 2006.
17. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” Journal of machine learning research, vol. 7, no. Nov, pp. 2399–2434, 2006.
18. P. D. Turney, P. Pantelet al., "From frequency to meaning: Vector space models of semantics," Journal of artificial intelligence research, vol. 37, no. 1, pp. 141–188, 2010.
19. T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, 2013.
20. I. Jolliffe, Principal component analysis. Wiley Online Library, 2002.
21. M. Fiedler, “Algebraic connectivity of graphs,” Czechoslovak mathematical journal, vol. 23, no. 2, pp. 298–305, 1973