Support Vector Mind Map of Wine Speak

Brendan Flanagan[1(✉)] and Sachio Hirokawa[2]

[1] Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan
b.flanagan.885@s.kyushu-u.ac.jp

[2] Research Institute for Information Technology, Kyushu University, Fukuoka, Japan

Abstract. Models created by blackbox machine learning techniques such as SVM can be difficult to interpret. It is because these methods do not offer a clear explanation of how classifications are derived that is easy for humans to understand. Other machine learning techniques, such as: decision trees, produce models that are intuitive for humans to interpret. However, there are often cases where an SVM model will out preform a more intuitive model, making interpretation of SVM trained models an important problem. In this paper, we propose a method of visualizing linear SVM models for text classification by analyzing the relation of features in the support vectors. An example of this method is shown in a case study into the interpretation of a model trained on wine tasting notes.

Keywords: Model visualization · SVM · Support vector weight

1 Introduction

SVM models have often been described as black box models because there is no comprehendible explanation or justification of the trained model and the classifications derived from it [1]. This stems from a lack of evidence as to how the model was derived from the analysis of the input data. Other machine learning methods, such as: decision trees, are easy for humans to interpret classification explanations and justifications by simply observing a visualization of the model. However, there are cases where an SVM trained model is a better fit for a classification problem, and will outperform other methods that generate models that are simple to interpret. A method often used in the interpretation of Linear SVM models for text classification is the extraction of feature weight [2]. However this method does not explain how different features in the model are related. In Sect. 2, we propose that the relation structure of features can be extracted from support vectors to help interpret the characteristics of an SVM model for text classification. Then in Sect. 3, the extracted relation characteristics are analyzed to generate visualizations of the model in the form of positive and negative feature trees. These two trees can be thought of as representing the

S. Yamamoto (Ed.): HIMI 2016, Part I, LNCS 9734, pp. 127–135, 2016.
DOI: 10.1007/978-3-319-40349-6_13

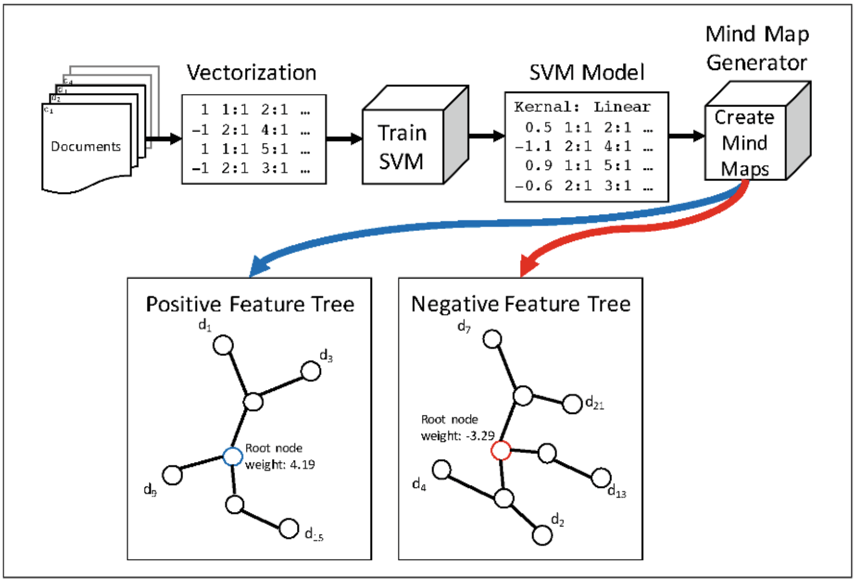

Fig. 1. Overview of the the visualization of SVM models as a Mind Map

support vectors that are placed on the positive and negative sides of the decision hyperplane, and will be referred to as the positive feature tree and negative feature tree respectively. The root node of the trees are the features that have the greatest positive and negative weight across all support vectors respectively. A case study of the proposed method is provided to demonstrate its application on real world data in Sect. 4, and how the generated visualizations can be interpreted. An overview of the proposed method is shown in Fig. 1.

2 Feature Scores and Relations

2.1 Extraction from Support Vectors

The output of linear kernel model trained in SVM^{light} [3] contains the support vectors V that describe the decision hyperplane, in which each support vector $v_j \in V$ is made up of a vector weight $\alpha_j y_j$ and the original feature vector $\{w_1, \ldots, w_n\}$ from the data analyzed in the training process. Equation 1 determines the weight of a feature word w_i.

$$weight(w_i) = \sum_{v_j \in V} \alpha_j y_j TF(w_i, v_j) \tag{1}$$

Where $TF(w_i, v_j)$ is the term frequency of the word w_i in the support vector v_j. To classify a document $d_l \in D$, the $score(d_l)$ in Eq. 2 is evaluated with respect to the model bias b.

$$score(d_l) = \sum_{w_i \in d_l} weight(w_i) \tag{2}$$

A linear SVM model can be interpreted by analysis and visualization of the feature vectors and $\alpha_j y_j$ weight of support vectors. In the present paper, we propose that visualization by automatically generating trees for the positive and negative features contained in the support vectors of the model. To generate the visualizations, first we analyzed the relation of the feature vectors and apply the vector weight and other characteristics of the features to create a ranking of the features and their relations.

2.2 Feature Scoring and Relation Representation

As described in the previous section, the score of an individual feature word can be calculated from the support vectors contained in the model as seen in Eq. 1. In addition to this method, attributes of features can be used in calculating a word score for ranking. In Eq. 3 the document frequency of a word $DF(w_i)$ is used to give more importance to words that occur in many documents.

$$WS(w_i) = \sum_{v_j \in V} \alpha_j y_j TF(w_i, v_j) DF(w_i) \tag{3}$$

A word co-occurrence matrix that describes the relations of features can be generated by analyzing the features that occur within the same support vector. A naïve co-occurrence frequency could simply be the number of support vectors in which two feature words have occurred. However this does not take into account the weights of the support vectors in the model. We calculate the values describing co-occurring feature word as seen in Eq. 4.

$$CS(w_u, w_v) = \sum_{v_j \in V} \alpha_j y_j \frac{1}{2} \sum_{w_t \in \{w_u, w_v\}} TF(w_t, v_j) DF(w_t) \tag{4}$$

3 Visualization Method

By analyzing the relations of features in the support vectors of Linear SVM models, we propose that the visualization of two trees representing the positive and negative features can be useful in model interpretation. A complete graph of the relation of features in the support vectors of the Linear SVM model can be generated by analyzing the Jaccard distance of pairs of features. The similarity of two feature word nodes u and v is calculated using the formula in Eq. 5.

$$Similarity(w_u, w_v) = \frac{CS(w_u, w_v)}{WS(w_u) + WS(w_v) - CS(w_u, w_v)} \tag{5}$$

Where $CS(w_u, w_v)$ represents the score of support vectors that the two features co-occur in, and $WS(w_u)$ and $WS(w_v)$ represent the score of the support vectors that the features w_u and w_v. Visualization of the relations of the features as a complete graph would be difficult to interpret as there is a large number of edges connecting all the nodes of the graph [4]. To help overcome this problem, we search for a minimum spanning tree of the complete graph that is made up of the strongest relations between the feature nodes. The pseudocode in Algorithm 1 searches for the minimum spanning tree by first creating a matrix of edges that are selected by finding the maximum similarity for nodes of decreasing importance.

```
Data: w
Result: nodes, edges
sort w in decending order by WS(w_i);
dim relmatrix[|w|][|w|];
for u = 1; u < |w|; u++ do
    maxsim = 0;
    from = 0;
    for v = 0; v < u; v++ do
        if CS(w_u, w_v) > maxsim then
            maxsim = CS(w_u, w_v);
            from = v;
        end
    end
    relmatrix[u][from] = maxsim;
end
for u = 1; u < |w|; u++ do
    nodes.addnode(u) unless nodes.u exists;
    for v = 0; v < u; v++ do
        nodes.addnode(v) unless nodes.u exists;
        edges.addedge(u,v);
    end
end
```

Algorithm 1. Tree generation algorithm

After the maximum similarity edge matrix is determined the graph is constructed by creating all the nodes and joining the edges found by the search. When generating tree for positive or negative features, the set of features w is limited to only positive or negative features respectively. This ensures that the trees do not contain overlapping features.

4 Case Study: Interpreting SVM Models of Wine Sensory Viewpoints

In previous work, we have analyzed wine tasting notes using SVM [5]. The data analyzed is a corpus that consists of 91,010 wine tasting notes, or 255,966 sentences, that were collected from the Wine Enthusiast website[1]. A subset of the

[1] http://buyingguide.winemag.com/.

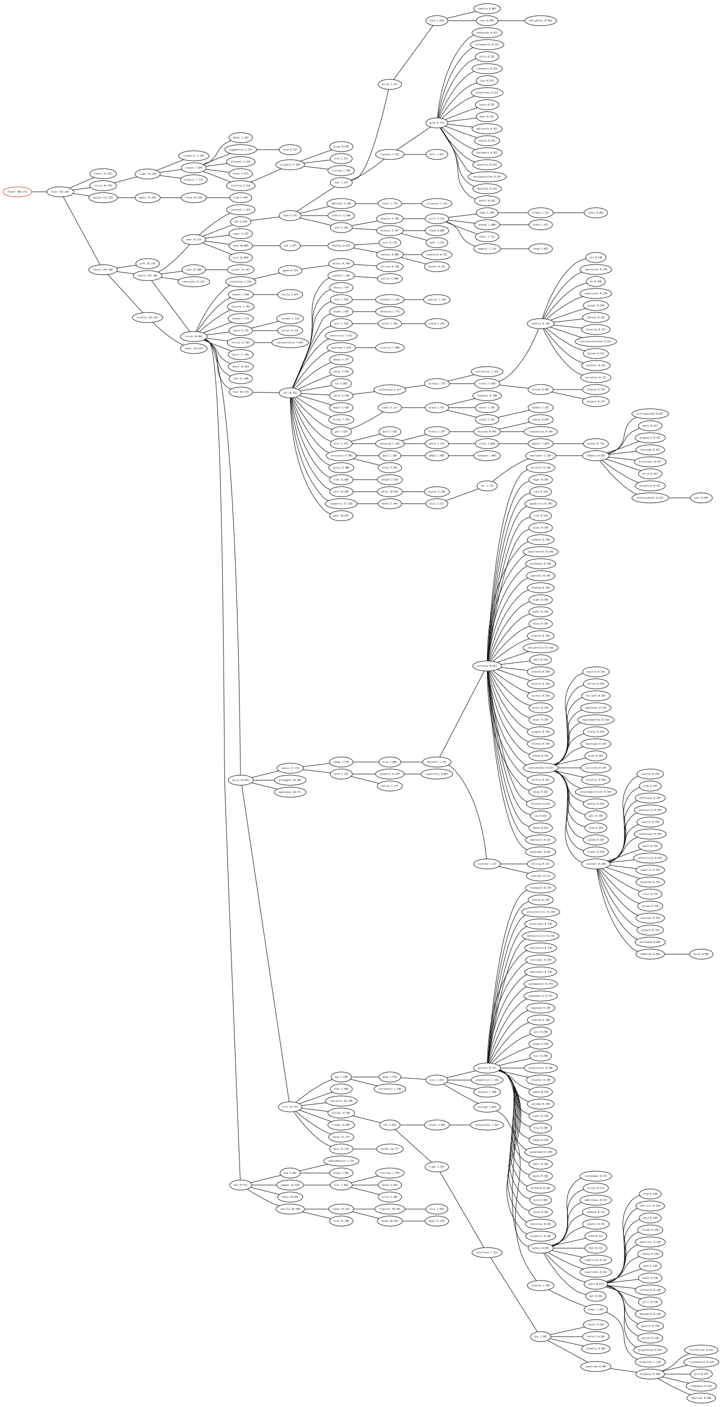

Fig. 2. Positive feature tree of the taste sensory model.

data consisting of 992 sentences from wine tasting notes was randomly selected for use in the training, testing and evaluation of sensory sentiment models. This data subset was manually classified by hand into four different sensory category viewpoints, as defined by Paradis and Eeg-Olofsson [6]. Optimal feature selection was achieved by a subset of 600 top positive and negative features.

In the present paper, we visualize the taste sensory viewpoint SVM model as a case study of interpreting linear models using our system to automatically generate positive and negative feature trees as seen in Figs. 2 and 3. Overall the two trees have quite different structures. The positive tree has groupings of words around hub words, whereas the negative tree has less groupings of features.

A possible explanation for this is that the positive tree only contains features from one class, the taste class, and the negative tree contains features from at least three different classes: smell, touch, and vision. As these trees have

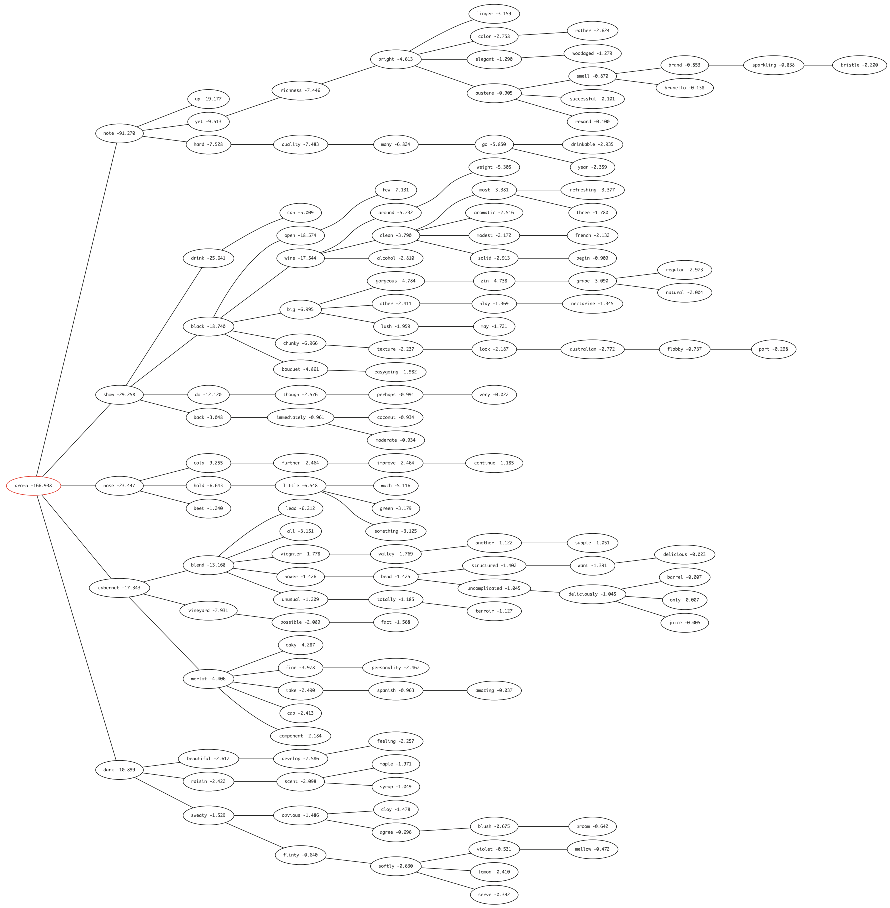

Fig. 3. Negative feature tree of the taste sensory model.

numerous nodes, we will focus our interpretation on a small section of the positive tree. We decided to target the *finish* feature node as it has many first generation child nodes. Table 1 contains a sample of the child nodes and an example wine tasting note that is representative of the support vectors in which the features co-occurred.

The child node word and all the parent node words have been hi-lighted, showing that some of the support vectors extend from the root node, while others are only locally represented by *finish* and a child node. This hierarchal structure represents the *finish* characteristic of a wines taste. This structure would not be apparent just by examining a simple ranking of feature words by WS, as seen in Table 2, as other features in separate surrounding branches share similar scores, but belong to a different taste sub-characteristic. The proposed method enables the interpretation of the structure of related features and their relevance to the SVM model.

Table 1. Child nodes of *finish* feature node from the positive tree and example wine tasting notes they represent.

Child node	Representative wine tasting note
Hint	Its full-bodied, with **hints** of smoke, vanilla and herbs that add complexity to the **cherry fruit**, while on the **finish**, the **tannins** are soft and the **flavors** linger elegantly. (Tapestry 2004 Cabernet Sauvignon)
Spice	The palate is huge and tannic but luxurious, and the **flavors** of tobacco, cedar, lemon peel, **spice** and black **fruit** set up a **finish** with espresso, black **cherry** and a wall of **tannins**. (Numanthia-Termes, S.L. 2009 Termanthia Tinta de Toro)
Sharp	Acidic, with a **sharp finish**, but ultimately it's clean. (Casa Julia 2003 Sauvignon Blanc)
Fruity	It could use more **fruity** concentration, though, as its thin and watery in the **finish**. (Easton 2004 Sauvignon Blanc)
Touch	Snappy on the **finish**, with a **touch** of caramel sweetness. (Llai Llai 2009 Pinot Noir)
Almond	It's a very well-executed wine with softer shades of vanilla and **almond** paste and a smooth, long-lasting **finish**. (Guicciardini Strozzi 2005 Vignarè Red)
Velvety	Thick and smooth, it has a **velvety**, **fruit**-driven **finish**. (Massolino 2007 Nebbiolo)
Tomato	The **finish** is quite herbal, and there's more than an accent of **tomato**. (Terrapura 2008 Merlot)
Chocolaty	The **finish** is warm and **chocolaty**, and as a whole it's delivering a lot more depth and quality than most $14 wines. (Vinã Santacruz 2006 Chama'n Gran Reserva Syrah)

Table 2. Rank of features by $WS(w_i)$ from *finish* to *hint*.

Rank	Feature	$WS(w_i)$
8	Finish	94.0618744453973
9	Oak	79.9109859470339
10	Vanilla	46.9400117871891
11	Citrus	46.4532939237085
12	Spice	44.0763867784245
13	Tannic	42.5907366996148
14	Taste	39.853999024464
15	Soft	38.1486820604408
16	Chocolate	33.5414635606771
17	Hint	30.7294073932907

5 Related Work

Previous work into the extraction of features for interpretation of SVM models has focused on creating rules that describe the classifications made by the model. Barakat et al. [1], argued that the interpretation of a model is an important step to gaining acceptance when applying black box machine learning techniques in medical settings. They proposed the extraction of rules from models to aid medical understanding into the classifications made for the diagnoses of type 2 diabetes. Few features were used in the study, which would make interpretation by rules applicable. In the present paper, we focus on methods for interpreting text classification models, which have numerous features, making rule based interpretation an unfeasible option.

In other previous work, the authors have visualized the contents of wine tasting notes from the perspective of words describing sensory modalities by a system that automatically generates radar charts of the predicted value by SVM model [5]. This method only analyzes the predicted document $score(d_l)$, and does not provide insight into the interpretation of the model. We previously examined the visualization of a corpus of documents by analyzing the SMART weight of single words and pairs of words selected by AND boolean search [4]. However, it was not possible to use a search engine for the visualization of Linear SVM models as the system needs to take into account both positive and negative scored words. To overcome this problem, we propose analyzing the ranking of features by $WS(w_i)$ and the relation of features by co-occurrence matrix.

6 Conclusion

Blackbox machine learning techniques are difficult for humans to interpret due to the lack of explanation on how classifications are derived. Other machine learning techniques, such as: decision trees offer a model that is easy for human's to interpret. However, a short coming of these techniques is that they are often out preformed by blackbox trained models, such as: SVM. In this paper, we proposed a method for extracting the relations from support vectors contained within a trained SVM model. These relations were then analyzed to automatically generate two trees that represent the positive and negative features of the SVM model. In a case study on the interpretation of an SVM model that classifies the taste sensory viewpoint of wine tasting notes, we found that the proposed method can reveal structures in the model that can be interpreted as sub-characteristics.

In future work, the propose method should be compared to other machine learning methods to evaluate the effectiveness of the visualization for model interpretation.

Acknowledgment. This work was supported by JSPS KAKENHI Grant Number 15J04830.

References

1. Barakat, M.N.H., Bradley, A.P.: Intelligible support vector machines for diagnosis of diabetes mellitus. IEEE Trans. Inf. Technol. Biomed. **14**(4), 1114–1120 (2010)
2. Sakai, T., Hirokawa, S.: Feature words that classify problem sentence in scientific article. In: Proceedings of the 14th International Conference on Information Integration and Web-Based Applications and Services, pp. 360–367 (2012)
3. Joachims, T.: Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms. Kluwer Academic Publishers, Berlin (2002)
4. Hirokawa, S., Flanagan, B., Suzuki, T., Yin, C.: Learning winespeak from mind map of wine blogs. In: Yamamoto, S. (ed.) HCI 2014, Part II. LNCS, vol. 8522, pp. 383–393. Springer, Heidelberg (2014)
5. Flanagan, B., Wariishi, N., Suzuki, T., Hirokawa, S.: Predicting and visualizing wine characteristics through analysis of tasting notes from viewpoints. In: Stephanidis, C., Tino, A. (eds.) HCII 2015 Posters. CCIS, vol. 528, pp. 613–619. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21380-4_104
6. Paradis, C., Eeg-Olofsson, M.: Describing sensory experience: the genre of wine reviews. Metaphor Symb. **28**(1), 22–40 (2013)