Net Promoter Sentiment Classifier Using OHPL-ALL

Bob Vanderheyden
PhD Candidate
Kennesaw State University
Kennesaw, GA, USA
rvanderh@students.kennesaw.edu

Ying Xie
Dept. of Information Technology
Kennesaw State University
Kennesaw, GA, USA
ying.xie@kennesaw.edu

Mohan Rachumallu
Market Development & Insights
IBM
Bangalore, KA, India
morachum@in.ibm.com

Abstract— **Net Promotor Score is an important business measurement process where customers are surveyed and asked to rate their likelihood of recommending the company's products and/or services. In many applications, customers are asked to respond on an 11-point ordinal scale of 0 to 10. In developing the score, the data are reformulated into a labelled 3 class scale (0-6: Detractor, 7-8: Passive and 9-10: Promoter). [1] Many companies that choose to use Net Promoter Score as a core management metric integrate the measurement into all phases of the company and seek every opportunity to assess company performance in terms of likelihood to promote the company. In addition to a variety of survey opportunities, the ability to score comments in survey, social media and blogs with promoter rating may provide an additional valuable source of business insight. Even on a three-point scale, Net Promoter is an ordinal classification problem. A number of successful algorithms, that develop ordinal classifiers have been developed. [2] None of the top performing classifiers can be used for applications like text classification or image classification, since they don't employ deep learning. Any appropriate strategy must utilize the ordering information of classes without imposing a strong continuous assumption or fixed spacing assumption on the ordinal classes. In this paper, we use a novel Deep Learning methodology called OHPLnet (Ordinal Hyperplane Loss Network) that is specifically designed for data with ordinal classes. [3] The algorithm is used to develop predictions of the eleven classes, that may be used in the standard Net Promoter Score generation process.**

Keywords—ordinal hyperplane loss, ordinal classification, deep learning, machine learning, Net Promotor Score, NPS

I. Introduction

The problem of ordinal classification occurs in a large and growing number of areas. Some of the most common sources and applications of ordinal data are:

- Ratings scales (e.g. Likert scales), like customer satisfaction ratings, "promoter" ratings and quality ratings
- Medical classification scales (e.g. classification of disease stage/severity) and student performance (i.e., letter grades)
- Socio-Economic scale (e.g., high, medium andlow)
- Meaningful groupings of continuous data (e.g., generational age groupings, grouping of noisy sensor data)
- Facial emotional intensity [1]
- Large storm severity ratings (e.g., Tropical Storms and Hurricanes)

Historically, data sources like surveys and medical ratings were relatively small in size, but this digitalized world has produced more and more truly big ordinal data sources, such as Amazon's purchase satisfaction surveys, Yelp's rating data, and electronic health records.

Ordinal data differ from nominal (unordered) data by providing additional information on the order of the classes, which leads to a different way to evaluate the results of classification. For instance, misclassifying a value of '3' as a value of '4' should be viewed as a "better" error than misclassifying it as a '5' for ordinal classification, although nominal classification treats these two error cases equally.

In Ordinal Hyperplane Loss, Vanderheyden and Xie introduce a new loss function that can be used to develop a new deep learning methodology, that can be used to develop models that predict ordinal class labels. Their methodology was demonstrated to be superior to a wide variety of high performing ordinal classifiers, when applied to structured ordinal problems that are used to benchmark algorithms. [3] The research in this paper demonstrates the power of OHPLnet when applied to unstructured data. In particular, when solving a text ordinal classification problem.

The majority of Net Promoter systems utilize survey data, where customers are asked rate the company, its services or its products, on the customer's likelihood of recommending the company to their friend and/or colleagues. [4] Net Promoter metrics have been demonstrated to correlate future revenue increases for companies. [1] The survey may include the opportunity for the customer to provide open ended text

Research funded in part by US NSF Grant SES-1853191

commentary related to their rating. These text comments combined with the customer rating provide an opportunity for the business to develop a "sentiment" like analyzer, that may be applied to other customer comments and/or social media comments.

The rest of the paper will be organized as follows. In section II, we report, in detail, our literature study. In section III, we review OHPLnet, as an evolutionary development of OHPL. The problem specifics and results from applying OHPLnet to an NPS text analysis problem, by developing an NPS text classifier is provided in section IV. Finally, we conclude our paper in section V.

II. Background

A. Original Ordinal Hyperplane Loss

The goal of OHPL is to provide a deep neural network with an appropriate loss function to estimate a nonlinear mapping of data, into a space that not only separates the classes but does so in a way that the separation maintains the ordering of the classes, in the new space (see Figures 1(a) and 1(b)). OHPLnet gets its name by using hyperplane centroids to represent the class centers space (see Figures 1(a) and 1(b)). [3] Enforcing a minimum distance between the centroids is a critical component of the process. [3]

Fig 1(a): Unseparated Fig 1(b): Separated

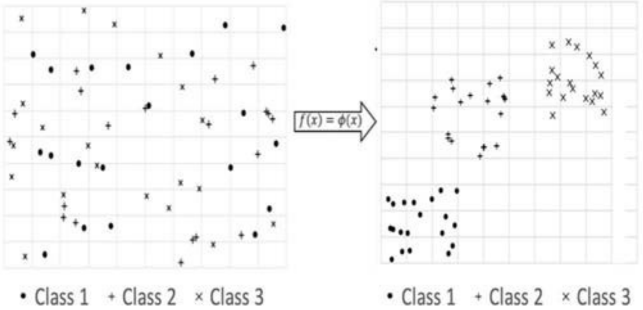


Fig 1(a): Unseparated Hyperplane Centroids.

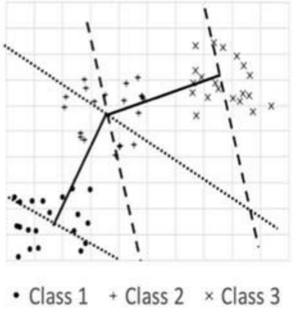
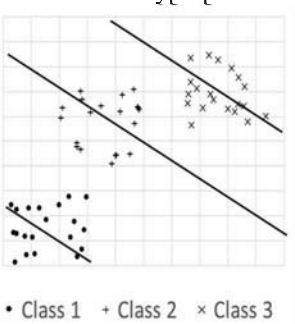


Fig 1(b): Separated Parallel Hyperplane



3]

The fundamental component of OHPL is the application of a unique loss function that uses L1 (absolute) distance to both assess "error" for the ordering and spacing of the hyperplane centroids as well as measuring the distance for points from their corresponding hyperplane centroid. [3]

One way to define linear hyperplane is as a set of points that satisfy a simple mathematical equation of the form:

$$w^T x + C = 0 \quad (1)$$

where $w$and $x\mathbf{x}$ are vector valued and $\mathbf{c}$ is a scalar constant. [3]

Different parallel hyperplanes the form in (1), differ in their constant values (i.e., $\mathbf{c}$ values). To extend the concept, the 'distance' between two parallel hyperplanes can be defined to be the absolute value of the difference in their $\mathbf{c}$values. [3]

The **Hyperplane Centroid (HC)** for the kth hyperplane centroid for the kth class, denoted as $HC_k HC_k$, is determined by:

$$HC_k: w^T\mathbf{x} - \frac{1}{n_k}\sum_{y_i=k} w^T x_i = 0 \quad (2)$$

Using the definition in (2), all ordinal classes are represented as the mean of the corresponding hyperplane centroids. The hyperplane centroids are parallel to each other, in the feature space.

Hyperplane Centroid Loss (HCL), is the class ordering component of OHPL which ensures the proper ordering of the hyperplane centroids and there for the ordering of the classes. If we ensure that the adjacent HCs are properly ordered, then by the transitive property all HC's are properly ordered. In aggregate, the total error in assessing the ordering of the hyperplane centroids can be expressed as

$$HCL = \sum_{i=1}^{k-1} \max(HC_i - HC_{i+1} + o, 0) \quad (3)$$

where $o > 0$ is a minimum margin, to ensure nontrivial distances between hyperplane centroids. In practice, o is set to a value of 1. In its execution, the algorithm uses the full dataset to establish the hyperplane centroids. Experiments on structured datasets resulted in classifiers that perform as well or better than a set of high performing benchmark algorithms. [3] Even with data sets with over 200K records, the algorithm was able to establish the hyperplane centroids, using the full training dataset.

"Hyperplane-Point Loss" " (HPL) is the second component of OHPL is "Hyperplane-Point Loss. For each point. the loss component is calculated as the L1 distance between the point and its corresponding hyperplane centroid. A margin is employed, so trivial distances from the hyperplane centroid are set to zero and optimization focuses on points that are closer to other hyperplane centroids (i.e., not their own) or are sufficiently close to the midpoint between the point's hyperplane centroid and the adjacent hyperplane centroid, though still closer to its own hyperplane centroid. [3] Fig 2 illustrates the HPL loss for points that are sufficiently far from their respective hyperplane centroids.

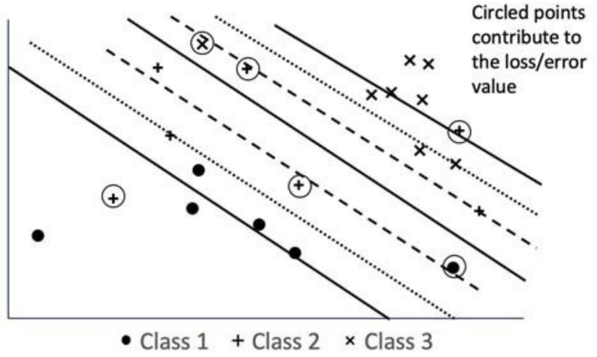


Fig 2 Hyperplane Point Loss

HPL is the sum of all of the individual errors. HCL and HPL are combined to create Ordinal Hyperplane Loss. To ensure that the ordering of classes is a priority, in the optimization, a weight may be applied to HCL.

HPL provides a "pull" of points, to make them closer together, while HCL is a contrasting "push" of points to establish and maintain a minimum distance between hyperplane centroids. For some datasets, this contrast isn't excessively "combative," while in others, it's a full battle between the two. In the original work, the algorithm assessed the ordering to allow for a reduction in the ordering weight, if the minimum distance between adjacent hyperplane centroids is greater than *o.*

B. OHPLall

Additional research and experimentation lead to advances in OHPLnet. Large datasets, of unstructured data, were unlikely to be able to be analyzed using the original variant of OHPLnet. Multiple variants of OHPLnet were created and tested. Two of the variants used stratified sampling strategies, for the establishment of the hyperplane centroids. One created a stratified sampling at the initialization of each epoch, while the other developed a single stratified sample that was used for all epochs. The stratified sampling approaches performed well and represented a meaningful advancement in OHPLnet.

A number of different variants and analysis strategies, were developed from the original OHPL work. This work will be published in separate reports. While testing these OHPL variants and strategies, on a large image classification problem, none of the variants except the mini-batch variant were not able to process images, using a CNN, due to memory limitations on a Nvidia GTX 1080 ti GPU that has 10 GB of memory. The problem required the processing of 20 or fewer images. As would be expected, the limitation became more restrictive, as the number of output channels for the first convolutional layer increased.

While the original mini-batch variant was able to process the images, it struggled to achieve the proper ordering and spacing of the hyperplane centroids. In the executions of the mini-batch variant on the images there was a single class that was out of order. If an extremely high weight was assigned to the HCL component of loss, the algorithm struggled to minimize HPL. To attempt to overcome this issue, a new variant of the mini-batch variant was developed. One simple change result in a very meaningful improvement in performance on the image classification problem. The HCL loss component was changed to compare all classes that were represent in the mini-batch to the other classes within the batch. The margin must be appropriately adjusted to account for ordinal "distances" that are greater than 1 (i.e., cases were the labels differ by more than 1). The new formula for HCL is documented in equation (4) [5].

$$HCL = \mathbf{L}_{i<} \max HC_i - HC + (\quad - \quad)\quad o, 0 \qquad (\quad)$$

The performance of this new variant of OHPLnet was so good on the image problem, that we decided to apply it to the NPS text sentiment analysis problem that is reported in the next section in preparation of applying the algorithm to larger text classification problems in the future [5].

The new algorithm, called OHPLall performed consistently better than the original OHPL. MZE results versus the benchmark datasets, reported in the original OHPL research are reported in Table 1 [5].

Table 1 MZE for Variants of OHPL

	Original OHPL	OHPLall
CPU Small	0.542	**0.516**
Census 10	**0.646**	0.681
Cars	0.024	**0.014**
Wine-Red	0.444	**0.418**
ERA	0.772	**0.755**
LEV	0.412	0.412
SWD	0.427	**0.407**

In terms of MAE, OHPLall performs better than the original OHPL on all seven benchmark data sets. MAE results are reported in Table 2 [5].

Table 2 MAE for Variants of OHPL

	Original OHPL	OHPLall
CPU Small	0.763	**0.709**
Census 10	1.267	**1.199**
Cars	0.024	**0.014**
Wine-Red	0.520	**0.457**
ERA	1.790	**1.543**
LEV	0.460	**0.442**
SWD	0.473	**0.425**

C. *Net Promoter Systems*

In late 2003 Frederick Reichheld originally proposed Net Promoter Harvard Business Review [6]. Since that introduction Net Promoter Score (NPS) became a popular client feedback system a company's offerings and performance.

Net Promoter measurement systems use survey programs that capture responses from the company's customers. Respondents are asked to estimate their likelihood of recommending the company, its products and/or its services to a friend or colleague [4]. The responses are given on a 10 or 11-point scale ('1'-'10' or '0'-'10'), with '10' being "extremely likely to recommend" and the lowest value being "extremely unlikely to recommend." The values are recoded into a 3-value ordered semantic scale (see Table 3) [4]:

Table 3 Net Promoter Value to Semantic Label Recode

Response Value	Semantic Label
'9'-'10':	Promoter
'7'-'8':	Passive
'0'-'6':	Detractor

The Net Promoter Score is the difference in percentage of respondents who are Detractors the percentage of respondents who are Promoters. This difference is multiplied by 100, to create a metric that has a scale of -100 to 100 [4]. Companies invest in a variety of customer touchpoints for their NPS measurement system [7]. Some companies are embedded the NPS system into all facets of their business, including internal services (e.g. employee helpdesks and HR employee touchpoints) [7].

The ability to assess likely Net Promoter Score in text, in social media (Twitter, Facebook, etc.), blogs (e.g., technical review sites), and customer surveys may provide multiple additional assessment touchpoints for the company. A text based NPS metric may form a basis for rating the company's competitor Net Promoter Scores. Companies like Uber, Facebook, and Twitter employ sophisticated sentiment analysis process to better understand customer attitudes [8]. For a company that uses Net Promoter Score as a core Key Performance Indicator (KPI), the ability to classify social media comments and survey responses without the need for costly survey based evaluation may open new areas of business analysis and measurement that are not currently available.

The survey database for the company that provided the NPS data for analysis has over 60,000 completed surveys with short responses that are linked to a respondents NPS score. The data includes responses from customers across the globe. In the cases where responses are provided in a language other than English, IBM Watson Language Translator was used to provide English versions of the response.

It should be noted that the data did not go through a screening process to validate the class labels, with the corresponding written statements. In some cases, respondents offer reasons as to why the rating was not a '10', so the response may appear to be negative or similar to negative comments that correspond to low rating values. In other cases, a very low rating may be provided may be paired with a positive response (e.g., the helpdesk agent was polite and worked hard to resolve a problem) which may be very similar (or even identical) to a response for a very high rating. In other cases, the respondent may include a very neutral comment (e.g. the listing of technical components/processes that resulted in a problem) and a promoter or detractor sentiment isn't clear. There is sufficient data for the algorithm to be able to discern patterns that are associated with each response class, in spite of these inconsistencies. As direct result, the pure accuracy does not reach that of well documented binary sentiment analyzers that can be found on-line.

While this is a test of verbatim responses of no more than 500 characters, other text applications may be quite large, so this test case used the OHPLall, with mini batches to assess algorithm performance on a text classification problem. An example application on a very large corpus might be the development of letter grade classifier predicting grade on a corpus of 1,000+ term papers that are each 25 pages in length. Assuming 300 words per page, a single document would have approximately 7,500 words per document (double spaced). If one of the larger word to vector embeddings, with vector length of 100 is used the size of a single document would be almost 100,000 values. While the data used for this application isn't this large, the text is a valid assessment of real data that is produced by real activities in businesses.

D. *Sentiment Analyzers*

Sentiment analysis of text, to determine the writer's positive or negative attitude in their communication is a widely used application of Natural Language Processing. Early efforts in sentiment analysis focus on binary Positive-Negative distinctions, using Bag of Words (set of most common words, across all records) that are one hot encoded (binary encoded). [9] This methodology has two meaningful limitations. The encoding of the bag of words doesn't include word sequences that may provide valuable "signal" for the analysis methodology. In addition, the encodings tend to be sparse matrices which can prose problems in developing predictive models. [9]

The uses of word embeddings, where words are represented by unique vectors provides a representation of the text, maintaining the word sequences within the text. The two-dimensional representation may be analyzed using a variety of methods. Convolutional Neural Networks (CNN's) and Recurrent Neural Networks (RNN's) have been demonstrated to provide excellent results. [9] While simple RNN's can be used to examine sequences of text, they may not be able to "extend" relationships that are more than a few words apart.

More recent work demonstrates that LSTM and Gated RNN's can over-come this potential issue. These specialized forms of RNN's include logic gates that are optimized to maintain useful information, "long-term", potentially over the entire sequence. In Gated RNN's one gate, commonly called the "Reset Gate" determines what information is brought into the node, from the prior node (hidden state). The other gate, commonly called the "Update Gate", determines what information is passed to the next node as a new hidden state. [10]

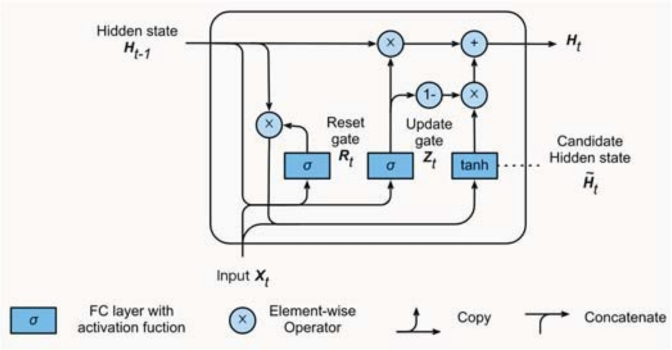


Fig 3 Gate Recurrent Neural Network Node
Image Source: https://www.d2l.ai/chapter_recurrent-neural-networks/gru.html#reset-gate-in-action. [10]

Many sentiment analyzers focus on a single binary outcome (positive-negative) or take the approach that used to analyze "multi-class" label data, where a label with N different classes is recode into an Nx1 vector of N-1 0's and a single value of 1. The first would require an over simplification of the NPS problem, while the second doesn't include the ordering information of the NPS labels.

A third approach uses "Ordinal Regression" which is essentially modification of the multi-class methodology, but instead of recoding the labels into a vector with a single value of 1, the labels are encoded into an N-1x1 vector. [11] The design compares grouped lower values to grouped higher values. For example, the lowest class, is compared to all others in the first element of the output vector. The lowest two classes grouped in a single class versus all others in the second element of the output vector and the process continues until the highest classes is predicted as separate from the lowest N-1 classes, in the last element of the vector.

The lowest value label would be encoded into a vector of all 0's. From that point, the next lowest label would have a 1 in the 1st position and zeros otherwise. The 3rd lowest would have 1's in the 1st two positions and 0's otherwise. This coding process continues until reaching the highest value which is encoded into a vector of all ones. Table 4 illustrates the encoding for a problem that has 4 ordinal classes of 1-4. [11]

Table 4 Ordinal Regression Label Encoding: 4 Class Case

Label	Vector
1	[0, 0, 0]
2	[1, 0, 0]
3	[1, 1, 0]
4	[1, 1, 1]

The neural network uses the sigmoid function to predict values between 0 and 1, in the output vector. From that point, a classification rule is applied. One common rule is using simple rounding to develop a vector of binary values. The resulting vector is assessed to determine how many values, starting with the 1st position, have a value of 1. That pattern is matched to the encoding to determine the class. Note that it is possible to have a predicted value with a 0 between two 1's. For those cases, the researcher must decide how to score the record. The simplest method is to essentially consider all values past the first 0 as though they are 0. Other strategies may be employed, but they bring concerns that the decision process may cause over fit of the classifier, since the record is an unusual case that violates the basic assumptions of the model.

To facilitate the use of word embeddings, large word to vector databases like Word2Vec [12] and GloVe [13] provide the ability to use pretrained embeddings. Doing so reduces model complexity. Additionally, these word embedding databases offer an opportunity to use pretrained initial weights. [9]

E. Net Promoter Score

Net Promoter was originally proposed by Frederick Reichheld, in late 2003. [6] In the intervening years, NPS has become a widely used management system, to assess overall client feedback, on a company's products and services. The metric has been demonstrated to have a strong association with future company revenues. [1]

Net Promoter measurement systems use survey responses from a company's customers, who are asked to estimate their likelihood of recommending the company, its products or its services to a friend or colleague. [4] The responses are given on a 10 or 11-point scale (1-10 or 0-10), with 10 being "extremely likely" and the lowest value being "extremely unlikely." The values are recoded into a 3-point semantic scale, as follows: [4]

Table 5 Net Promoter Value to Semantic LabelRecode

Response Value	Semantic Label
9-10:	Promoter
7-8:	Passive
0-6:	Detractor

To create a score, on a scale of -100 to 100, the percentage of respondents who are Detractors is subtracted from the percentage of respondents who are Promoters. [4] Companies use a variety of customer touchpoints for their measurement system. [7] Some companies are embedding NPS in all facets of their business. Not only are customers being surveyed, on overall company performance, but they are surveyed regarding specific product/service offering. In addition, process, internal to the company (e.g., helpdesks that employees use for workstation issues) are also measuring NPS. [7] The ability to access text, in social media (Twitter, Facebook, etc.), blogs (e.g., technical review sites) and customer surveys, provides multiple additional touchpoints for the company to assess.

III. Experimental Results

Survey data from a large IT company that included Net Promoter survey rating as well as verbatim text related to the Net Promoter response was available for over 60,000 surveys. This data represents a significant challenge as an ordinal classification problem. The ordinal labels haven't been vetted for accuracy, nor were they manually applied to the data, by researchers who read the text and assigned the class label. Survey respondents were allowed to respond as they wished [5].

The data went through a partial cleansing that was very minimal in scope. Almost 1,000 records were removed from the dataset due to the nature of the entered text. In some cases, the values were an url. In others, the respondent entered numbers only, question marks only or some other punctuation. Some single word text values were text values were also removed. In the largest case, over a dozen text responses were the word "on". Sample records for each response class are provided in Table 6 [5].

Table 6 Sample Response Records

Label	COMMENTS
0	the response time for a pmr is pessimistic
1	no technical support contact
2	the problem is not solved
3	they never called in the time they promised
4	the same error has already been fixed in older versions but the bug in the current version is still there the remedy took then still months
5	the functionality is promising but some of it is a bit limited to build end user experiences on top of
6	still waiting for ifix for upgrade from 7 6 1 5 to 7 7 x
7	there is room for improvement in the relationship with the ep usually the initiative to talk to ibm comes from us
8	works pretty well but needs to keep up with the latest specs
9	prompt and professional answer
10	satisfied with ibm support services attitude and technology

The Stanford GloVe word embedding database is used to provide initial weights for the embedding layer and the embedding weights are further trained to optimize model performance. A Gated RNN (GRNN) plus additional "standard" hidden layers complete the neural network architecture.

Table 7 Label Frequencies [5]

Response Class	Training Set Counts	Test Set Counts	Validation Set Counts	Percentage of Sample
0	1,544	193	193	3.2%
1	655	82	82	1.4%
2	868	109	109	1.8%
3	1,053	132	132	2.2%
4	767	96	96	1.6%
5	2,416	302	302	5.0%
6	1,820	227	227	3.7%
7	3,595	449	449	7.4%
8	7,596	950	950	15.7%
9	9,195	1,149	1,149	19.0%
10	18,964	2,371	2,371	39.1%

The data were split into Training, Validation and Test sets using 80:10:10 splits. As can be seen in Table 7, the dataset labels are highly unbalanced. Not only does the 10 class represent 39% of the data, the 9 and 10 classes combined represent 58% of the data. Since OHPLnet, uses the mean value of the DNN prediction as the hyperplane centroids. Weighting one class over the others isn't appropriate. As a result, an over sampling strategy is employed to address class imbalance[5].

From the training dataset, under-represented classes were over sample to have a balanced dataset. Since the NPS process recodes the 11-class labels into an ordered 3-class semantic scale, the oversampling was executed to result in a balance of the 3-class frequencies. Promoters make up 58% of the class labels with just over 28,000 records (see Table 8). The other two classes will be over sampled to have 28,159 records, in the training set. The original unbalance training set will be used to help assess whether or not the model is over trained on the oversampled training set [5].

Table 8 Three-Class Frequencies [5]

Response Class	Training Set Counts	Test Set Counts	Validation Set Counts	Percentage Of Sample
Detractor	9,123	1,141	1,141	19%
Passive	11,191	1,399	1,399	23%
Promoter	28,159	3,520	3,520	58%

The validation sample is used to determine the optimal GRNN architecture. Several dozen different GRNN architectures as well as architectures using CNN and LSTM were tested. The "winning" GRNN architecture has 128 output channels (values) from the Gated Recurrent Unit layer with three hidden layers of 64, 32 and 8 nodes each (see Fig 4 below) [5].

In assessing ordinal class labels, two standard methodologies are used. Instead of using traditional accuracy (proportion of records that are correctly classified), Ordinal Class problems use Mean Zero Error (MZE), which is related to accuracy. To calculate MZE, the number of misclassified records is divided by the total number of records. Accuracy is 1 – MZE.

The other key metric is Mean Absolute Error (MAE). In calculating MAE, the absolute differences between actual label value and the predicted labels are summed and divided by the number of records. For the NPS classification problem, MAE is arguably the more important metric when developing an 11 ordinal class predictor, since the values will be recoded into the three-point semantic scale.

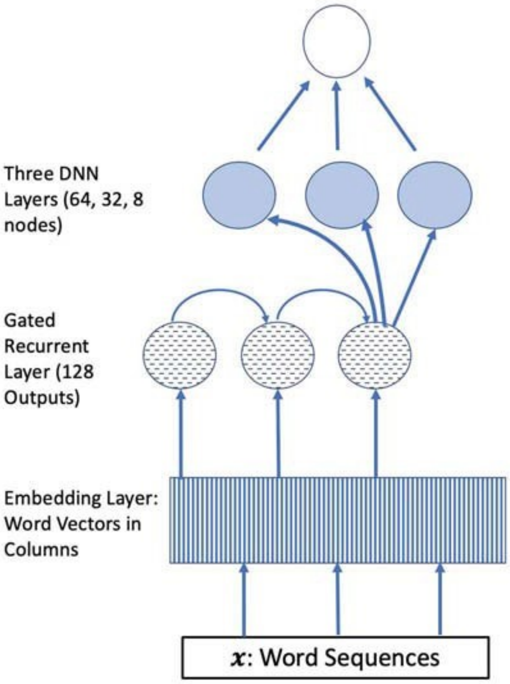

Three DNN Layers (64, 32, 8 nodes)
Gated Recurrent Layer (128 Outputs)
Embedding Layer: Word Vectors in Columns
$\boldsymbol{x}$: Word Sequences

Fig 4. Gate Recurrent Neural Network [5]

Ordinal Regression is used to provide an appropriate benchmark. The methodology can take advantage of the GloVe word embeddings as well as the GRNN architecture. Similar to the application of OHPLnet, the validation sample is used to identify a "winning" neural network architecture. Two similar GRNN's using Ordinal regression performed comparably. One used the same network architecture as the best OHPLnet and was chosen to serve as the performance benchmark [5].

Twenty executions of the algorithms were done, to create 20 models each. While, the performance differences are huge, the mean values for OHPLnet model MZEs, for the 11-class model were lower than for the Ordinal Regression based models. This advantage carried over in the recoding of the predictions into the three-label recoding. The advantage in MAE, for the 11-class solution lead to even better MZE performance, in the three-class recode [5].

Many examples of binary classifiers, that can be found on line, achieve 80% or higher accuracy rates, corresponding to a 0.20 MZE or lower. [9] [14] Low accuracy rates for both methodologies is in part due to the inconsistencies in rating versus the content of verbatim responses. Table 13 provides a sampling of examples where the respondent's rating isn't consistent with his/her verbatim response. In the example cases, the verbatim comments would appear to be inconsistent with the rating, while the predictions are consistent. While these cases are explicitly selected to illustrate the potential challenges of developing semantic analyzers based on survey data, they suggest that even with a less desirable accuracy and higher mean absolute error than desired, the classifier may provide a good basis to enhance the company's NPS program and associate KPI metrics [5].

Table 9 NPS Sentiment Analyzer Results For 20 Iterations of Each Algorithm [5]

		3-Class MZE	3-Class MAE	11-Class MZE	11-Class MAE
OHPLall	Mean	**0.320**	**0.370**	**0.652**	**1.281**
	Std Dev	**0.006**	0.007	0.014	0.032
Ordinal Regress-ion	Mean	0.360	0.406	0.724	1.352
	Std Dev	0.007	0.007	**0.010**	**0.011**

.

From a manual assessment of the misclassified cases in the test sample, in 46% of the records the NPS rating isn't consistent with text, on the three-point scale (see Table 13 at the end of this section) Assessing the subtle differences that would result different values in the eleven-point scale would be next to impossible. For a chosen model, if the inconsistent records are removed, the MZE and MAE values are reduced by more than 1/3 (see Table 10). While these kinds of inconsistent records are likely to continue in any survey process, their removal for this evaluation provides a better evaluation of the classifier that was built using OHPLnet. An accuracy rate above 80% for the classifier is comparable to binary classifiers that are reported in published papers. [14]

Table 10 Change in Accuracy After Removing Inconsistent Records

Metric	Full Test Set	Inconsistent Records Removed
MZE	0.311	0.195
MAE	0.362	0.211
Counts	6,060	5,187

The inconsistent records tend to be positive responses for scores below '9'. As a direct consequence, the models tend to score more values a "Promoter" than the actual number of responses that would be classified as "Promoter." If these inconsistent records are removed, the prediction counts by three-point semantic class are much more closely aligned.

In practice, companies that employ Net Promoter systems create a single metric that is used as a key part of their executive management system. The metric, called the Net Promoter Score (NPS) is calculated as:

$$NPS = 100\ (\%\ Positive - \%\ Negative)\ (6)$$

Table 11 Test Set Counts By 3-Point Semantic Scale

Label	Full Set Actual	Full Set Predicted	Inconsistent Records Removed Actual	Inconsistent Records Removed Predicted
Detractor	1,141	1,105	877	953
Passive	1,399	960	927	787
Promoter	**3,520**	**3,995**	3,383	3,447
Total	6,060	6,060	5,187	5,187

Table 12 NPS Actual versus Predicted

Label	Full Set Actual	Full Set Predicted	Inconsistent Records Removed Actual	Inconsistent Records Removed Predicted
Detractor	18.8%	18.2%	16.9%	18.4%
Promoter	58.1%	65.9%	65.2%	66.5%
NPS	39.3	47.7	48.3	48.1

The inconsistent records result in a skewing of the NPS score that's calculated from the predictions to be higher than the actual score. As a result, if an NPS Sentiment Analyzer becomes a part of a company's Net Promoter system, they will need to choose to track the impact of text inconsistencies, in their classifier training data. From that tracking, the scores can be adjusted to offset the effect or they can simply choose to use the scores as they occur, while retaining the knowledge that the sentiment analyzer based scores are inflated (in this case by 8.4 points)

IV. Conclusions

From the initial research that resulted in Ordinal Hyperplane Loss, the work represents a meaningful improvement in developing machine learning algorithms that attempt to produce classifiers for Ordinal Label problems. Since that time Ordinal Hyperplane Loss evolved into OHPLnet. The latest variant performs very well on a challenging text classification problem. The resulting classifier provides the company with a viable methodology to expand its Net Promoter Score program to include verbatim text that occurs in a multitude of sources.

Future work will focus on additional testing of the algorithm, with possible additional improvements. The OHPL algorithm will also be fully developed into a loss function that can be included as an available option in packages like TensorFlow, Keras and Pytorch.

Table 13 Inconsistencies Between NPS Responses and Verbatim Comments

Survey Response 3-Class Labels	Verbatim Comments
Detractor	1 experience 2 the support was fantastic
Detractor	because it was very carefully supported
Detractor	because of the quick response
Detractor	interface and graphics capabilities
Detractor	because we could respond promptly and as expected
Detractor	after calling we quickly arranged replacement parts and technical personnel it was very helpful to solve problems in a few hours
Detractor	good service
Detractor	quick response and accurate answer
Detractor	the positive experience prevails
Detractor	competent friendly patient
Promotor	1 very long and complex bureaucratic procedures 2 long lead times for orders
Promotor	bass guitar
Promotor	because we cannot access the system without our pcomm in our pc os environment
Promotor	because the printing function of acs is not stable when it comes to printing it becomes pcomm which is the way to recommend it
Promotor	this pmr has been very long and has already had a predecessor pmr with the same problem which could not be solved at the time
Promotor	IBM's price competitiveness is still weaker
Promotor	time did not change the quality of the system ie of its granite operating system
Promotor	vacations at grundfos and at [Company] prolonged the handling time
Promotor	the solution was not satisfactory
Promotor	no good communication in this case

V. References

[1] T. L. Keiningham, B. Cooil, T. W. Andreassen and L. Aksoy, "A Longitudinal Examination of Net Promoter and Firm Revenue Growth," *Journal of Marketing,* p. 39–51, 2007.

[2] P. Gutiérrez, M. Pérez-Ortiz and J. Sánchez-Mone, "Ordinal regression methods: survey and experimental study," *IEEE Trans. Knowl. Data Eng. 28,* no. 1, p. 127–146, 2016.

[3] B. Vanderheyden and Y. Xie, "Ordinal Hyperplane Loss," in *2018 IEEE International Conference on Big Data*, Seattle, WA, 2018.

[4] Medallia Corporation, "Net Promoter Score®," Medallia Corporation, 2019. [Online]. Available: https://www.medallia.com/net-promoter-score/. [Accessed 11 March 2019].

[5] B. Vanderheyden, "Ordinal HyperPlane Loss," 1 Nov 2019. [Online]. Available: https://digitalcommons.kennesaw.edu/dataphd_etd/4/. [Accessed 1 Nov 2019].

[6] F. F. Reichheld, "The One Number You Need to Grow," *Harvard Business Review,* vol. December 2003, 2003.

[7] Medallia, Inc, "Medallia Recognizes World's Most Innovative Customer Experience Leaders," Medallia, Inc, 16 May 2018. [Online]. Available: https://www.medallia.com/press-release/medallia-recognizes-worlds-most-innovative-customer-experience-leaders/. [Accessed 11 July 2019].

[8] S. Gupta, "Sentiment Analysis: Concept, Analysis and Applications," Towards Data Science, 7 January 2018. [Online]. Available: https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17. [Accessed 5 July 2019].

[9] X. Wang, Y. Liu, C. Sun, B. Wang and X. Wang, "Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory," in *ACL*, 2015.

[10] A. Zhang, Z. C. Lipton, M. Li and A. J. • Smola, "Dive into Deep Learning: Chapter 8.8. Gated Recurrent Units (GRU)," 2019. [Online]. Available: https://www.d2l.ai/chapter_recurrent-neural-networks/gru.html#reset-gate-in-action. [Accessed 5 July 2019].

[11] J. Cheng, "A Neural Network Approach to Ordinal Regression," 2007. [Online]. Available: http://arxiv.org/abs/0704.1028. [Accessed 5 July 2019].

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Proceeding of Neural Information Processing Systems (NIPS)*, Lake Tahoe, 2013.

[13] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014.

[14] . Tarımer, A. Çoban and K. A. Emre, "Sentiment Analysis on IMDB Movie Comments and Twitter Data by Machine Learning and Vector Space Techniques," 2019. [Online]. Available: http://arxiv.org/abs/1903.11983. [Accessed 5 July 2019].

[15] B. Carremans, "Word embeddings for sentiment analysis," Towards Data Science , 27 August 2018. [Online]. Available: https://towardsdatascience.com/word-embeddings-for-sentiment-analysis-65f42ea5d26e. [Accessed 5 July 2019].