[redacted]

[redacted]

Sentiment Quantification

Andrea Esuli and Fabrizio Sebastiani, *Italian National Council of Research*

Opinion mining has come to play a key role in text mining applications for customer relationship management, consumer attitude detection, brand and product positioning, and market research. Interest in these applications has spawned a new generation of companies and products devoted to online reputation management, market perception, and online content monitoring.

Historically, one of the most important incarnations of opinion mining has been sentiment classification, the task of classifying a given piece of natural language text (be it a short remark, blog post, or full-blown product review) not according to its topic (as in standard text classification) but according to the opinions expressed in it. One interesting instance of sentiment classification is detecting whether a given product review is positive or negative, which is an example of *binary* classification. More subtly, it might be interesting to detect *how* positive or negative the review is; if we express the possible values on a finite scale of integers, such as between one (very negative) and five (very positive), this is an example of *ordinal* classification (also known as "ordinal regression").

Sentiment classification is pervasive in all contexts where opinions must be mined from large quantities of text. For instance, in a typical customer relationship management application, a company might ask customers to fill out a questionnaire to determine their opinions on a product or service they recently purchased. If the questionnaire contains open questions, the company will need to bin the textual answers into classes that represent different types of opinions. For example, an online bank that polls its customers on how satisfied they are with their online account might use classes such as "satisfied overall," "unhappy with website navigation," "customer ready to churn," and so forth. When the large amount of questionnaires received makes manual processing too expensive or simply infeasible given the time constraints, automatically classifying respondents becomes the only available option. Because the "opinion" dimension is of key importance to this classification endeavor, the technology used must combine sentiment analysis techniques and (more traditional) text classification techniques based on supervised learning.[1]

1541-1672/10/$26.00 © 2010 IEEE
Published by the IEEE Computer Society

Sentiment classification of textual answers returned within questionnaires could serve other purposes as well. Other applications might include survey coding for the social or political sciences (such as when open questions inquire about the respondents' beliefs, social status, or political leanings)[2] or market research (such as when open questions deal with the respondents' perception of products, brands, or advertising campaigns).

Another important sentiment classification application is managing online product reviews. Such reviews are available across numerous specialized websites (Amazon, Epinions.com, Ratingz.net, and TripAdvisor.com are only a few examples) and increasingly influence consumers' product-purchasing decisions. While structured reviews from such websites consist of a textual product evaluation and a score expressed on an ordered scale of values, many others (such as those to be found in newsgroups, blogs, and other venues for spontaneous discussion) contain only a textual evaluation, with no score attached. These latter reviews are difficult for an automated system to manage, especially when we need to determine, based on the reviews alone, the best perceived product in the lot or whether product x is considered better than product y.

Tools capable of interpreting a text-only product review and classifying it according to how positive it is are thus of the utmost importance. Such a tool would "star-rate" a product review—that is, assign it a certain number of "stars" (from one to five) based on its textual content. Additionally, it could compute the average star-rating obtained by a given product (as resulting from the product reviews written by different consumers) and rank all the products in a given range (for example, all horror movies released between 2006 and 2008 and produced in the US) according to their computed average star-rating.

Individual or Aggregate?

The opinion mining community has traditionally neglected whether the analysis of these large quantities of text should be carried out at the individual or aggregate level. This is an important issue because some of the applications we have discussed so far (namely, open-answer classification for customer satisfaction analysis) demand attention at the individual level, while others (such as open-answer classification for market research or review classification for product or brand positioning) are best analyzed at the aggregate level.

When classifying thousands of questionnaires according to whether the respondent belongs to the class "customer ready to churn," a telecom company is likely interested in accurately classifying each individual customer because it might want to contact them individually to offer improved conditions. Conversely, in a market research application in which the questionnaire asks about the respondent's perception of a given ad campaign, the company is likely not interested in whether a specific individual belongs to the class "liked the campaign," but rather it wants to know the percentage of respondents that belong to the class. Similarly, given a large set of star-rated reviews of a given MP3 player, we are interested in knowing the statistical distribution of the answers across the possible star-ratings, and we are not interested in individual ratings. These examples demonstrate that not all these contexts are alike in terms of the granularity at which the results are to be analyzed. Some applications (ideally) demand that every single item be correctly classified, while others instead (ideally) demand that the true percentage of items that belong to the class be correctly *quantified*. Although in most applications of classification by topic the individual level of analysis seems the more (if not the only) appropriate one, the aggregate level of analysis features prominently in sentiment classification applications. We thus argue for a new focus shift within the opinion-mining community, from sentiment classification to *sentiment quantification*, a shift that recognizes the two as distinct application needs, each requiring specific tools in order to be addressed optimally.

Obviously, classification is a more difficult task than quantification. In fact, the ideal classifier is by definition also an ideal quantifier, but an ideal quantifier is not necessarily an ideal classifier. In fact, to perfectly estimate the percentage of items that belong to the class, a classifier must only deliver an equal number of false positives and false negatives since the two compensate each other when quantifying class frequencies.

Interestingly, George Forman noted only recently (although not in the context of sentiment classification) that the results of classification sometimes need to be analyzed purely at the aggregate level.[3] The history of classification is thus a history of analysis at the individual level.

Evaluating Sentiment Quantification ...

Which mathematical measure should we use to evaluate quantification accuracy? Quite reasonably, for the case of binary classification, Forman proposed the use of normalized cross entropy,[3] better known as

Kullback-Leibler Divergence (KLD) and defined as

$$KLD(p,q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

KLD is a measure of the error made in estimating a true distribution p by means of a predicted distribution q. Thus, KLD is in principle suitable to our needs because quantifying exactly means predicting how the test items are distributed across the classes.

It might seem that optimizing classification *a fortiori* means optimizing quantification. In other words, on the surface it would seem obvious that the more we improve a classifier's accuracy at the individual level, the higher its accuracy at the aggregate level will become, and that the only way to improve a classifier's ability to correctly estimate the distribution of test cases across classes is to improve its ability to classify individual items. Unfortunately, we contend this is not true, or at least that this depends on what we mean by "accuracy at the individual level." To see this, we need to look at the definition of F_1, the standard evaluation function for binary classification, which is defined as

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (1)$$

where TP, FP, and FN indicate the numbers of true positives, false positives, and false negatives, respectively, from a standard contingency table. Equation 1 shows that F_1 deteriorates with $(FP + FN)$ and not with $|FP - FN|$, as would instead be required of a function that truly optimizes quantification. For example, according to F_1, a classifier $\hat{\Phi}_1$ for which $FP = 50$ and $FN = 50$ is worse (all other things being equal) than a classifier $\hat{\Phi}_2$ for which $FP = 0$ and $FN = 10$. However, $\hat{\Phi}_1$ is better than $\hat{\Phi}_2$ according to KLD, and according to any reasonable measure for evaluating quantification accuracy. Indeed, $\hat{\Phi}_1$ is a perfect quantifier since FP and FN are equal and thus compensate each other, so that the distribution of the test items is estimated perfectly.

The situation is the same for ordinal classification, the task we need to solve for star-rating product reviews. The standard evaluation measure for ordinal classification is mean absolute error (MAE), which is the numerical distance between the item's true and predicted classes, averaged across the test items. For instance, assigning two stars to a review that is really worth five stars incurs in an absolute error of three. MAE is obviously not a good measure for ordinal quantification. In fact, an ordinal classifier that has classified all test items correctly aside from swapping equal numbers of items between two classes c_i and c_j, has perfectly estimated the distribution of items across the ordered classes, regardless of the number of swapped items and of the "distance" between c_i and c_j. Examples analogous in spirit to the previous one can show that an ordinal classifier $\hat{\Phi}_1$ might be better than another ordinal classifier $\hat{\Phi}_2$ in terms of MAE but would be worse than $\hat{\Phi}_2$ in terms of any reasonable evaluation function for ordinal quantification.

Therefore, which functions should be used to evaluate ordinal quantification? To the best of our knowledge, we know of no measure that has been proposed for this task. To this purpose, in our ongoing research we are adopting the Earth Mover's Distance (EMD),[4] a function often used in content-based image retrieval for computing the distance between two images' color histograms. EMD computes the minimal cost incurred in turning one distribution into the other, where the cost is computed as the probability mass that must be moved from one class to another, weighted by the distance between the two classes.

... and Optimizing It

The examples of the previous section demonstrate that simply improving classification accuracy is not the optimal way of improving quantification accuracy. This not only indicates that classification and quantification are two different, albeit related tasks, it also indicates that quantification should be tackled according to methods different from the ones that prove optimal for classification.

Concerning this, Forman proposed several learning methods explicitly devised for binary quantification and experimentally showed that they improve quantification accuracy with respect to standard methods originally devised with just (individual) classification in mind.[3] However, none of these methods are based on explicitly optimizing the function eventually used in evaluating quantification. We are currently pursuing this line of research in our ongoing work. In particular, the idea is that of adopting the SVM_{multi} approach,[5] which consists of using a learning device based on support vector machines (SVMs) that lets us optimize any nonlinear evaluation function that can be directly computed from a contingency table, such as KLD. The approach is fundamentally different from conventional learning algorithms: instead of generating a binary classifier that classifies individual test instances one at a time, SVM_{multi} generates a classifier that conceptually classifies an entire set of test instances in one shot. By doing so, SVM_{multi} can optimize properties of entire sets of instances

that, as KLD, are not linear functions of individual instances.

We hope to report the results of experimenting with this approach on sentiment quantification data sets in the near future. Concerning the optimization of *ordinal* quantification, instead, further research is still needed to devise ordinal regression methods that can explicitly optimize EMD.

References

1. T. Macer, M. Pearson, and F. Sebastiani, "Cracking the Code: What Customers Say, in their own Words," *Proc. 50th Ann. Conf. Market Research Soc.* (MRS 07), MRS, 2007.
2. D. Giorgetti and F. Sebastiani, "Automating Survey Coding by Multiclass Text Categorization Techniques," *J. Am. Soc. Information Science and Technology*, vol. 54, no. 14, 2003, pp. 1269–1277.
3. G. Forman, "Quantifying Counts and Costs via Classification," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, 2008, pp. 164–206.
4. Y. Rubner, C. Tomasi, and L.J. Guibas, "A Metric for Distributions with Applications to Image Databases," *Proc. 6th Int'l Conf. Vision* (ICCV 98), IEEE CS Press, 1998, pp. 59–66.
5. T. Joachims, "A Support Vector Method for Multivariate Performance Measures," *Proc. 22nd Int'l Conf. Machine Learning* (ICML 05), ACM Press, 2005, pp. 377–384.

Andrea Esuli is a researcher at ISTI-CNR. He has a PhD in information engineering from the University of Pisa, Italy. Contact him at andrea.esuli@isti.cnr.it.

Fabrizio Sebastiani is a senior researcher at ISTI-CNR. He has a "Laurea" degree in computer science from the University of Pisa, Italy. Contact him at fabrizio.sebastiani@isti.cnr.it.