Electoral Predictions with Twitter: a Machine-Learning approach

M. Coletto[1,3], C. Lucchese[1], S. Orlando[2], and R. Perego[1]

[1] ISTI-CNR, Pisa
[2] University Ca' Foscari of Venice
[3] IMT Institute for Advanced Studies, Lucca

Abstract. Several studies have shown how to approximately predict public opinion, such as in political elections, by analyzing user activities in blogging platforms and on-line social networks. The task is challenging for several reasons. Sample bias and automatic understanding of textual content are two of several non trivial issues.
In this work we study how Twitter can provide some interesting insights concerning the primary elections of an Italian political party. State-of-the-art approaches rely on indicators based on tweet and user volumes, often including sentiment analysis. We investigate how to exploit and improve those indicators in order to reduce the bias of the Twitter users sample. We propose novel indicators and a novel content-based method. Furthermore, we study how a machine learning approach can learn correction factors for those indicators. Experimental results on Twitter data support the validity of the proposed methods and their improvement over the state of the art.

Introduction and Related Work

The use of the Twitter micro-blogging platform as a tool to predict the outcomes of social phenomena is a recurrent task in the recent social network analysis literature. Successful studies can be found in different contexts using Twitter for predictive tasks: from prediction of stock market [4] to movie sales [1], and pandemics detection [12]. Indeed, social systems can be studied through sophisticated models being validated through online social networks and blogging platforms, since these new digital contexts provide large scale data sets including millions of users. *Computational Social Science* is becoming a leading research area in understanding communication patterns and social behaviors, in tracking tastes and in predicting opinions [13]. A relevant context which received a lot of attention is the prediction of elections and opinions on political events and decisions. Many articles propose quantitative approaches to predict the electoral results in different countries: US [15], Germany [19], Holland [16], Italy [5]. In particular, we distinguish two classes of methods used in literature: volume-based approaches and content-based approaches.

The first class refers to metrics consisting in counting tweets, users, mentions for a given candidate or a political party. [19] shows that volumes of mentions of parties reflects the distribution of votes in the election among six parties in 2009 German elections. Similar results were achieved by other studies [16, 3]. Counting users, instead of

tweets, is effective as we can consider each user to be a single elector [6]. Similar approaches were applied to Facebook data as well [10, 20]. For instance, the number of Facebook supporters can be used as an indicator of electoral success. Other works highlight some concerns about using tweet volumes to predict elections [17, 14, 9], showing how in practical cases these approaches may under-perform the baseline. For instance, in [11] it is shown that some arbitrary choices (e.g., the set of considered parties, the time frame, etc.) strongly affect the results, exhibiting a not consistent predictive behavior.

The second class of methods aims at exploiting text information in tweets, and most approaches are based on *sentiment analysis* [3]. In this context sentiment indicates the degree of agreement expressed in a tweet in relation to a political party or candidate. A few studies applied a machine learning approach to classify tweets according to their polarity, either by training on a manually annotated sample [16, 3] or through dictionary-based unsupervised methods [2]. Sentiment analysis methods have been used to improve the predictive results of counting methods, but they still are an open research challenge due for instance to the not trivial identification of sarcasm and irony.

Results of both approaches seem not to be consistent across datasets [9]. Predictions vary significantly in relation to the observation period, the data collection and cleansing methods, and the performance evaluation strategy. In fact, all predictive studies have been performed after the outcomes, thus evaluating correlations but not prediction power [14], and scientific papers are mostly biased towards positive results and they do not report negative ones [7]. Finally, the predictive power of Twitter is very sensitive to the bias of its users, as Twitter users are not a representative sample of users involved in the elections, neither of people in general. In particular, [18] discusses this issue, stating that demographic groups can have different political opinions not equally detectable from new social media. [16, 8] proposed some de-biasing strategies.

In this work we adopt as baselines the approach used in [19, 16], i.e., counting the mentions of the political candidates in the election, and the one used in [6], i.e., counting unique users mentioning a candidate. We analyze a data set of tweets related to the 2013 primary elections of the major Italian political party. The data set is partitioned on the basis of the twenty Italian regions from which the tweets were posted; since we know the electoral results per each region, we can study them as independent election events.

First, we evaluate and discuss state-of-the-art methods based on tweet and user volumes. We, then, propose several new predictors that exploit some enhanced classifications of tweets based on hash-tags. We show that, by properly classifying tweets, it is possible to reduce the error of baseline methods by a factor of 25%.

We also address the bias issue. We propose to learn the degree of bias of each candidate using external polls on expected demographic distribution of voters, so that the prediction can be adjusted accordingly. It turned out that our data set is biased mainly towards young people between 25 and 44 years old and we show that by learning the Twitter bias degree, the electoral ranking outcome can be correctly predicted in 75% of the Italian regions. We conclude that machine learning approaches can be exploited successfully to learn correcting factors for the prediction if training data is available.

Data

In this work we investigate the echo on Twitter of the primary elections of the Italian major political party: the "Partito Democratico". Our study is conducted on a data set of ≈1.7 million tweets. The election took place on December 8th 2013, and the data-set covers about 10 days before and 5 days after the election day. We considered only the geo-located tweets in Italian. In Fig. 1 we report a chart with the daily volumes of collected geo-located tweets.

Political context

The "Partito Democratico" is the greatest social-democratic political party in Italy. Three candidate were selected to run for the primary election that took place on December 8th 2013: Mr. Renzi, Mr. Cuperlo, and Mr. Civati. They appeared in the traditional media (TV shows and Press interviews), and they also invested a lot of effort on social media, including Twitter, in order to create hype and discussions. The candidates received 67.55%, 18.21% and 14.24% of votes, respectively. This result is difficult to predict if we simply base the prediction on Twitter data volumes, because, as shown in the following sections, the presence of Mr. Cuperlo is quite limited compared to the other two candidates. This makes this data set very challenging. Note that Mr. Renzi and Mr. Civati were leading emerging and younger factions in the party.

Data collection and cleansing

The data used in the case study was collected through Twitter API by querying a list of keywords related to the elections and the candidates [4]. The selection of keywords and hash-tags was large enough to guarantee a good coverage of the elections[5].

Data cleansing is a core activity to analyze reliable data. Our initial dataset contained about ≈1.7 million tweets. We deleted partial data and irrelevant tweets provided by Twitter APIs. We selected the Italian tweets on the basis of the language declared by Twitter users and the language detected by a machine learning classifier by Twitter. Only about 8 thousand tweets provided GPS information, whereas the remaining tweets were geo-located by matching the user profile location with the Italian cities and regions. We finally filtered 95,627 geo-located tweets across the 20 regions of the country, taking into consideration only the tweets published before the election day. The final data set size (≈95 thousand) is comparable with the data sets used in literature, in particular, considering our baseline approaches: namely [19] where the authors analyzed about 104 thousand tweets covering one month preceding the German elections

[4] Data were collected by Michelangelo Puliga, IMT for Advanced Studies. We thank IMT and LinkaLab for the courtesy.

[5] The list of users (through mentions), hash-tags and keywords tracked is the following: *matteorenzi*, *cuperlo*, *civati*, *giannicuperlo*, *vvattuone*, *giannipittella*, *pippocivati*, *giuseppecivati*, *renzi*, *primarie pd, partito democratrico, primariepd, iovotoperché, pd, matteorisponde, congressopd, PrimariePD2013, cambiaverso, pdnetwork, ilconfrontopd,iostoconcivati, ciwati, segretario, pittella, insultacivati, d'alema, massimoleaderpd, dalema, giuseppecivati*.

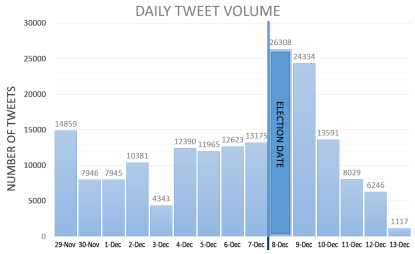

Fig. 1. Temporal distribution of tweet volume

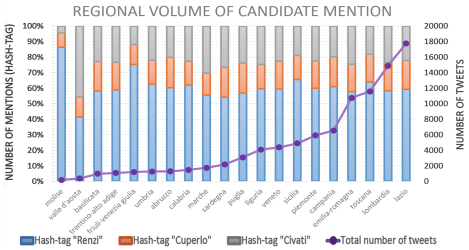

Fig. 2. Regional volume of candidate mention

in 2009, and [6] where the authors compared different predictive approaches on a data set of about 114 thousand tweets, covering the three months before U.S. congressional election of 2010. The time window considered in our work is limited to only 10 days before the elections, in line with other works which consider a short time range before the election date being more relevant for predictive tasks, for instance [16] (1 week). Fig. 2 shows the amount of data collected per region before the election date, and the percentage of mentions of each candidate. Fig. 3 shows the distribution of hash-tag occurrences of candidate names over time, before and after the election date. We investigated those users with the highest posting rate to remove anomalous users. From our evaluation, even the most active users (more than 1 thousand tweets, written in the 10 days before the election) posted meaningful tweets, different from one another, indicating a human behavior. Surprisingly more active users turned out to be individual supporters or local organized groups, not newspapers or official institutional pages.

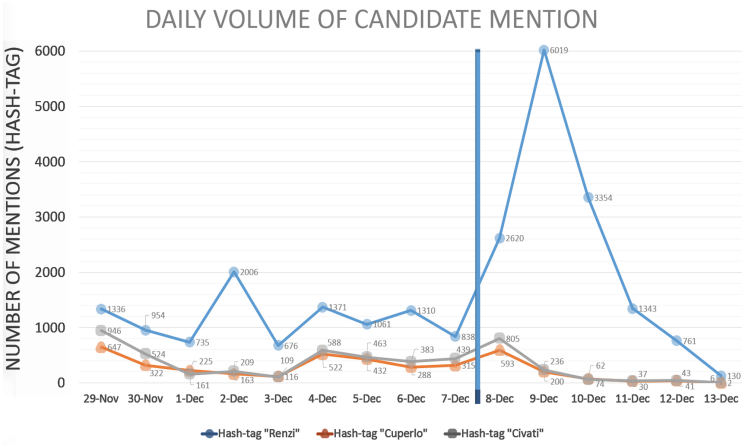

Fig. 3. Daily volume of candidate mention

Methods and algorithms

In the following we evaluate several *estimators*, or *predictors*. A predictor ϕ produces an estimate $\phi(c)$ of the share of votes that the candidate c will receive. Each predictor is normalized over the set of candidates $\mathcal{C}$ before the evaluation. The normalized version $\bar{\phi}(c)$ is defined as:

$$\bar{\phi}(c) = \frac{\phi(c)}{\sum_{c' \in \mathcal{C}} \phi(c')}$$

We used three different evaluation measures to assess the approaches discussed in this work. The most commonly used evaluation measure is the Mean Absolute Error (MAE). We also report the Root Mean Squared Error (RMSE), as it is more sensitive to large estimation errors. Finally, since we are also interested in the capability of predicting the correct ranking of the candidates, we also introduced the Mean Rank Match (MRM) measure, i.e., the mean number of times that the correct ranking of all the candidates was produced. Note that we conducted a per-region analysis, meaning that a prediction is produced for every region by exploiting the regional data only. The presented results are averaged across the 20 Italian regions.

Baseline methods

A basic approach is described in [19]. They estimated the share of votes of a political party as the share of tweets mentioning it. Let $\mathcal{T}$ be set of tweets in the observed period, and let $\mathcal{C}$ be the set of parties, the popularity $f(c)$ of a party is defined as:

$$\phi(c) = f(c) = |\{t \in \mathcal{T} \mid c \in t\}|$$

where $c \in t$ holds *iff* the tweet t mentions the party c (in our case study we consider different candidates in a primary election, which are assimilated to parties running in a political election). Understanding whether a tweet discusses a given political party may not be straightforward. In [19], a tweet is considered to mention a given political party if its text contains the party acronym or th name of selected politicians of the party. This simple estimator achieved a MAE of 1.65% and it was able to predict the correct ranking of the elections. Authors concluded that $f(c)$ can be used as a plausible estimation of vote shares, and they show that this estimator is very close to traditional election polls.

Users count, instead of tweet counts, were considered in [6]. Let $\mathcal{U}$ be the set of twitter users, the popularity $u(c)$ of a party is defined as the number of users mentioning c at least once in the observed period:

$$\phi(c) = u(c) = |\{u \in \mathcal{U} \mid \exists t_u \in \mathcal{T} \wedge c \in t_u\}|$$

where t_u denotes a tweet t authored by user u. The $u(c)$ predictor showed to be only marginally better. We named the above two methods TweetCount and UserCount respectively.

In our analysis, we considered a tweet to mention a candidate if it contains a hashtag with his family name, i.e., `#renzi`, `#cuperlo` or `#civati`. The performance

Table 1. Baseline methods performance.

Algorithm	MAE	RMSE	MRM
TweetCount	0.0818	0.1024	0.35
UserCount	0.0940	0.1080	**0.45**

Table 2. Classification methods performance.

Algorithm	MAE	RMSE	MRM
UserShare	0.0616	0.0792	0.35
ClassTweetCount$_{\mathcal{H}}$	0.1056	0.1248	0.30
ClassUserCount$_{\mathcal{H}}$	0.0924	0.1090	0.30
ClassTweetCount$_{\mathcal{C}}$	**0.0636**	**0.0786**	0.34
ClassUserCount$_{\mathcal{C}}$	0.0804	0.1033	0.38

measures on our data set are reported in Table 1. The performance of the first two methods are very close both in terms of MAE and RMSE. We can observe some improvement in terms of MRM, suggesting the focusing on Twitter users as estimators of the behavior of voters is a valuable approach. Considering the full text instead of hash-tags with these predictors did not provide any significant benefit, and therefore results are not reported here. We exploit the full text in some content-based predictor presented later.

Exploiting tweet/user classification

We first propose an improvement over the UserCount strategy. According to UserCount, the relation according to which one Twitter user corresponds to one voter is not satisfied as users mentioning more than one candidate are taken into consideration multiple times. We correct this behavior with a normalization by the number of candidates mentioned. We say that a user $u \in \mathcal{U}$ is likely to vote for candidate $c \in \mathcal{C}$ with probability $P(c|u)$, defined as:

$$P(c|u) = \frac{\mathbb{I}\{\exists t_u \in \mathcal{T} \wedge c \in t_u\}}{|\{c' \in \mathcal{C} | \exists t_u \in \mathcal{T} \wedge c' \in t_u\}|}$$

where $\mathbb{I}\{x\}$ is equal to 1 if x is true and 0 otherwise. Clearly, $\forall u \in \mathcal{U}, \sum_{c \in \mathcal{C}} P(c|u) = 1$. We thus estimate the number of users likely to vote candidate c as:

$$\mathsf{UserShare}(c) = \sum_{u \in \mathcal{U}} P(c|u)$$

In the following we propose some enhanced classification of tweets polarity for the candidates. We try to evaluate what is the probability that mentioning a hash-tag h leads to a vote for a given candidate c. We introduce an approximation here, with the usual assumption that mentioning a candidate is equivalent to voting a candidate. Then, we can easily estimate $P(c|h)$ as follows:

$$P(c|h) = \frac{P(c,h)}{P(h)} = \frac{|\{t' \in \mathcal{T} | c \in t' \wedge h \in t'\}|}{|\{t' \in \mathcal{T} | h \in t'\}|}$$

This has the effect of smoothing the impact of very frequent hash-tags which are likely to occur frequently with every candidate mention, thus not providing any significant signal. By focusing on the subset of the 100 most frequent hash-tags $\mathcal{H}$, each tweet $t \in \mathcal{T}$ is associated with a candidate $c \in \mathcal{C}$ according to the score:

$$S_{\mathcal{H}}(c|t) = \sum_{h \in t \cap \mathcal{H}} P(c|h)$$

According to $S_{\mathcal{H}}(c|t)$ every hash-tag in t may contribute to strengthen the relation with a given candidate $c \in \mathcal{C}$. We can now use $S_{\mathcal{H}}(c|t)$ to *label* a tweet with a candidate. We say that t is labeled with c, or equivalently $\lambda_{\mathcal{H}}(t) = c$, if $c = \arg\max_{c' \in \mathcal{C}} S_{\mathcal{H}}(c'|t)$. Whenever $\lambda_{\mathcal{H}}(t)$ is non uniquely defined, i.e., multiple candidates have the same score, t is assigned to c with probability $\bar{f}(c)$, where $\bar{f}(c)$ is the normalized tweet count. We finally introduce a new indicator measuring the count of tweets labeled with a given candidate:

$$\mathsf{ClassTweetCount}_{\mathcal{H}}(c) = |\{t \in \mathcal{T} \mid c = \lambda_{\mathcal{H}}(t)\}|$$

This indicator is extended to consider users rather than tweets. We say that u is labeled with c, or equivalently $\lambda_{\mathcal{H}}(u) = c$, if $c = \arg\max_{c' \in \mathcal{C}} |\{t_u \in \mathcal{T} | c' = \lambda_{\mathcal{H}}(t_u)\}|$. Whenever $\lambda_{\mathcal{H}}(u)$ is non uniquely defined, i.e., multiple candidates have the same score, u is assigned to c with probability $\bar{f}(c)$. We therefore define an indicator counting the number of users labeled with a given candidate:

$$\mathsf{ClassUserCount}_{\mathcal{H}}(c) = |\{u \in \mathcal{U} \mid c = \lambda_{\mathcal{H}}(u)\}|$$

We finally found interesting to focus on the candidates mentions only instead of the set of hash-tags $\mathcal{H}$. Analogously to $\mathsf{ClassTweetCount}_{\mathcal{H}}$ and $\mathsf{ClassUserCount}_{\mathcal{H}}$, we can define new labeling functions $\lambda_{\mathcal{C}}$ based on a new score function $S_{\mathcal{C}}$:

$$S_{\mathcal{C}}(c|t) = \sum_{h \in t \cap \mathcal{C}} P(c|h)$$

Given $\lambda_{\mathcal{C}}$, we thus define the following strategies:

$$\mathsf{ClassTweetCount}_{\mathcal{C}}(c) = |\{t \in \mathcal{T} \mid c = \lambda_{\mathcal{C}}(t)\}|$$
$$\mathsf{ClassUserCount}_{\mathcal{C}}(c) = |\{u \in \mathcal{U} \mid c = \lambda_{\mathcal{C}}(u)\}|$$

Table 2 shows the performance of the above strategies exploiting classification of tweets and users. The two most promising are UserShare and $\mathsf{ClassTweetCount}_{\mathcal{C}}$. These strategies are both very simple as they consider only the hash-tags corresponding to candidates mentions. In UserShare, a single user vote is *split* among the candidates, while in $\mathsf{ClassTweetCount}_{\mathcal{C}}$ a tweets is classified as a vote to only one of the candidates. Both approaches provide a significant improvement of about 25% over the baseline strategies both in terms of MAE and RMSE. The MRM score is still too low to draw final conclusions.

Training correcting factors

One of the assumptions of this work is that Twitter users are not a representative sample of the voters population. Even if we were able to correctly classify each Twitter user, we would not be able to make a reliable estimate of the voting results as (i) several Twitter users may not vote, (ii) several voters are not present on Twitter, and (iii) the voters of each candidate have a different degree of representativeness in Twitter.

Given a predictor $\phi(c)$, we aim at learning a set of weights w_c, one for each candidate, such that $w_c\phi(c)$ improves the estimate of actual votes received. The weights

Table 3. Machine-learned weighting performance.

Algorithm	MAE	RMSE	MRM
ML-UserShare	0.0536	0.0705	**0.75**
ML-ClassTweetCount$_\mathcal{C}$	0.0533	0.0663	0.69
ContentAnalysis	**0.0525**	**0.0630**	0.70

w_c should act as a bridge correcting an estimate based on Twitter users to fit real world users behavior.

We aim at *learning* the weights w_c. For each region of Italy and for each candidate c, we create a training instance $\langle y_c, x_c \rangle$, where y_c is the *target variable* being equal to the percentage of votes actually achieved by c in the given region, and x_c is the *input variable* equal to a given estimator $\phi(c)$. In general, a vector of *input variables* can be used. We thus have a *training data set* with 60 training instances coming from 20 regions and 3 candidates. To conduct a 5-fold cross validation the data set was split region-wise in training and test sets. The training set was used to learn a weight w_c via linear regression that minimizes $(y_c - w_c \cdot \phi(c))^2$. We applied this approach to the two most performing predictors evaluated so far, i.e., UserShare and ClassTweetCount$_\mathcal{C}$. We name the corresponding *machine learned* strategies ML-UserShare and ML-ClassTweetCount$_\mathcal{C}$. As reported in Table 3 these new approaches provide a significant improvement according to all metrics. The improvement is of about 15% in terms of MAE and 10% in RMSE. A huge improvement is observed according to the MRM metric. For instance, ML-UserShare is able to provide the correct candidate ranking in 15 out of 20 regions. This means that we were able to reduce the prediction error on the votes share (both MAE and RMSE) up to the point of being able to correctly predict the final ranking of the candidates. By inspecting the weights learned by the ML-UserShare strategy, we see that Renzi, Cuperlo and Civati have weights 1.02, 1.24 and 0.70 respectively. This means that the second candidate is *under-represented* in the Twitter data, and symmetrically for the third candidate. In Fig. 4 we show the actual voting results and the estimations produced by UserShare and ML-UserShare. The correcting weights of ML-UserShare have sometimes the effect of inverting the rank generated by UserShare of the two candidates Cuperlo and Civati, in agreement with the actual election results.

The drawback of this approach is that it requires a training data where to learn the correction weights w_c. This makes it not possible to directly apply the method before the election takes place. On the other hand, we can assume that weights are sufficiently stable, i.e., that the degree of representativeness of the Twitter sample for a specific sample does not change abruptly. If this is the case, then we can learn those weights by exploiting data from previous events. Indeed, it would be possible exploit elections at municipality, regional and European level to learn a proper set of weights for national elections. Another interesting case is that of a two-round voting system, where the model could be trained after the first round and used to predict the outcome of the second. Yet another option is to complement prediction with traditional polls data.

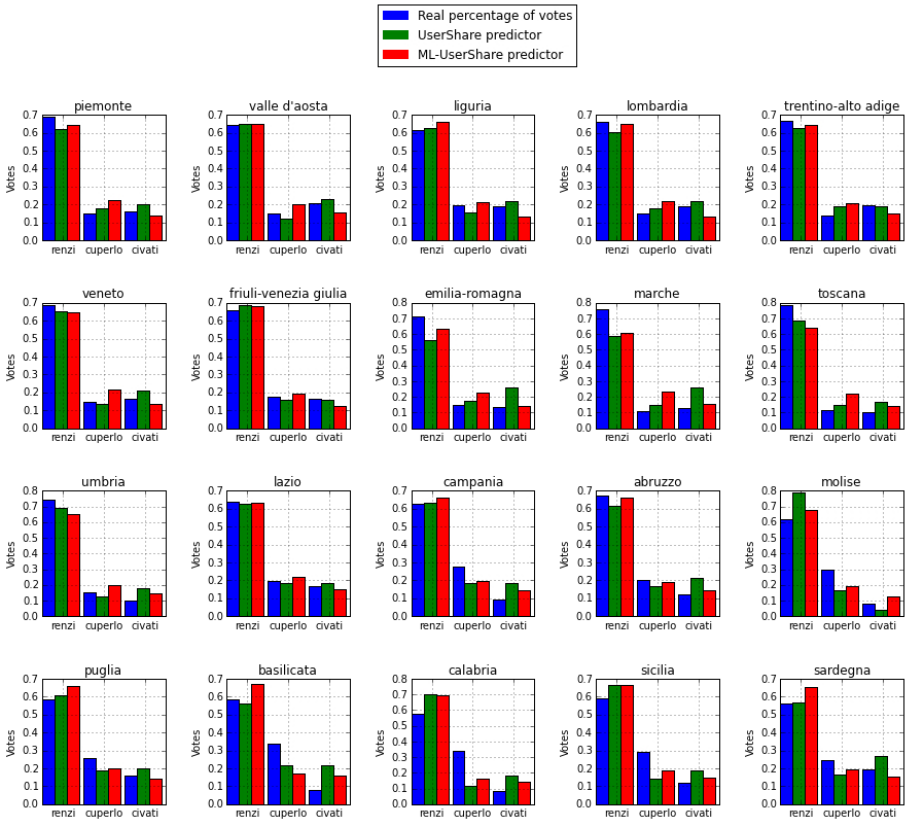

Fig. 4. Regional predictions and actual voting results.

Including Content-Based Analysis

The above approach is very general as several features about a candidate can be considered altogether by extending the input variable x to a vector of input variables. We propose to include text analysis and semantic analysis as follows. We considered the top 100 words (not only hash-tags) most frequently occurring in the data set after stop-word removal and stemming. These 100 words $\mathcal{W}$ where used to build a *content-based feature vector*. For each candidate and for each region, we computed the number of occurrences of each word in $\mathcal{W}$ normalized by the number of tweets in the region considered. This new feature vector include the names of the candidates, but it is also likely to include, if frequent, other significant *names*, *topics* or *catch-phrases* which are relevant to estimate the reach of a candidate.

Similarly as for ML-UserShare, we built a new training data set where for each training instance $\langle y_c, x_c \rangle$, x_c contains the *content-based feature vector* for c, to which we also included the predictor UserShare. The resulting model should be able to blend text analysis with the predictor UserShare. The weight vectors for each candidate were learned with LASSO linear regression. The resulting predictor is named ContentAnal-

ysis. As shown in Table 3, ContentAnalysis achieves the best MAE and RMSE, and a good value of MRM.

Demographic analysis

We think that the main issue of any social network analysis, aiming at understanding public opinion, is that social networks are not a representative sample of people, or, in this context, of the voters. The bias introduced by Twitter should be carefully taken into consideration. From the data we collected, it was not possible to infer details about users, e.g., age, education or other. We resorted to analyze the Twitter demographical bias through external polls on the age distribution of voters[6]. We compared the UserShare predictor against the expected result of 5 age range classes. The results are reported in Table 4 ordered by MAE, showing that UserShare is more accurate in predicting the votes of people in the range of 25-44 years old. It is known that the average age of an Italian Twitter user is 32 years (larger than the world average age which is 24), according to a report of Pew Research published in 2013, confirming our preliminary results. This suggests that Twitter analyses and traditional polls can be complemented together in order to achieve a wider coverage.

Table 4. Error of UserShare by age class.

Age class	MAE
16-24 years	0.1409
25-44 years	0.0216
45-54 years	0.0476
55-64 years	0.0636
$>$ 65 years	0.0709

Table 5. Estimations at national level

Algorithm	MAE	RMSE
TweetCount	0.0541	0.0641
UserShare	0.0413	0.0462
Polls	0.0386	0.0418

Aggregated outcome

Finally, in order to provide a full picture of our analysis, we provide estimations at national level, i.e., by considering the whole data-set without partitioning by region without machine learning approach.

Table 5 shows the performance of TweetCount [19] and UserShare. We also report the average error of the electoral polls made by different polling institutes (period 26 Nov - 04 Dec), as it is reported in *termometropolitico.it*, a website which collects and comments political polls before elections.

The two methods TweetCount and UserShare are very close to the polls error, and we can explain this error with the age sampling bias which is discussed in the previous section. Note that we didn't use any machine learning to improve the prefiction in this case. Finally, recall that the cost of traditional polling is obviously higher that the cost of twitter monitoring.

[6] Data from polls performed by Quorum (polling Institute).

Conclusion

In this work, we tackled the problem of providing accurate estimation of real world phenomena through social network analysis with three novel contributions.

First, we evaluated counting-based state-of-the-art methods, and we proposed an enhanced user centered predictor that models every single user with a voting probability across the candidates. This predictor improved by 25% the baseline methods.

Then, we addressed the main issue of the social network sample bias. We proposed a few machine learning approaches, also including content-based analysis, with the goals of learning bias correcting factors. In our case, we were able to estimate the over or under representativeness of each candidate in our data. We believe that exploiting machine learning, both for an improved classification of users and for correcting the sample bias is a crucial task in social network analysis. The main drawback of such techniques is that they require training data. We believe that such drawback can be overcome by exploiting continuous analysis over time leveraging related events, e.g., political elections at any level. How to transfer the knowledge gained in one analysis to other scenarios is an open research problem.

In conclusion, we believe that major improvements in the field can be achieved by integrating several sources of information, such as traditional polls, multiple social networks, demographic data, historical data, analyses of related events, content-based and network-based properties. Such wealth of information can be exploited altogether through machine learning approaches. The integration of all of these approaches may open up new research challenges and opportunities in the field.

References

1. Asur, S., Huberman, B.A.: Predicting the future with social media. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. vol. 1, pp. 492–499. IEEE (2010)
2. Bermingham, A., Smeaton, A.F.: Classifying sentiment in microblogs: is brevity an advantage? In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 1833–1836. ACM (2010)
3. Bermingham, A., Smeaton, A.F.: On using twitter to monitor political sentiment and predict election results (2011)
4. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of Computational Science 2(1), 1–8 (2011)
5. Caldarelli, G., Chessa, A., Pammolli, F., Pompa, G., Puliga, M., Riccaboni, M., Riotta, G.: A multi-level geographical study of italian political elections from twitter data. PloS one 9(5), e95809 (2014)
6. DiGrazia, J., McKelvey, K., Bollen, J., Rojas, F.: More tweets, more votes: Social media as a quantitative indicator of political behavior. PloS one 8(11), e79449 (2013)
7. Fanelli, D.: Do pressures to publish increase scientists' bias? an empirical support from us states data. PloS one 5(4), e10271 (2010)
8. Gayo-Avello, D.: Don't turn social media into another'literary digest'poll. Communications of the ACM 54(10), 121–128 (2011)
9. Gayo-Avello, D., Metaxas, P.T., Mustafaraj, E.: Limits of electoral predictions using twitter. In: ICWSM (2011)

10. Giglietto, F.: If likes were votes: An empirical study on the 2011 italian administrative elections. (2012)
11. Jungherr, A., Jürgens, P., Schoen, H.: Why the pirate party won the german election of 2009 or the trouble with predictions: A response to tumasjan, a., sprenger, to, sander, pg, & welpe, im "predicting elections with twitter: What 140 characters reveal about political sentiment". Social Science Computer Review 30(2), 229–234 (2012)
12. Lampos, V., De Bie, T., Cristianini, N.: Flu detector-tracking epidemics on twitter. In: Machine Learning and Knowledge Discovery in Databases, pp. 599–602. Springer (2010)
13. Lazer, D., Pentland, A.S., Adamic, L., Aral, S., Barabasi, A.L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., et al.: Life in the network: the coming age of computational social science. Science (New York, NY) 323(5915), 721 (2009)
14. Metaxas, P.T., Mustafaraj, E., Gayo-Avello, D.: How (not) to predict elections. In: Privacy, security, risk and trust (PASSAT), 2011 IEEE third international conference on and 2011 IEEE third international conference on social computing (SocialCom). pp. 165–171. IEEE (2011)
15. O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From tweets to polls: Linking text sentiment to public opinion time series. ICWSM 11, 122–129 (2010)
16. Sang, E.T.K., Bos, J.: Predicting the 2011 dutch senate election results with twitter. In: Proceedings of the Workshop on Semantic Analysis in Social Media. pp. 53–60. Association for Computational Linguistics, Stroudsburg, PA, USA (2012), http://dl.acm.org/citation.cfm?id=2389969.2389976
17. Skoric, M., Poor, N., Achananuparp, P., Lim, E.P., Jiang, J.: Tweets and votes: A study of the 2011 singapore general election. In: System Science (HICSS), 2012 45th Hawaii International Conference on. pp. 2583–2591. IEEE (2012)
18. Smith, A.W., Rainie, H.: The Internet and the 2008 election. Pew Internet & American Life Project (2008)
19. Tumasjan, A., Sprenger, T.O., Sandner, P.G., Welpe, I.M.: Predicting elections with twitter: What 140 characters reveal about political sentiment. ICWSM 10, 178–185 (2010)
20. Williams, C., Gulati, G.: What is a social network worth? facebook and vote share in the 2008 presidential primaries. American Political Science Association (2008)