Predicting Gaming Related Properties from Twitter Profiles

Alfredo Kalaitzis*, Maria Ivanova Gorinova†, Yoad Lewenberg‡,
Yoram Bachrach§, Michael Fagan*, Dean Carignan** and Nitin Gautam**

*Microsoft, London, UK
†University of Cambridge, Cambridge, UK
‡The Hebrew University of Jerusalem, Jerusalem, Israel
§Microsoft Research, Cambridge, UK
**Microsoft, Redmond, US

Abstract—**We present a system for predicting gaming-related properties from Twitter profiles. Our system predicts various traits of users based on the tweets publicly available on their profiles. Such inferred traits include degrees of tech-savviness, knowledge on computer games, actual gaming performance, preferred platform, degree of originality, humor and influence on others. Our approach is based on machine learning models trained on crowd-sourced data. Our system enables people to select Twitter profiles of their fellow gamers, examine the trait predictions made by our system, and the main drivers of these predictions. We present empirical results on the performance of our system based on its accuracy on our crowd-sourced dataset. Ultimately, we are motivated by the automated discovery of influential gamers in social media, and its potential for streamlining product campaigns.**

1. Introduction

Big-data organizations strive to make consumer outreach increasingly data-driven. To this end, it is vital to automatically infer from user-data traits that are thought to be instrumental in launching a successful product campaign.

Social media plays a pivotal role in our lives as consumers, and services such as Twitter, Facebook and Google+ are used regularly by over a billion users. Recent research has uncovered many ways in which online information, including social network data, can be used to predict personal traits of users [1], [2], [3], [4], [5], [6], [7], [8], [9].

Such information can provide insight on users [10], [11], or accommodate commercial applications such as personalized search [12], targeted advertising [13], or improving the quality of collaborative-filtering-based recommender systems. [1]

This earlier work focuses on the general population, whereas our focus is on the specific target group of computer gamers. Gamers are predominantly active in social media, and use distinct online communication styles and language.

We focus on the following *perceived* traits of gamers, assumed to affect their standing in the gaming community: *tech-savviness*, degree of *knowledge* on computer games, and *gaming skill* in various genres. We also infer variables such as their *life-stage*, degree of *originality*, and level of *influence* on their peers.

Our system accepts a Twitter handle, and predicts the traits of its owner. These predictions are the result of applying *supervised* machine learning on the textual tweets made by the target user.

The rest of the paper describes the **methodology** for building our prediction system. Namely, the crowd-sourcing of annotations (2.1), the set of *features* extracted from every raw tweet (2.2), and the training of machine learning *classifiers* and *regressors* (depending on the trait), using instances of such *feature-sets*. In section 2.1 we dive into the specific traits that we are interested in predicting. In the **experiments**, section 3, we show empirical results of system in

[1] Standard recommender systems only use information on consumer items and rely on fingerprinting or dimensionality reduction techniques [14], [15], [16], but can be adapted to incorporate more detailed user profiling [17], [18], [19], [20].

predicting gaming-related traits. We also address *inter-rater disagreement* in section 3.1, a common problem in crowd-sourcing scenarios, by computing a specific type of *intra-class correlation* (ICC) for each rated trait. These measures clarify our perspective on the impact to predictive capacity and, ultimately, highlight the limitations of our system. In **conclusion**, section 4, we discuss the main results of this prediction exercise, the merits of our tool from a business point of view, as well as possible extensions.

2. Methodology

We build a suitable machine learning model (regressor or classifier) for each gaming-related traits (c.f. list of traits in 2.2), and train it on a dataset of 2,000 Twitter profiles, annotated by workers on Amazon's Mechanical Turk.

2.1. Crowd-sourced annotation

We asked 646 workers to rate 2,000 English-speaking Twitter profiles. Each worker was asked to examine several of those profiles and form an opinion regarding the traits of the profile owners. Each Twitter profile was annotated 3.12 times on average. [2] For each profile rated, we asked the worker to provide their opinion regarding the following traits of the profile owner:

Categorical:

- gender `(male, female)`
- fan of Xbox `(yes, no)`
- fan of Playstation `(yes, no)`

Ordinal:

- age range `(18-,18-25,25-30,30-45,45+)`
- life stage `(high school, university, young professional, established pro, retired)`
- tech-savviness `(1-5)`
- knowledge level of games `(1-5)`
- trustworthiness `(1-5)`
- content quality / depth `(1-5)`
- humour `(1-5)`
- originality `(1-5)`
- level of influence `(1-5)`

where the ordinal ratings `(1-5)` map to (`very low`, `low`, `medium`, `high`, `very high`), respectively.

In addition, we augment these crowd-sourced ratings, with direct measures of time that a target user has spent playing Xbox games, and their actual achievement scores .

2.2. Feature extraction

The textual data of the users in the training dataset are pre-processed by reducing all words to their root form, via a *Porter stemmer* [28]. The stemmed text is then used to extract a vocabulary, which consists of those words and hashtag words (those prefixed by '#') that are present in at least 3 user profiles and at most 80% of all user profiles.

We combine *lexical* and *stylistic* features to create a training dataset for the machine learning models, and to predict the traits of users previously unseen by our models.

- The *lexical* part of a feature-set is obtained by mapping the terms of all tweets in a profile into a vector representation of *term frequencies-inverse document frequencies* (TF-IDF). This is done with respect to the vocabulary extracted from the tweet-corpus. A TF-IDF weight captures the respective term's importance to a document, relative to the term's usage in the corpus [29].
- The *stylistic* part of the feature-set includes the occurrences of elongated words, fully capitalized words, consecutive punctuation marks, hashtags, as well as the percentage of the profile's messages that were retweets or replies, and the number of URLs that the user has shared.

2.3. Predictive models

As per the list in 2.1, we distinguish between *categorical* and *ordinal* traits. To predict an ordinal

[2] We have used redundant labels as crowdsourced data is know to be very noisy. [21] Earlier work has shown that aggregating responses can improve data quality even using simple aggregation such as majority vote [22], [23], and there are also various Bayesian data aggregation methods [24], [25], [26], [27]. We sourced redundant labels so as to achieve more robust models.

trait y from tweet features $\mathbf{x}$, we use an *ordinary least squares* (OLS) regression model:

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + \mathbf{w}^\top \mathbf{x} \ , \tag{1}$$

where the learnt model-weights $\mathbf{w}$ minimize the sum of squared errors over n datapoints:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2 \ . \tag{2}$$

Similarly, to classify a categorical trait y from tweet features $\mathbf{x}$, we use a *logistic regression* model:

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-(w_0 + \mathbf{w}^\top \mathbf{x})}} \in [0, 1] \ , \tag{3}$$

where $\mathbf{w}$ minimizes the negative log-likelihood:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \ -\log \prod_{i=1}^{n} h_{\mathbf{w}}(\mathbf{x}_i)^{y_i} (1 - h_{\mathbf{w}}(\mathbf{x}_i))^{(1-y_i)} \ . \tag{4}$$

Due to the size of the feature-set being in the order of 10^5, and because (4) admits no closed-form solution, these objectives are solved via gradient descent.

For prediction on an unseen profile, our system accepts its Twitter handle, scrapes the recent tweets of the user via the Twitter API (first 20 pages) and, where appropriate, automatically translates them through the Microsoft Translator API.

3. Experiments

3.1. Inter-rater disagreement and reliability

We quantify the disagreement amongst raters of a Twitter profile with the *intra-class correlation* (ICC), a statistical measure of inter-rater reliability. The ICC is apt for annotations[3] that can be ordered, for instance, when a difference in *humour* ratings 1 and 5 shows more disagreement than ratings 2 and 4. In contrast, any difference in categorical annotations carries the same weight in disagreement. In this paper, we use the ICC to quantify inter-rater reliability for all traits. This is because the interpretation of ICC on a binary categorical trait is as valid as on any quantitative trait[4].

We also compute a secondary agreement measure, *Fleiss' kappa* (or κ), which does not account for any intrinsic order in the annotations, if present. This makes κ particularly suitable to categorical traits with more than two categories. Regardless, $kappa$ remains a valid choice for ordinal traits, but tends to underestimate the *reliability* in such traits. We discuss the interpretation of κ in 3.1.2.

3.1.1. ICC. There are several definitions of intra-class correlations used in the earlier literature. The most recent and used is a family of ICC measures defined in [30]. The class of ICC used in this paper is denoted in this literature as $\mathrm{icc}(1, k)$. This class is suitable in scenarios where for each of n subjects, k raters are randomly selected from a population of raters, and asked to rate that subject. Hence, each subject can be rated by different raters, as is the case with our crowd-sourced annotation setup.

For a particular trait t, we denote

$$\mathbf{A}^{(t)} \in \mathbb{R}^{n \times k}, \tag{5}$$

to be its matrix of annotations, where each row is a subject annotated by k random raters, and

$$\mu_i = \frac{1}{n} \sum_j A_{i,j} \tag{6}$$

$$\mu = \frac{1}{nk} \sum_{i,j} A_{i,j} \ , \tag{7}$$

as the mean of row i and the general mean, respectively.

All classes of ICC can be computed from within the *analysis-of-variance* (ANOVA) framework. Specifically, $\mathrm{icc}(1, k)$, or simply icc as denoted henceforth, is the ratio:

$$\mathrm{icc} = \frac{\mathrm{v}_b - \mathrm{v}_w}{\mathrm{v}_b} \ , \tag{8}$$

where

$$\mathrm{v}_w = \frac{1}{n(k-1)} \sum_{i,j} \left(A_{i,j}^{(t)} - \mu_i \right)^2 \text{ and} \tag{9}$$

$$\mathrm{v}_b = \frac{1}{n-1} \sum_i k \left(\mu_i - \mu \right)^2 \tag{10}$$

[3] Strictly speaking, *annotations* are categorical and *ratings* are quantitative. In this paper we use the terms interchangeably.

[4] ICC becomes invalid for more than two categories.

are the *within subject* and *between subject* variances, respectively. Because $v_b \geq v_w$, the icc is always measured in the $[0, 1]$ interval, from *no agreement* to *perfect agreement*. It is now clear from (8) how the interpretation of icc is taken directly from the ANOVA context, and more concretely, as the *percentage of variation that is not explained by inter-rater disagreements*. It also shows why icc would be a valid choice for binary categorical traits, like *gender*.

3.1.2. Fleiss' kappa. An alternative reliability measure is *Fleiss' kappa* [31], or κ, which has a different interpretation to the icc. The distance between annotation now has no effect and only the number of agreeing pairs of raters is taken into account.

Let $\mathbf{A}^{(t)} \in \mathbb{N}^{n \times k}$ be an integer annotations matrix for trait t, where n is the number of subjects and k is now the number of categories or levels of the trait. $A^{(t)}_{i,j}$ is the number of raters that annotate subject i into category j. [5] Then κ of trait t is computed with the following simple algorithm:

```
for $i = 1$ to $n$ do
    $R_i \leftarrow \sum_j A_{i,j}$   # number of raters of i
    $P_i \leftarrow R_i(R_i - 1)/2$   # all pairs of raters of i
    $Q_i \leftarrow \sum_j A_{i,j}(A_{i,j} - 1)/2$   # agreeing pairs
end for
$p_a \leftarrow \frac{1}{n}\sum_i Q_i/P_i$   # mean agreement probability
$S \leftarrow \sum_{i,j} A_{i,j}$   # total ratings
for $j = 1$ to $k$ do
    $\pi_j \leftarrow \sum_i A_{i,j}/S$   # proportion of category j
end for
$p_c \leftarrow \sum_j \pi_j^2$   # probability of agreement by chance
```

$$\kappa = \frac{p_a - p_c}{1 - p_c} \tag{11}$$

By the last equation, κ is interpreted as the *degree of agreement over that which would be expected by chance*. As such, κ takes value in $(-\infty, 1]$ and it is read as:

- *poor agreement*, for values in $(-\infty, 0]$;
- *poor agreement* to *perfect agreement*, for values in $(0, 1]$.

We list the icc and κ reliability scores for each trait in figure 1. Amongst the most 'agreeable' traits, we encounter *gender*, *age*, *life stage*, *Xbox fan* and *knowledge*. We note that κ is always lower than icc, but their values are correlated. Surprisingly, *PS fan* did not rank as high on agreement as *Xbox fan*, but it is unclear how to best interpret this result.

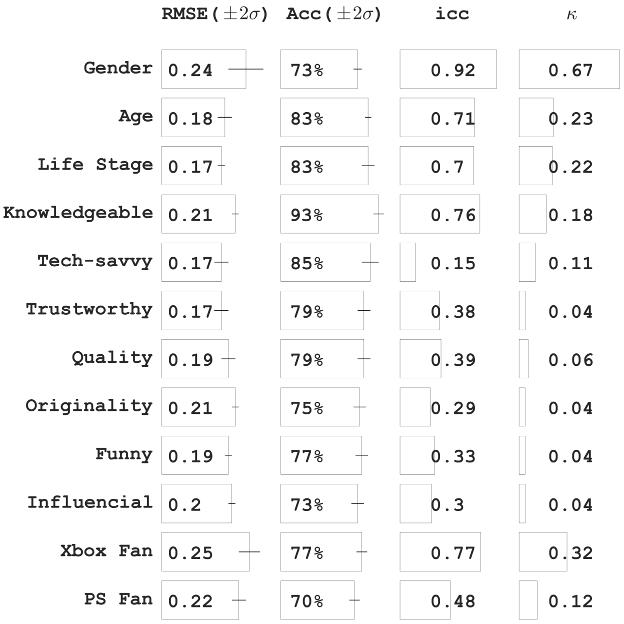


Figure 1. Inter-rater agreement (icc and κ) and performance of predictive models (rmse and accuracy – averages of 10-fold cross-validation). Each column is a bar-chart with horizontal lines as 95% confidence intervals, where appropriate.

3.2. Prediction performance

3.2.1. Distribution of data. We train *both* classification and regression models for each trait, binary-categorical or ordinal. [6] We do this because the response variable for each trait and profile is an average across the annotations of that profile's raters. So even though a trait of interest can be binary in nature, its *averaged* perception across social media spans a real-valued interval, c.f. 3.1. This is illustrated in figure 2, which shows the distribution of annotations for each traits. Intuitively, the two most correlated bivariate distributions are *life-stage – age* and *tech-savvy – knowledgeable*. We are interested in the bottom row of figure 2, as it conveys the traits most correlated to *influential*: *trustworthy*, *quality* and *original*.

[5] Note that this definition allows for the number of raters to vary from subject to subject.

[6] Models and experiments are coded with the *scikit-learn* open-source Python library [32].

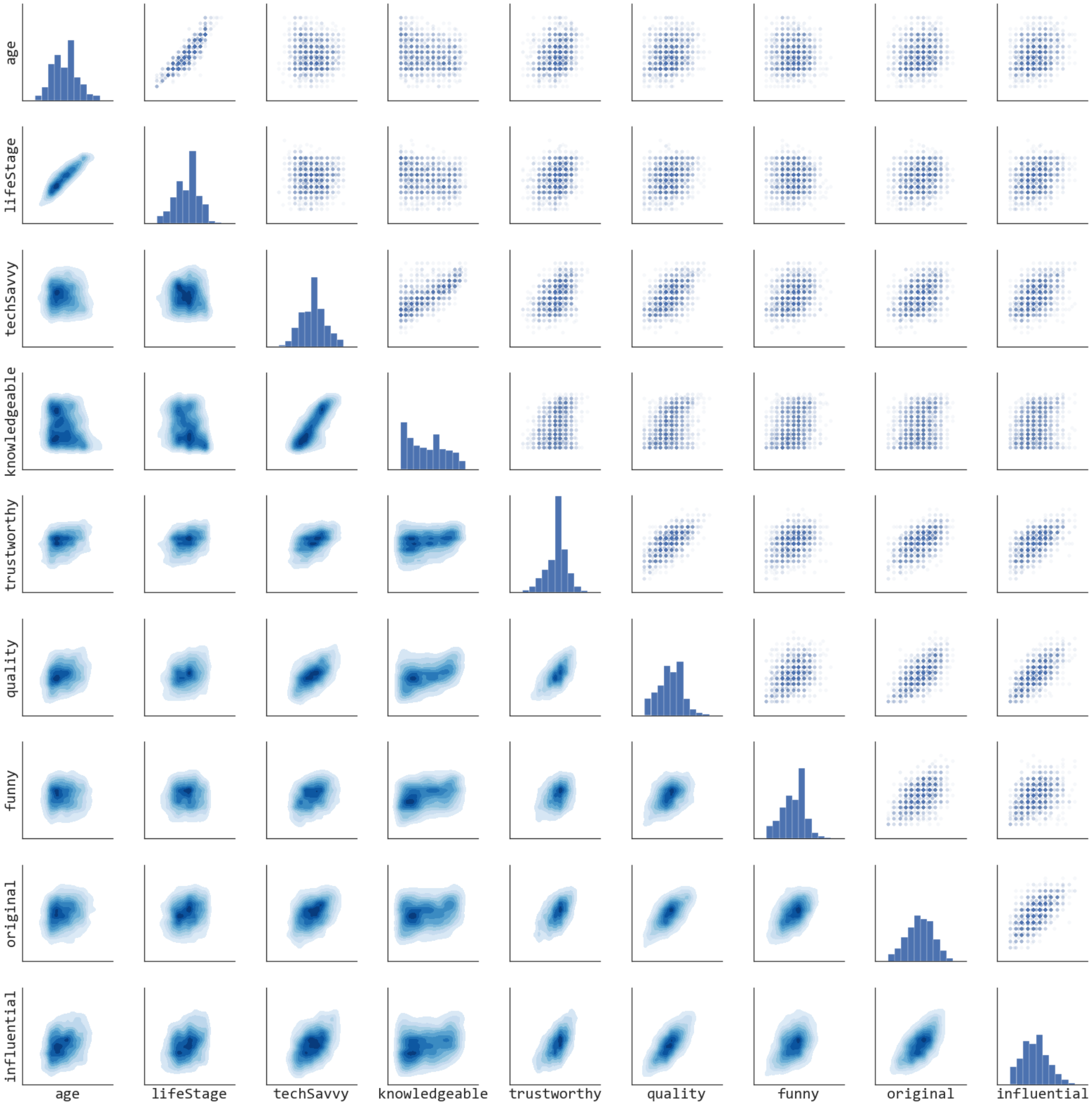
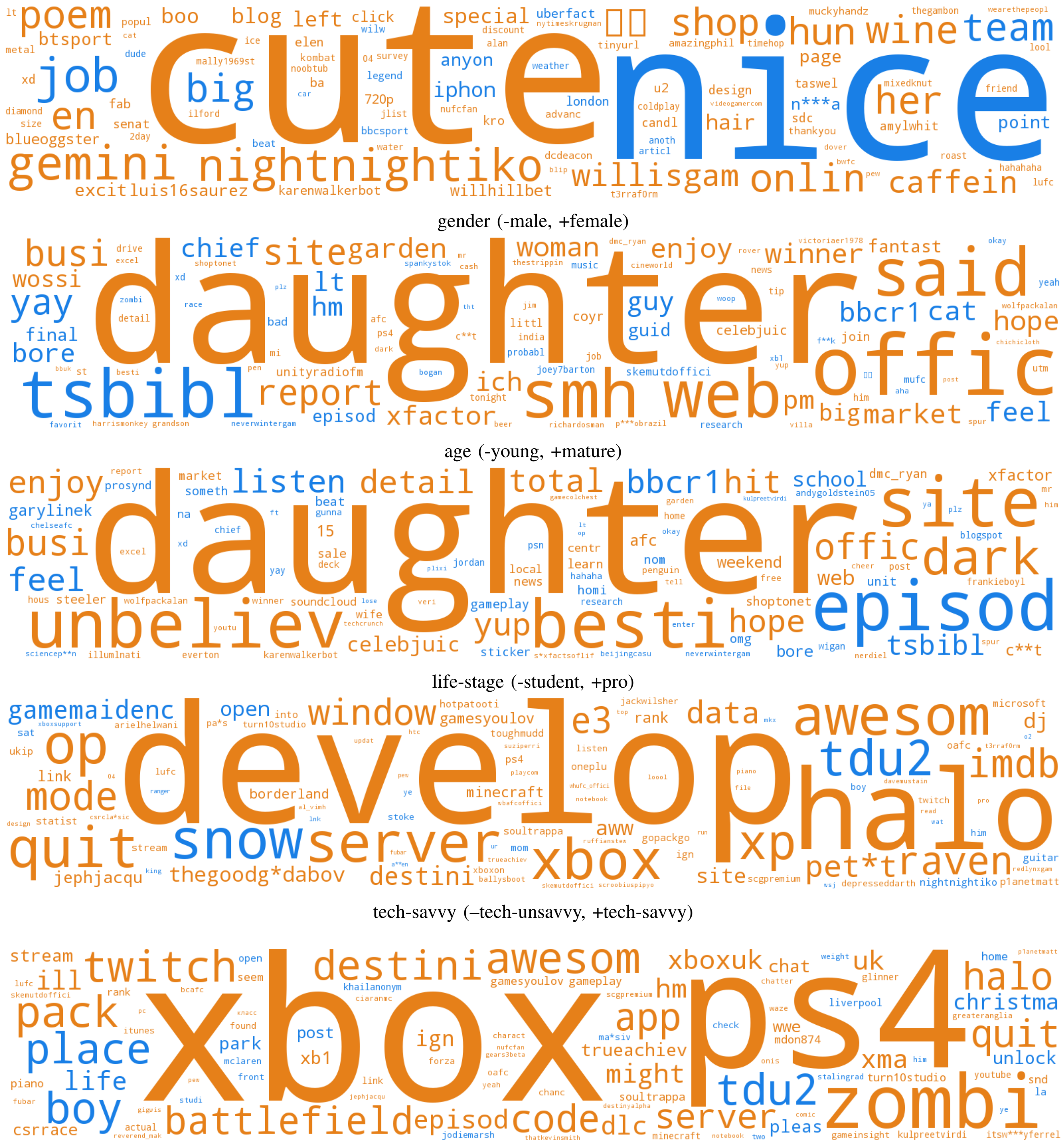


Figure 2. Pair-plot of all bivariate distributions between all but *gender*, *Xbox fan* and *PS fan*. The histograms rendered on the diagonal of the grid are marginal distributions of annotation per trait, across the 2,000 rated profiles. Scatterplots are featured on the upper triangular grid and kde-plots (*kernel density estimators*) on the lower triangular. All variables are normalized in the $[0, 1]$ range prior to training and testing, so all plots are rendered on the unit-square. Coded with the *seaborn* open-source Python library [33].



gender (-male, +female)

age (-young, +mature)

life-stage (-student, +pro)

tech-savvy (–tech-unsavvy, +tech-savvy)

knowledgeable (–unknowledgeable, +knowledgeable)

Figure 3. Bag-of-word features as word-clouds, where size indicates the magnitude of a feature-weight after training (TF-IDF coefficient in linear regression) and the color indicates the sign of the weight – orange for positive, blue for negative. Censored parts of words are shown with asterisks. Coded with the *wordcloud* open-source Python library.

We use two measures for the quality of prediction. One is the *root mean square error* (RMSE) of the numerical predictions (on the five point scale for ordinal or $\{0, 1\}$ Boolean scale, for categorical). The RMSE is measured in the units of the annotations normalized in the $[0, 1]$ interval. The second is based on partitioning the user population into thirds, by sorting the users from the highest to lowest score of the predicted trait. We can then examine the two extreme thirds, and check the prediction accuracy of determining whether a user is in the top or bottom third (ignoring the middle third). Figure 1 shows the accuracy of out predictions (measured using 10-fold cross validation).

The word-clouds in Figure 3 show the relative importance of each feature in the bag-of-words vocabulary for the best classified traits. At first glance, the word-clouds do not offer much insight into how these traits are perceived, but a quick comparison *between* word-clouds reveals strong similarities between expected pairs, such as *age* and *life-stage*, as well as *tech-savvy* and *knowledgeable*. Besides this cursory observation, a correlation analysis between the perception of traits is outside the scope of this paper.

In addition to perceived traits, we have also built similar models to predict the time that a gamer has spent playing computer games and their actual performance in computer games. [7] Our results indicate a prediction accuracy 59% for the time spent playing games, and 64% for actual performance in playing games, referred to as "Gamer Score" (the accuracy is for separating users in the top third and those in the bottom third of these properties). [8]

4. Conclusion

Figure 1 indicates that it is indeed possible to predict many perceptions on gamers from the language they use in online social networks. A few properties are more difficult to determine than others, and in particular, those with the smallest of inter-rater agreement (ICC) scores. These experiments indicate that information from online social networks is predictive of several objective gaming-related traits of users. We emphasize that the choice of feature representation is at least as important as the choice of learning algorithm. Instead, the focus and novelty is on the demonstrated prediction pipeline, and its potential as a marketing tool.

Naturally, many trait perceptions are expected to be correlated. An obvious improvement is the consideration of such correlations. As of yet, we are learning models independently for each trait, without accounting for inter-trait correlations. A more accurate model would be one of simultaneous prediction of multiple traits, akin to multi-output linear regression, in the case that the variables or responses were real-valued numbers.

For future work, we aim to quantify the uncertainty in estimates of inter-rater reliability, via probabilistic graphical models and Markov chain Monte Carlo (MCMC) sampling. This will provide a means to access the reliability of individual rater is a rigorous probabilistic manner, therefore mitigating the main limitation of our pipeline.

References

[1] J. Golbeck, C. Robles, M. Edmondson, and K. Turner, "Predicting personality from Twitter," in *Proceedings of SocialCom/PASSAT*, 2011.

[2] Y. Bachrach, M. Kosinski, T. Graepel, P. Kohli, and D. Stillwell, "Personality and patterns of Facebook usage," in *Proceedings of WebSci*, 2012, pp. 24–32.

[3] M. Kosinski, D. Stillwell, P. Kohli, Y. Bachrach, and T. Graepel, "Personality and website choice," 2012.

[4] M. Kosinski, Y. Bachrach, P. Kohli, D. Stillwell, and T. Graepel, "Manifestations of user personality in website choice and behaviour on online social networks," *Machine Learning*, vol. 95, no. 3, pp. 357–380, 2014.

[5] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *National Academy of Sciences*, 2013.

[6] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. Seligman *et al.*, "Personality, gender, and age in the language of social media: The open-vocabulary approach," *PloS one*, vol. 8, no. 9, p. e73791, 2013.

[7] Y. Bachrach, "Human judgments in hiring decisions based on online social network profiles," in *DSAA*, 2015.

[8] D. Preoţiuc-Pietro, S. Volkova, V. Lampos, Y. Bachrach, and N. Aletras, "Studying user income through language, behaviour and affect in social media," *PloS one*, vol. 10, no. 9, p. e0138717, 2015.

[9] S. Volkova, Y. Bachrach, M. Armstrong, and V. Sharma, "Inferring latent user properties from texts published in social media," in *AAAI*, 2015.

[7] We had access to such data from the user profiles in the Xbox platform data.

[8] Interestingly, our methods achieved better predictions for the actual ability in playing computer games than for the time spent playing them.

[10] Y. Bachrach, T. Graepel, P. Kohli, M. Kosinski, and D. Stillwell, "Your digital image: factors behind demographic and psychometric predictions from social network profiles," in *AAMAS*, 2014, pp. 1649–1650.

[11] Y. Lewenberg, Y. Bachrach, and S. Volkova, "Using emotions to predict user interest areas in online social networks," in *DSAA*, 2015.

[12] Y. Ustinovsky and P. Serdyukov, "Personalization of web-search using short-term browsing context," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, 2013, pp. 1979–1988.

[13] W.-S. Yang, J.-B. Dia, H.-C. Cheng, and H.-T. Lin, "Mining social networks for targeted advertising," in *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, vol. 6. IEEE, 2006, pp. 137a–137a.

[14] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[15] Y. Bachrach, Y. Finkelstein, R. Gilad-Bachrach, L. Katzir, N. Koenigstein, N. Nice, and U. Paquet, "Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces," in *RecSys*, 2014.

[16] Y. Bachrach and E. Porat, "Fingerprints for highly similar streams," *Information and Computation*, 2015.

[17] M. Clements, P. Serdyukov, A. P. de Vries, and M. J. T. Reinders, "Personalised travel recommendation based on location co-occurrence," *CoRR*, vol. abs/1106.5213, 2011.

[18] Y. Bachrach, S. Ceppi, I. A. Kash, P. Key, F. Radlinski, E. Porat, M. Armstrong, and V. Sharma, "Building a personalized tourist attraction recommender system using crowdsourcing," in *AAMAS*, 2014, pp. 1631–1632.

[19] Y. Bachrach, R. Herbrich, and E. Porat, "Sketching algorithms for approximating rank correlations in collaborative filtering systems," in *SPIRE*, 2009.

[20] Y. Bachrach, E. Porat, and J. S. Rosenschein, "Sketching techniques for collaborative filtering," in *IJCAI*, Pasadena, California, July 2009.

[21] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 2010, pp. 64–67.

[22] Y. Bachrach, T. Graepel, G. Kasneci, M. Kosinski, and J. Van Gael, "Crowd iq: aggregating opinions to boost performance," in *AAMAS*, 2012, pp. 535–542.

[23] M. Kosinski, Y. Bachrach, G. Kasneci, J. Van-Gael, and T. Graepel, "Crowd iq: Measuring the intelligence of crowdsourcing platforms," in *WebSci*. ACM, 2012, pp. 151–160.

[24] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The multidimensional wisdom of crowds," in *NIPS*, 2010, pp. 2424–2432.

[25] Y. Bachrach, T. Graepel, T. Minka, and J. Guiver, "How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing," *ICML*, 2012.

[26] M. Salek, Y. Bachrach, and P. Key, "Hotspotting-a probabilistic graphical model for image object localization through crowdsourcing." in *AAAI*, 2013.

[27] B. Shalem, Y. Bachrach, J. Guiver, and C. M. Bishop, "Students, teachers, exams and moocs: Predicting and optimizing attainment in web-based education using a probabilistic graphical model," in *ECML/PKDD*. Springer, 2014, pp. 82–97.

[28] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[29] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[30] P. E. Shrout and J. L. Fleiss, "Intraclass correlations: uses in assessing rater reliability." *Psychological bulletin*, vol. 86, no. 2, p. 420, 1979.

[31] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] M. Waskom, O. Botvinnik, P. Hobson, J. Warmenhoven, J. B. Cole, Y. Halchenko, J. Vanderplas, S. Hoyer, S. Villalba, E. Quintero, A. Miles, T. Augspurger, T. Yarkoni, C. Evans, D. Wehner, L. Rocher, T. Megies, L. P. Coelho, E. Ziegler, T. Hoppe, S. Seabold, S. Pascual, P. Cloud, M. Koskinen, C. Hausler, kjemmett, D. Milajevs, A. Qalieh, D. Allan, and K. Meyer, "seaborn: v0.6.0 (june 2015)," Jun. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.19108