Typeface Semantic Attribute Prediction from Rasterized Font Representations

Suvir Mirchandani Lucia Zheng Julia Gong
{smirchan, zlucia, jxgong}@stanford.edu

Motivation and Problem

- **Characterizing attributes of typefaces** is of interest for font search, selection, and pairing.
- Related work has used grouped typefaces using **font styles** (e.g. 'bold', 'italic') or **typographic attributes** (e.g. 'serif', 'all-capitals') for font clustering/generation [1, 2, 4, 5, 7], which is less *interpretable* and *meaningful*.
- Recent work [6] has introduced human-labeled **semantic attributes** for 200 typefaces, such as 'artistic', 'playful', and 'boring'.
- Machine learning can be applied to **semantic attribute regression** to generalize these human annotations to **unseen typefaces.**
- We improve on [3] in inducing a semantic attribute dataset for 18 typefaces from the dataset of 200 in [6] and introduce a novel downstream task: semantic attribute prediction.

Dataset and Task Formulation

Input: human-labeled 31-dimensional real-valued semantic attribute vectors with entries in [0, 1] for 200 typefaces (e.g. 'Source Sans Pro Semibold') in the semantic attribute dataset in [6]; FontJoy[1] embeddings for 1883 typefaces to be utilized for dataset induction.

Task: induce corresponding 31-D semantic attribute vectors for all 1883 FontJoy typefaces.

Figure 1. Examples of the diverse typefaces in the FontJoy dataset.
[1]See fontjoy, https://github.com/Jack000/fontjoy

NOTATION
F = set of all 1883 FontJoy embedding vectors
F_{161} = embedding vectors from FontJoy with known semantic attribute labels from O'Donovan (intersection of datasets [3] and [6])
f = font embedding vector from FontJoy CNN
a_s = predicted 31-D semantic attribute vector
a_s' = ground truth 31-D semantic attribute vector

We experiment with different data induction methods by using F_{161} to predict a_s' for each typeface in F.

Method: Semantic Attribute Induction

4-NN: As our baseline, we reproduce the optimal result in [3]. We cluster the F_{161} font embedding vectors and predict each a_s' using *k* = 4, a cosine distance metric, and relative distance weighting.

7-NN: We perform grid search over multiple values of *k,* three distance metrics (cosine, Manhattan, Euclidean), and weightings (unweighted, inverse distance, relative weighting). We out-perform the model in [6] with *k* =7, a cosine distance metric, and inverse distance weighting.

$$1 - \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2}\sqrt{\sum_{i=1}^{d} y_i^2}} \qquad \vec{f} = \sum_{i=1}^{k} w_i \vec{f}_i \quad w_i = \frac{1}{|d(f', f_j)|}$$

Cosine distance metric, *k*-NN weighting of nearest neighbors f_i, and inverse distance weighting

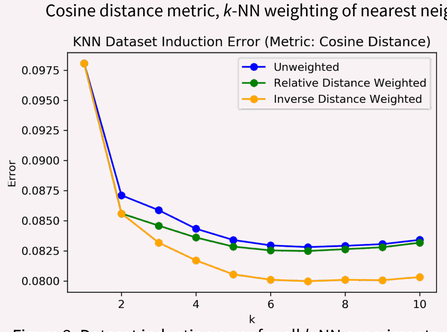


Figure 2. Dataset induction error for all *k*-NN experiments using the cosine distance for each of the three nearest neighbor weighting schemes across values of *k*.

The inverse distance metric performs the best across the board and performs best at *k* = 7.

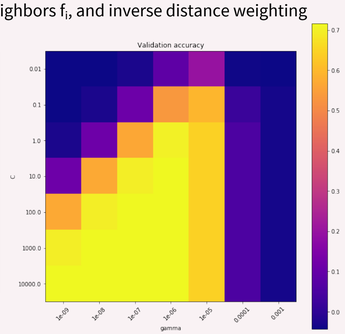


Figure 3. Support Vector Regressor hyperparameter grid search over gamma (γ) and C.

SVR (Support Vector Regressor): We use non-linear SVR models, where the kernel transforms the data into a higher-dimensional feature space to enable linear separation. We chose a *radial basis function kernel*:

$$K(x, x') = \exp\left(-\gamma ||x - x'||^2\right) \qquad (1)$$

We conduct exhaustive cross-validation grid search over hyperparameters γ and C for each of the 31 semantic attributes. C is a regularization parameter.

Final Semantic Attribute Induction Model

For our final data induction model, to leverage the power of all three data induction models (4-NN, 7-NN, and SVR), we ensemble a **meta-estimating voting regressor** performs weighted averaging over the individual predictions of these models to form a final prediction.

To measure the quality of these models in inducing each of our resulting datasets for the 1883 semantic font attribute vectors, we use the leave-one-out cross-validation error procedure of [6] for each data induction model *m*, which is model invariant. See our Results and Discussion.

Results and Discussion

Our error metric for each of our experiments, per [3], is as follows:
- Train model *m* on $F_{161} \setminus f$ → predict *m*(f) for each f in F_{161} in turn → $error_f = a_s' - a_s$
- *e* = error vector with element-wise absolute value applied to $error_f$
- Average error vector $\bar{e}$ = element-wise average of all 161 *e* vectors
- Error for *m* = avg error over semantic attributes = avg of elements of $\bar{e}$

We visualize and present these errors in Figure 4 and in Table 1, along with our semantic attribute prediction experiment results.

Model	Error
Kulahcioglu KNN	0.084
Lowest Error KNN	0.080
Lowest Error SVR	0.079
Lowest Error Ensemble	0.076

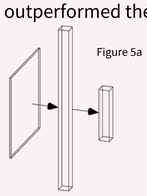
Table 1. Data Induction Error for highest-performing models of each category. Kulahcioglu KNN is from [3].

For dataset induction, the SVR model outperformed the *k*-NN models significantly. We believe this is because the RBF kernel projects the data into a higher-dimensional space to make the regression boundary more linearly characterizable. Moreover, it can exploit both magnitudes of the FontJoy embeddings as well as implicitly the angular distances between them (eqn. 1); conversely, *k*-NN can only leverage cosine distance, limiting its power. For attribute prediction, the CNN outperformed both linear baseline models. We believe this is because while the baselines cannot leverage spatial information since the images are linearized, the convolutional filters can characterize glyphs by learning the relationships between glyph features in a spatial sense, enabling it to "see" edges, curves, and other characteristics.

Semantic Attribute Prediction

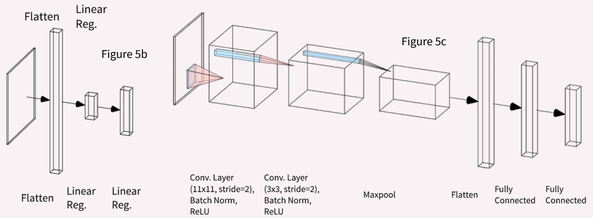
With the best dataset, we investigate how well models can predict semantic attributes given a typeface. We test two baseline models:
- 1-step LR: Linear regression model with rasterized image input of a collection of visually distinct glyphs à la Figure 1 (Figure 5a)
- 2-step LR: Predict typographic attribute vector from rasterized image input with linear regression; then predict semantic attributes using linear regression (Figure 5b)

We also test a convolutional neural network (CNN) with two layers, which outperformed these baselines (Figure 5c). See our results in Table 2.



Model	$RMSE_{tr}$	R^2_{tr}	$RMSE_{val}$	R^2_{val}
1-step LR	1.34753	0.99999	0.07974	0.71879
2-step LR	0.10327	0.50344	0.11761	0.41222
CNN	0.04263	0.88317	0.06434	0.81176

Table 2. Results from semantic attribute prediction experiments.



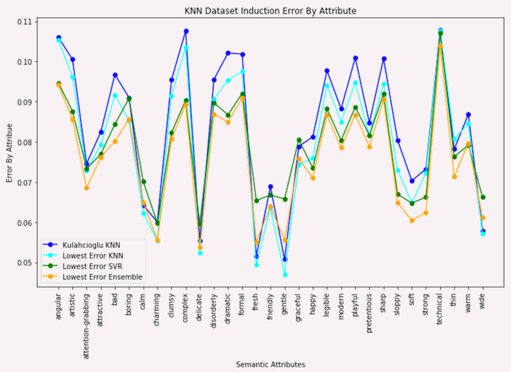


Figure 4. Dataset Induction Error of each model for each of the 31 semantic attributes. The ensemble performs best overall.

Future Work

We hope to further investigate clustering algorithms, such as spectral clustering, to induce lower-error datasets. In addition, we wish to create higher-performing models for rasterized and SVG image inputs for our downstream task.

References

[1] Avilés-Cruz, C., Villegas, J., Arechiga-Martínez, R., and Escarela-Perez, R. Unsupervised font clustering using stochastic version of the em algorithm and global texture analysis. In Sanfeliu, A., Martínez Trinidad, J. F., and Carrasco Ochoa, J. A. (eds.), *Progress in Pattern Recognition, Image Analysis and Applications*, pp. 275–286, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-30463-0.
[2] Azadi, S., Fisher, M., Kim, V. G., Wang, Z., Shechtman, E., and Darrell, T. Multi-content GAN for few-shot fontstyle transfer. *CoRR*, abs/1712.00516, 2017. URL http://arxiv.org/abs/1712.00516.
[3] Kulahcioglu, T. and de Melo, G. Predicting semantic signa-tures of fonts. pp. 115–122, 01 2018. doi: 10.1109/ICSC.2018.00025.
[4] Lin, X., Li, J., Zeng, H., and Ji, R. Font generation based on least squares conditional generative adversarial nets. Multimedia Tools Appl., 78(1):783–797, January 2019. ISSN 1380-7501. doi: 10.1007/s11042-017-5457-4. URL https://doi.org/10.1007/s11042-017-5457-4.
[5] Lopes, R. G., Ha, D., Eck, D., and Shlens, J. A learned representation for scalable vector graphics. *CoRR*, abs/1904.02632, 2019. URL http://arxiv.org/abs/1904.02632.
[6] O'Donovan, P., Libeks, J., Agarwala, A., and Hertzmann, A. Exploratory Font Selection Using Crowdsourced Attributes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 33(4), 2014.
[7] Oztürk, S., Sankur, B., and Abak, A. Font clustering and classification in document images. *European Signal Processing Conference*, 2015, 01 2000.