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 prediction 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 Bold’) in the semantic attribute dataset in [6]; FontJoy embeddings for 1883 typefaces to be utilized for dataset induction.
Task: induce corresponding 31-D semantic attribute vectors for all 1883 FontJoy typefaces.

Method: Semantic Attribute Induction

4-NN: As our baseline, we reproduce the optimal result in [3]. We cluster the F̂ for font embedding vectors and predict each âs 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, and 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.

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:

\[ K(x, x') = \exp(-\gamma ||x - x'||^2) \]  

We conduct exhaustive cross-validation grid search over hyperparameters \( \gamma \) 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. This model 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:
1. Train model \( m \) on \( F̂ \) \[ m \] predict \( m \) for each \( l \) in \( F̂ \) in turn \( \rightarrow \) error \( = \) \( \lVert â_s - a_s \rVert \)
2. \( e \) = error vector with element-wise absolute value applied to error
3. Average error vector \( \bar{e} \) = element-wise average of all 161 vectors

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

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 characteristic. 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. 1-step LR: Linear regression model with rasterized image input of a collection of visually distinct glyphs à la Figure 1 (Figure 5a)
2. 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.

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