CIS 520: Final Project Report

Sira Sriswasdi. PennKey: sirasris. Team: PeaceTS46

December 2nd, 2009

1 Blogs

1.1 Word and sentence characteristics

The first natural attempts we made are to calculate the following statistics in each blog:

- Sentence length distribution.
- Word length distribution.
- Non-dictionary word usage.

To split the blog into sentences, we label all words containing full stops, exclamation points, and question marks as "stop word" and then we use these to mark the end of each sentence (note that features like "....." or "!?!!!?" usually end a sentence too). For word length distribution, we remove the bias toward short-length common verbs, preposition, emoticons, and others by only keeping track of words with length at least 4. Finally, for non-dictionary usage, we count only the words which contain no non-letter characters and doesn't exists in dictionary. We make sure that informal words such as "gotta" or "coz" and rude words do not appear in our downloaded dictionary.

However, despite some publications suggesting that these features are indicative of the blog writer's gender and/or age. We found little to no correlation between them and the writer's information (< 0.01 for sentence and word lengths and around 0.35 for non-dictionary word usage).

1.2 Poisson-related models

Since Poisson distribution models the number of occurrences for some event with fixed rate of happenning λ, we find it quite naturally that the number of usages for word W by writers of gender group g or age group a should be Poisson distributed with rate $\lambda(W, g, a)$ depending on the word, gender, and age. Note that $g \in \{1, 2\}$ and $a \in \{1, 2, 3\}$.

Subsequently, for gender or age predictions, we compute the corresponding maximum likelihood estimates $\hat{\lambda}(W, g), \hat{\lambda}(W, a)$ being the mean usage of word W by writers from gender group g or age group a. We also correct for the case where $\hat{\lambda}(W, g) = 0$ or $\hat{\lambda}(W, a) = 0$ by adding 1's to both the numerator and denominator when computing the mean usage.

The resulting model has a very low training error (more than 0.998 accuracy on the training set) but perform very poorly on 10-fold crossvalidation (0.35 accuracy for age prediction and 0.53 for gender). The reason was because the set of rare words (used < 5 times overall and usually used in only some and not all of the writer groups), which is actually the majority of our feature space, dominates the overall probability calculation (Poisson distribution with low *lambda* is almost equivalent to a dirac delta function $\delta(0)$).

When trained over all word features, merely the presence/absence of these rare words can pinpoint both the gender and age groups. However, in each round of the 10-fold crossvalidation, we lose as much as 11,000 features (which tend to be rare words) and the perfect performance is lost. The fact that the accuracy drops to almost the same level as any random predictions suggests that the simple Poisson model does not really explain the data.

We make another quick attempt at modeling word usage as a Poisson mixture with two components based on the following assumption:

Each writer from gender group g or age group a has a probability $p(g, a)$ of writing a blog with content related to word W in which W should appear with rate $\lambda_1(W, g, a) > 1$ and a probability of $1 - p(g, a)$ of writing a blog unrelated to word W in which W should appear $\lambda_0(W, g, a) << \lambda_1(W, g, a)$.

To estimate the probabilities $p(g, a)$ and rates $\lambda_i(W, g, a)$'s, we use EM algorithm (downloaded from MATLAB central). The performance slightly improves (0.47 accuracy for age prediction and 0.6 for gender) but still could not beat the baseline model.

1.3 Feature space reduction

Next, we turn our focus to improving the quality of feature space by noticing that some words are actually equivalent in their content/intention. Specifically, we did the followings:

- Group all emoticons and expression symbols (such as ".....", "!?!!!?", and "argghhhh").
- Group all CAPITALIZED words (they tend to be used by younger writers).
- Group informal words (such as "gotta" and "kinda").
- Group possible typos (defined as words of length > 4, not in dictionary, and differ from some dictionary word by at most 2 hamming distance).
- De-stem all the words that exist in dictionary (PorterStemmer).
- Discard the remainings (to remove the effect of rare words we found earlier).

This reduces the feature space size from 89K to roughly 20K. The predictors trained by Poisson mixture model and Naive Bayes model on this reduced space perform roughly the same or slightly better at times compared to the baseline result (0.6 - 0.7 accuracy for both gender and age predictions on a 10-fold crossvalidation).

1.4 An attempt at Gaussian models

As a side note, we quickly tried Gaussian mixture model with two components (mimic the rationale of the Poisson mixture model but replace the explanation of word usage by Normal distributions - truncated at zero). The performance on 10-fold validation is the similar to Poisson mixture model which is troubling at first because the two models assume different natures of word usage in blog writing. However, with more detailed inspections, we found out that the shape of Poisson distribution with high λ closely resembles Normal distribution and for very low λ, it resembles truncated Normal.

2 Faces

2.1 Finding the positions of eyes, nose, and mouth

In order to calculate the proportion between different facial features, our first task is to locate the positions of the eyes, nose, and mouth. Here is the rough flowchart of the algorithm:

- Crop for the face by imcrop([80 90 95 88]) - this happens to be the way all images are positioned.
- Convert the image to binary by using auto threshold $-$ $0.7\times$ graythresh() to be exact.
- Only eyes and eye brows are dark enough to always pass this filter (nostrils and mouth often do not).
- Using peak detection and spline smoothing on the horizontal sum of binary pixels, we can locate the y coordinate for the eyes region, called eye_y.
- Crop again with full width around $\text{eye}_y \pm \delta_y$ to focus on the eyes.

- Using threshold conversion and then peak detection and spline smoothing on the vertical sum of pixels, we can locate the x coordinates for both eyes, called eye$_x$'s.
- The mouth's y coordinate is found by converting the bottom half of the image to edges via Sobel operator, computing horizontal sum of pixels, and then detecting the peak.

Note that the nostrils often disappear during the sobel edge finding operation or do not produce a strong enough peak to be detected (compared to mouth region).

2.2 Predicting gender

To predict the gender, we use three features: the distance between two eyes, the distance between eye level and mouth level, and the proportion between the two. Since we have 4 images for each single person, the data is collapse and the geometric mean of the features are used (geometric mean gives better prediction performance than arithmetic mean).

The SVM model is trained using MATLAB's bioinformatics toolbox with radial basis kernel and its default parameters (the prediction accuracy are pretty much invariant to small perturbation in the parameters). The overall accuracy based on 10-fold crossvalidation is variable between 0.75 and 0.93.

2.3 LBP feature

Local Binary Patterns (LBP) feature is computed by using each pixel to threshold its neighbors and is descriptive of the region's texture and smoothness. When applied to facial recognition, non-uniform LBP can indicate wrinkles or other marks occurred due to age while uniform LBP represents smoothness. Specifically, we have

$$LBP_{P,R}(X_c) = \sum_{p=0}^{P-1} u(x_p - x_c)2^p$$

where $u(y) = \mathrm{I}\{y \geq 0\}$ is the step function, x_c is the center pixel, P is the number of its neighbors to be considered on a circle of radius R away from the center pixel.

2.4 Predicting age

To generate features for age prediction, we first split the face images into 5 regions (whole-face, forehead, left cheek, right cheek, and chin ares) according to the values for eye$_y$, eye$_x$'s, and mouth$_y$ we obtained previously. Then, for each region, we compute the LBP profile using $P = 8$, $R = 1$ which is equivalent to using the 8 adjacent neighbors of the center pixel. The resulting profile is binned to 59 bins with the first 58 bins for each of the uniform pattern and the last bin containing all non-uniform ones.

Finally, we implement a nearest neighbor approach with majority votes. For each new image, we generate 5 vectors of size 59 corressponing to the 5 regions. Then each region's profile is compared to the mean profiles (from each age group) of that region in our training set, and get 1 vote from whichever age group the nearest mean profile is from. In the end, we will have 5 votes for 3 different age groups and one of the winners is picked randomly. This results in a variable performance on 10-fold crossvalidation ranging from 0.38 to 0.9 accuracy.