Machine Learning Project: machine learning part

Michael

1 Features Construction

As it has been discussed in the last section, the task can be formatted as the standard classification problem, which regards the kinship as one class and the nonkinship as other. The binary classification on the kinship in the dataset can only be implemented when the dimensions of features are less than the number of observations. Since we have 3598 pairs of kinship in the dataset, the features constructed from the dataset has to be less than 3598. Without employing too much computing power, the feature for each picture is constructed as follows:

- Step 1: for picture $p_i = [224, 224, 3]$, take the first two dimensions and let the transformed picture $x_i = [224, 224]$;
- Step 2: let $w = x_i x_i^T$, and then take the eigenvalues of w, and let $f_i =$ `eigenvalues(w)`;
- Step 3: for each pair of kinship, calculate the element-wise differences of features:

$$d_p = f_{p1} - f_{p2}$$

- Step 4: among all pictures, randomly match the new pairs and make sure those pairs are not kinship, then calculate the element-wise differences of features by repeating step 1 to 3.

2 Properties of Features

With 3598 pairs of kinships and 3598 pairs of nonkinships, we can examine the properties of features before using machine learning models to do classification. Figure 2.1 presents the comparison of features for kinship and nonkinship.

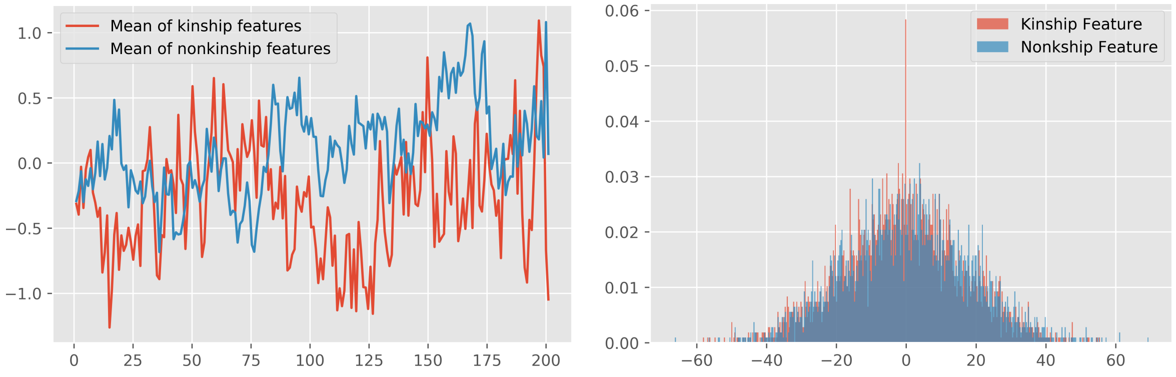

Figure 2.1: Comparison of Kinship and Nonkinship

According to Figure 2.1, certain amount of feature differences exist between kinship and nonkinship pairs. However, the difference is not big enough to allow the model to detect the boundaries for classification. What's more, the distributions of selected features for kinship and nonkinship are homogenous. This means that the models by using Gaussian densities will not work well on those features.

When it comes to the distances of features differences, figure 2.2 gives the comparison. Again, the distributions of the distances[1] are very similar to each other for kinship pairs and nonkinship pairs. This brings the challenges for the methods employing the distance measurements, such as KNN, etc.

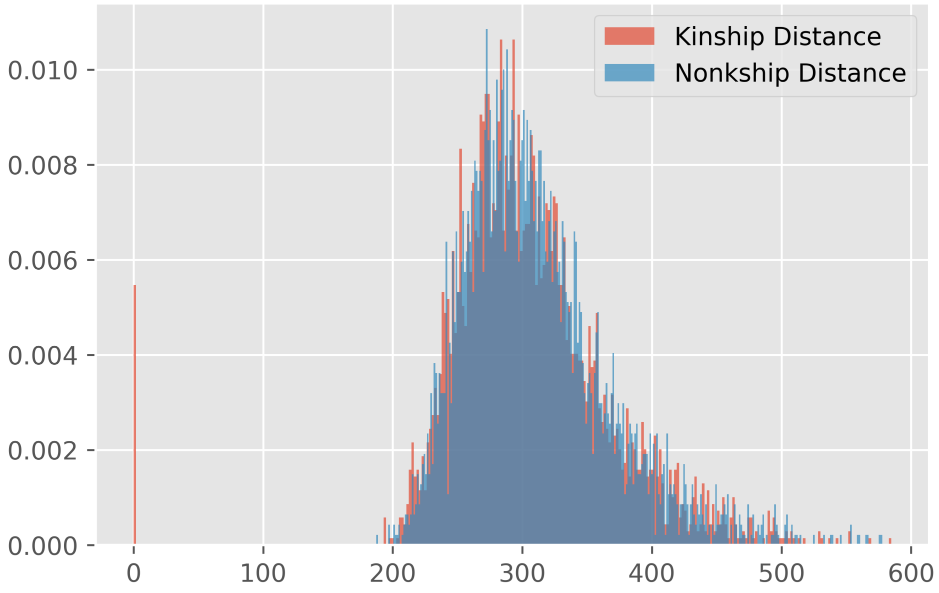

Figure 2.2: Histogram of Distances

3 Review and Results of Machine Learning Models

Before presenting the results, it is worth to review the main models in machine learning field. In our project, we have employed linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), logistic regression, naive Bayes classification, k-nearest-neighbor classification (KNN), support vector machines (SVM) and neutral network methods.

For LDA, QDA and logistic regression, they all employe the Gaussian densities. Suppose we model each class density as multivariate Gaussian

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp[-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)], \tag{3.1}$$

where LDA assumes that the classes have a common covariance matrix $\Sigma_k = \Sigma \forall k$, and the observations are classified by using the following function

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log \pi_k \tag{3.2}$$

[1] The distance is measured by Euclidean distance.

However, QDA assumes that each class has its own covariance matrix Σ_k, and the observations are classified by using the following function

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k \tag{3.3}$$

The logistic regression uses the continuous function (or map) between covariates x and dependent variable probability π:

$$\pi = h_\theta(x) = \frac{e^{\theta' x}}{1+e^{\theta' x}} = \frac{1}{1+e^{-\theta' x}} \tag{3.4}$$

where the general format

$$g(z) = \frac{1}{1+e^{-z}}$$

is called the logistic function. With this function, the log-likelihood function was employed to find the coefficients:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} P(y^i|x^i;\theta) \\ &= \prod_{i=1}^{m} (h_\theta(x))^y (1-h_\theta(x))^{1-y} \end{aligned}$$

The naive Bayes classification has the same structure with LDA and QDA, which means it also assumes the certain density function behind the model.

The k-nearest-neighbor methods employs the Euclidean distance in feature space:

$$d_i = ||x_i - x_0||,$$

where the distance is used to search for the nearest neighbors. The classification is made by following the majority rule.

Table 3.1 gives the results from different machine learning methods. The comparison of those results can show why some methods outperform others. For instance, LDA wins out over the QDA as the classes in the dataset has the common covariance matrix. This means the QDA will not work well as it assumes the classes should have separate covariance matrixes.

Table 3.1: Results of Different ML Methods

Methods	Accuracy	True Positive	AUC	Explanations
LDA	54.28%	55.09%	54.28%	Common covariance
QDA	51.69%	52.61%	51.68%	Separate covariance for each class
Logistic	54.47%	55.25%	54.46%	Log Likelihood
Bayes	52.43%	62.97%	51.83%	Diagonal variance
KNN	53.41%	66.54%	52.39%	Neighbors = 3
SVM	56.55%	59.85%	56.49%	scale the input
Neural NW	54.18%	65.08%	53.47%	layer size = 12, 6

References