

Large-Scale Image Pattern Recognition Parallel Machine Learning Implementation

(According to first name's initial's order in English vocabulary)

Haitang Hu, Huizhan Lv, Jian Jin, Tianyi Chen

December 1, 2014

Abstract

Image recognition requires high workload because of its intrinsic complex feature space. When it comes to large amount of pictures, common machine learning algorithm shows obvious limitation in pattern recognition in terms of computation efficiency, resource occupation, etc. Thus techniques developed specifically to deal with large-scale data processing could be expected to have obviously superior performance. In this project we implement parallel SVM based on PCA in large-scale image pattern recognition and verify this scheme works in such kind of task with favourable performance as we have expected.

1 Background and Motivation

Pattern recognition concerning image has the mark of complex feature space. Consider a 32×32 color picture, each picture will have 3072 features take RGB scheme into account. For a dataset of 60,000 pictures, it is quite a huge load for the classification task.

However the ability to deal with classification in image recognition is of quite significance. For instance, when we look for images by inputting keyword "car" through a search engine, clearly it should distinguish between distinct types of objects and we don't expect a picture of a boat in the search result.

We aim to develop an algorithm which makes classification of large-scale image dataset in an efficient manner, that is to say, gives out result quickly with a high level of accuracy.

2 Approach

2.1 Preprocessing

We use PCA for preprocessing. In PCA, the goal is to project the data \mathbf{x} with dimensionality D onto a space having dimensionality $M < D$ while maximizing the variance of the projected data.

Suppose S is the data covariance matrix defined by

$$S = Cov(\mathbf{x}) \quad (1)$$

Let $\{z_i\}$ represent M linear combinations of our original D predictors:

$$z_m = \mathbf{w}_m^T \mathbf{x} = \sum_{j=1}^D w_{jm} x_j, \quad m = 1, 2, \dots, M \quad (2)$$

We now maximize the projected variance

$$Var(z_1) = \mathbf{w}_1^T S \mathbf{w}_1 \quad (3)$$

Constraint condition is $\mathbf{w}_1^T \mathbf{w}_1 = 1$. Make an unconstrained maximization of

$$w_1^T S w_1 - \lambda_1 (w_1^T w_1 - 1). \quad (4)$$

where λ_1 is a Lagrange multiplier. The optimization result is

$$S \mathbf{w}_1 = \lambda_1 \mathbf{w}_1 \quad (5)$$

So

$$w_1^T S w_1 = \lambda_1 \quad (6)$$

which reveals that the variance will be a maximum when we set \mathbf{w}_1 equal to the eigenvector having the largest eigenvalue λ_1 . This eigenvector \mathbf{w}_1 is the first principal component.

The second principal component \mathbf{w}_2 should also maximize variance, be of unit length, and be orthogonal to w_1 . We could find that \mathbf{w}_2 should be the eigenvector of S with the second largest eigenvalue. Similarly, we can show that the other dimensions are given by the eigenvectors with decreasing eigenvalues.

Since there are too many features in the task, we use PCA as preprocessing, which reduces 3072 features to 100 features. We projected all the features on the first 100 eigenvectors thus acquiring the data for classification.

2.2 Processing

We use parallel SVM with Gaussian kernel for classification, which runs on four separate machines. The standard SVMs are binary classifiers using a linear decision boundary with discriminant function

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x}^t + w_0 \quad (7)$$

of which the decision function is

$$y_{new} = \text{sign}(\mathbf{w}^T \mathbf{x}_i + w_0) \quad (8)$$

In our tasks there is no guarantee that the dataset is linear separable, so we introduce kernel function which allow non-linear decision boundaries.

Kernel is based on transformation of original features, define the basis functions

$$\mathbf{z} = \phi(\mathbf{x}) \text{ where } z_j = \phi_j(\mathbf{x}), j = 1, \dots, k \quad (9)$$

mapping from the d -dimensional \mathbf{x} space to the k -dimensional \mathbf{z} space where we write the discriminant as

$$g(\mathbf{z}) = \mathbf{w}^T \mathbf{z} \quad (10)$$

$$g(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \sum_{j=1}^k w_j \phi_j(\mathbf{x}) \quad (11)$$

The dual in standard SVM problem is now

$$L_d = \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \phi(x_i)^T \phi(x_j) \quad (12)$$

subject to

$$\sum_i \lambda_i y_i = 0, \quad 0 \leq \lambda_i \leq C, \quad \forall i \quad (13)$$

The idea in kernel machines is to replace the inner product of basis functions, $\phi(x_i)^T \phi(x_j)$, by a kernel function, $K(x_i, x_j)$, thus the kernel function also shows up in the discriminant

$$g(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \left[\sum_{i=1}^k \lambda_i y_i \phi(\mathbf{x}_i)^T \right] \phi(\mathbf{x}) = \sum_{j=1}^k \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) \quad (14)$$

Use this kernel function, we do not need to map it to the new space at all. There are multiple types of kernels and we use Gaussian kernel

$$K(x, x') = \exp\left(-\frac{1}{2}(x - x')^T \Sigma^{-1}(x - x')\right) \quad (15)$$

3 Framework and Implementaion

3.1 Framework-Sparks

3.2 Implementations

4 Result and evaluation

5 Conclusion

Image recognition requires high workload because of its intrinsic complex feature space. When it comes to large amount of pictures, common machine learning algorithm shows obvious limitation in pattern recognition in terms of computation efficiency, resource occupation, etc. Thus techniques developed specifically to deal with large-scale data processing could be expected to have obviously superior performance. In this project we implement parallel SVM based on PCA in large-scale image pattern recognition and verify this scheme works in such kind of task with favourable performance as we have expected.