Data Mining: Learning from Large Data Sets - Fall Semester 2015

luchen@student.ethz.ch
myhshirley@student.ethz.ch
zhhan@student.ethz.ch

November 7, 2015

Large Scale Image Classification

Gerenal Procedure

The task is to train a svm model for image classification. On-line method with Map-Reduce is used to learn large data set. The general procedure is as follows:

- **mapper** Each mapper first transforms original data from original feature space to Random Fourier Feature space. Then, it uses Pegasos [2] with AdaGrad [1] to learn the optimal ω from transformed data. Finally, $mapper_i$ outputs its optimal ω_i to reducer.
- **reducer** The reducer simply averages outputs from all mappers $\{\omega_1, \omega_2, \ldots, \omega_n\}$ (n is the number of mappers) and output the final ω, $\omega = \sum_{i=1}^{n} \omega_i$.

Detailed methods are discussed in the following sections.

Mapper – training model on each part of data

We first applys nonlinear transformation to data. Then we applys linear classifier to the transformed data. These steps are done in mapper:

1. **Feature Transform** We applied Random Fourier Features of Laplacian kernel to original data. Dimension of new feature space m is 6000. The Random Fourier Features are generated in following way:
   1.1. Generate m independent multivariate variables $\{\omega_1, \omega_2, \ldots, \omega_m\}$, the length of ω_i ($i \in [1, m]$) equals to the dimension of original features (400). Items ω_{ij} ($j \in [1, 400]$) in ω_i are independently sampled from Cauchy distribution, $P(\omega_{ij}) = \frac{1}{\pi\gamma[1+(\omega_{ij}/\gamma)^2]}$ (we set γ to 0.7).
   1.2. Generate m independent variables $\{b_1, b_2, \ldots, b_m\}$ from uniform distribution $[0, 1]$.
   1.3. For each data point x_i, transform it to new feature space $z(x_i) = \big(z_{\omega_1,b_1}(x_i), \ldots, z_{\omega_m,b_m}(x_i)\big)/\sqrt{m}$, $z_{\omega_i,b_i}(x_i) = \sqrt{2}\cos(\omega_i^T x + b_i)$

2. **Model Training** We apply Pegasos [2] algorithm with AdaGrad [1] method on the transformed data $\{z(x_1), z(x_2), \ldots, z(x_N)\}$ and get the optimal linear SVM model. The method works as follows:

   2.1. Read the current training data $(z(x_t), y_t)$ (y_t is the label of x_t). If $(z(x_t), y_t)$ is correctly classified by current ω_t ($\omega_t^T z(x_t) y_t \geq 1$), do nothing. Else if $\omega_t^T z(x_t) y_t \leq 1$, update ω_t to ω_{t+1} by following rule:

   2.2. **Updating rule for** ω_t: First compute the gradient $\nabla_t = \lambda * \omega_t - y_t * z(x_t)$ (we set λ to 0.00002) and update the diagonal matrix G_t in AdaGrad algorithm. Then ω_{t+1} is computed in this way: $\omega_{t+1} = \text{argmin}_{\omega \in S} \|w - (w_t - \eta_t G_t^{-1} \nabla_t)\|$

3. Iterate above steps until no data left.

4. Send the optimal ω to reducer.

We have also tried Random Fourier Features of Gaussian kernel with different parameters. However, result of Laplacian kernel is better on this data set.

Reducer – averaging results of mappers

The reducer collect all $\{\omega_1, \omega_2, \ldots, \omega_n\}$ from mappers and simply average them as the final result. $\omega = \frac{1}{n} \sum_{i=1}^{n} \omega_i$ (n equals to the number of mappers).

Contribution

Each of us worked out a solution and we submitted the one with highest score. We have also tried different kernels with different parameters. However, Random Fourier Features of Laplacian kernel with parameters setted as indicated above achieved our best result.

References

[1] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[2] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.