Supplement to Multi-View Multi-Class Classification for Identification of Pathogenic Bacterial Strains

Evgeni Tsivtsivadze[1], Tom Heskes[2], and Armand Paauw[1]

[1] MSB Group, The Netherlands Organization for Applied Scientific Research, Zeist, The Netherlands
firstname.lastname@tno.nl
[2] Institute for Computing and Information Sciences, Radboud University, The Netherlands
firstname.lastname@science.ru.nl

Efficient kernel matrix approximation.

When working with large datasets, the cubic runtime for obtaining a solution of the co-regularized algorithm as well as the quadratic space consumption for storing the kernel matrix form bottlenecks. We demonstrate a way to deal with these drawbacks.

One approach to speed up the algorithm consists in approximating the kernel matrix. In our optimization problem the first term corresponds to the loss evaluated over all training examples while the second and third term are the regularizers penalizing "complex" and "too different" solutions. Let $R = \{i_1, \ldots, i_r\} \subseteq [n]$ be a subset of indices such that only $\mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_r}$ are nonzero for each view. Thus, by randomly selecting a subset of data points, we can write the objective function as

$$
\begin{aligned}
&J(\hat{\mathbf{A}}) = \\
&\quad \sum_{v=1}^{M} \mathrm{tr}\Big(Y - K^{(v)}_{\cdot,R} A^{(v)}_{R,\cdot}\Big)^t \Big(Y - K^{(v)}_{\cdot,R} A^{(v)}_{R,\cdot}\Big) + \\
&\quad \lambda \sum_{v=1}^{M} \mathrm{tr} A^{(v)t}_{R,\cdot} K^{(v)}_{R,R} A^{(v)}_{R,\cdot} + \\
&\quad \nu \sum_{v,u=1}^{M} \mathrm{tr}\Big(K^{(v)}_{\cdot,R} A^{(v)}_{R,\cdot} - K^{(u)}_{\cdot,R} A^{(u)}_{R,\cdot}\Big)^t \Big(K^{(v)}_{\cdot,R} A^{(v)}_{R,\cdot} - K^{(u)}_{\cdot,R} A^{(u)}_{R,\cdot}\Big),
\end{aligned}
$$

where $A^{(v)}_{R,\cdot} = (\mathbf{a}^{(v)}_{i_1}, \ldots, \mathbf{a}^{(v)}_{i_r})^t \in \mathbb{R}^{r \times p}$, and $\hat{\mathbf{A}} = (A^{(1)t}_{R,\cdot}, \ldots, A^{(M)t}_{R,\cdot})^t \in \mathbb{R}^{Mr \times p}$. Note that, by applying this approximation scheme, the loss is still evaluated over all training examples but the kernel matrix and prediction function are constructed using a subset of data points leading to the benefits in a runtime and space reduction. The approach for this matrix approximation, known as

"subset of regressors", was pioneered (for a single view methods) in [1] and is frequently applied in practice. Although it may seem over-simplistic (e.g. other methods such a matching pursuit [2] might be more suitable rather than random selection of the regressors) it appears to be particularly useful in the context of co-regularization with multiple views. To clarify, consider the predictions for the training data points using a single view in our algorithm, namely $K_{\cdot,R}A_{R,\cdot}$. Using the Woodbury matrix identity [3] and by defining $\hat{K} = \frac{1}{\lambda}K_{\cdot,R}K_{R,R}^{-1}K_{\cdot,R}^t$, we can reformulate this expression as follows:

$$
\begin{aligned}
&K_{\cdot,R}A_R = \\
&K_{\cdot,R}(K_{\cdot,R}^t K_{\cdot,R} + \lambda K_{R,R})^{-1}K_{\cdot,R}^t Y = \\
&K_{\cdot,R}(\frac{1}{\lambda}K_{R,R}^{-1} - \frac{1}{\lambda}K_{R,R}^{-1}K_{\cdot,R}^t(\frac{1}{\lambda}K_{\cdot,R}K_{R,R}^{-1}K_{\cdot,R}^t + I)^{-1} \\
&\frac{1}{\lambda}K_{\cdot,R}K_{R,R}^{-1})K_{\cdot,R}^t Y = \\
&(\hat{K} - \hat{K}(\hat{K} + I)^{-1}\hat{K})Y = \\
&(\hat{K}(I - (\hat{K} + I)^{-1}\hat{K})Y = \\
&(\hat{K}((\hat{K} + I)^{-1}(\hat{K} + I) - (\hat{K} + I)^{-1}\hat{K})Y = \\
&\hat{K}(\hat{K} + I)^{-1}(\hat{K} + I - \hat{K})Y = \\
&\hat{K}(\hat{K} + I)^{-1}Y.
\end{aligned}
$$

The last term can be rewritten as $\hat{K}(\hat{K} + I)^{-1}Y = \bar{K}(\bar{K} + \lambda I)^{-1}Y$, where $\bar{K} = K_{\cdot,R}K^{-1}K_{\cdot,R}^t \in \mathbb{R}^{n\times n}$. The above calculation demonstrates that the solution obtained by using the kernel matrix approximated by a subset of regressors corresponds to the one obtained by the not approximated kernel matrix $\bar{K}$. By considering different (non-overlapping) subsets of regressors for different views one can also obtain feature spaces that can be used in the co-regularization approach. To conclude, the subset of regressors method can serve not only to speed up our algorithm, but in principle can also be used for constructing different views.

Regularization parameter estimation.

By rewriting D as $D = \lambda\acute{D}$ in the equation (4) with (positive definite) matrix $\acute{D}$ and using the Cholesky decomposition [3] $\acute{D} = PP^t$, we obtain

$$
\begin{aligned}
&(B + C + D)^{-1} = \\
&(B + C + \lambda\acute{D})^{-1} = \\
&(PP^{-1}(B + C)(P^t)^{-1}P^t + \lambda PP^t)^{-1} = \\
&(P^t)^{-1}(P^{-1}(B + C)(P^t)^{-1} + \lambda I)^{-1}P^{-1}.
\end{aligned}
$$

The matrix $P^{-1}(B+C)(P^t)^{-1}$ can be eigen decomposed to $V\Lambda V^t$, where Λ is a diagonal matrix containing the eigenvalues and V is the matrix composed of the eigenvectors [3]. Hence, we get

$$\begin{aligned}(B+C+D)^{-1} &= (P^t)^{-1}(V\Lambda V^t + \lambda I)^{-1}P^{-1} \\ &= (P^t)^{-1}V(\Lambda + \lambda I)^{-1}V^t P^{-1}\end{aligned}$$

and the solution in (4) can be rewritten as

$$\mathbf{A} = (P^t)^{-1}V(\Lambda + \lambda I)^{-1}V^t P^{-1}E.$$

Thus, by fixing the parameter ν, we can efficiently search for the second regularization parameter λ. The decompositions and the inversion of P can be calculated in $\mathcal{O}(M^3n^3)$ time, and hence, training complexity of the algorithm is not increased.

References

1. Poggio, T., Girosi, F.: Networks for approximation and learning. In: Proceedings of the IEEE. Volume 78(9). (1990) 1481–1497
2. Vincent, P., Bengio, Y.: Kernel matching pursuit. Machine Learning **48**(1-3) (2002) 165–187
3. Golub, G.H., Loan, C.F.V.: Matrix Computations. Johns Hopkins University Press (1996)