ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

A Multi-kernel Framework for Inductive Semi-supervised Learning

Xilan TIAN, Gilles GASSO and Stéphane CANU

LITIS EA4108 - INSA de Rouen
St Etienne du Rouvray - France

Abstract. We investigate the benefit of combining both cluster assumption and manifold assumption underlying most of the semi-supervised algorithms using the flexibility and the efficiency of multi-kernel learning. The multiple kernel version of Transductive SVM (a cluster assumption based approach) is proposed and it is solved based on DC (Difference of Convex functions) programming. Promising results on benchmark data sets suggesting the effectiveness of proposed work.

1 Introduction

Most of semi-supervised algorithms rely on two assumptions: (1) the cluster assumption believes in that points of the same cluster tend to be of the same class, which leads to low density separation strategy [1]; (2) the manifold assumption supports that the (high-dimensional) data lie (roughly) on a low-dimensional manifold, which results in perfect solution of ”curse of dimensionality” [2]. Therefore, this raises the question of ”how much benefit one can get by combining the two assumptions” and represents a worth pursuiting task. In the kernel framework, Xu et al. [3] implemented the aforementioned combination from perspective of the regularization strength induced by the unlabeled data. Dai and Yeung [4] proposed an integrated regularization framework from perspective of kernel selection. However, the pool of kernels considered for selection is obtained via the basic kernels or from the pseudo-inverse of the Laplacian matrices built by labeled and unlabeled data. Those algorithms solely provide predicted labels of unlabeled samples and need an entire retrain to handle new samples.

In this paper, we pursue the goal of combining the cluster and manifold assumptions to yield an inductive model. For this sake, we resort to transductive SVM (TSVM) whose decision function lies in the span of functions $\kappa(x_i, .)$ (where κ is a kernel and x a sample) and hence can perform well in the out-of-sample case. This flexibility also permits to adapt multiple kernel framework to our problem by including kernels built following the cluster assumption as well as the manifold preserving constraints. Compared with the complex optimization procedures involved in [3] and [4], our TSVM-MKL (transductive multi-kernel SVM) benefits from the scalability of TSVM [5] and the efficiency of fully supervised multi-kernel SVM solvers [6, 7]. Experimental results show its effectiveness.

2 Transductive SVM

We first describe the notations used in this paper. Let $\mathcal{D} = \{x_1, \dots, x_n\}$ denote the entire data set. The first l samples are labeled $\mathcal{D}_\mathcal{L} = \{(x_i, y_i) \in \mathcal{X} \times \{-1, 1\}\}$ and followed by u unlabeled samples $\mathcal{D}_\mathcal{U} = \{x_i \in \mathcal{X}\}$. The unkown labels of $\mathcal{D}_\mathcal{U}$ are denoted as $y_\mathcal{U} = [y_{l+1}, \dots, y_{l+u}]^T$. TSVM leverages the unlabeled data and solves the following optimization:

$$\min_{f,b} \frac{1}{2}\|f\|^2_\mathcal{H} + C\sum_{i=1}^{l} V_\mathcal{L}(y_i, g(x_i)) + C^* \sum_{i=l+1}^{l+u} V_\mathcal{U}(|g(x_i)|) \tag{1}$$

where the loss over labeled data $V_\mathcal{L}$ and unlabeled data $V_\mathcal{U}$ is weighted by C and C^*, which reflect confidence in labels and in the cluster assumption respectively. The decision function is defined as $g(x) = f(x) + b$, where f is a function in a Reproducing Kernel Hilbert Space (RKHS). To avoid the situation that all unlabeled data are assigned to the same class, a balancing constraint is added: $\frac{1}{u}\sum_{i=l+1}^{l+u} g(x_i) = \frac{1}{l}\sum_{i=1}^{l} y_i$.

We adopt the Hinge loss ($H_s(z) = \max(0, s - z)$, where $0 \leq s \leq 1$) on $V_\mathcal{L}$. $V_\mathcal{U}(|z|) = R_s(z) + R_s(-z) - (1 - s)$ is employed for the unlabeled data. R_s is the Ramp loss defined as $R_s(z) = H_1(z) - H_s(z)$ (see [5] for more details). Solving (1) with previous loss functions is equivalent to solve a classical SVM with labeled data and also unlabeled data that counted twice with artificial labels $\{-1, 1\}$ [5]. That means $y_i = 1$ when $l + 1 \leq i \leq l + u$, and $y_i = -1$ when $l + u + 1 \leq i \leq l + 2u$. In the next section, we formulate a multi-kernel version of TSVM whose solution requires the full description of DC programming.

3 Multiple kernel TSVM

Multiple kernel learning is a way to incorporate information from different sources to tackle a learning problem in kernel machinery framework. Numerous efficient methods were proprosed recently [6, 7]. For our TSVM problem, we consider the approach proposed in [6] whose formal setup is given by:

$$\min_{f_k,b,\mathbf{d}} \frac{1}{2}\sum_{k=1}^{m} \frac{a_k}{d_k}\|f_k\|^2_{\mathcal{H}_k} + C\sum_{i=1}^{l} H_1(y_i g(x_i)) + C^* \sum_{i=l+1}^{l+2u} R_s(y_i g(x_i)) \tag{2}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} d_k = 1, \quad \frac{1}{u}\sum_{i=l+1}^{l+u} g(x_i) = \frac{1}{l}\sum_{i=1}^{l} y_i, \quad d_k \geq 0 \quad \forall k = 1, \dots, m.$$

The decision function is defined as: $g(x) = \sum_{k=1}^{m} f_k(x) + b$, where f_k are defined over different RHKSs induced by different kernels $\kappa_k(\cdot, \cdot)$. The vector $\mathbf{d}$ with entries d_k $(1 \leq k \leq m)$ acts as the selector of appropriate kernels. a_k is a normalization term, usually set as the trace of the kernel matrix induced by κ_k.

3.1 Solving the TSVM-MKL problem

SVM-MKL inherits the non-convexity of TSVM which is related to Ramp loss. We employ DC programming to circumvent this shortcoming: first decomposes the non-convex function as the difference of two convex functions, and then approximate the concave part by its affine minorization iteratively. The following decomposition of (2) are attained by decomposing the Ramp loss:

$$\begin{aligned} J_1(\theta) &= \frac{1}{2}\sum_{k=1}^{m}\frac{a_k}{d_k}\|f_k\|^2_{\mathcal{H}_k} + C\sum_{i=1}^{l} H_1(y_i g(x_i)) + C^*\sum_{i=l+1}^{l+2u} H_1(y_i g(x_i)) \\ J_2(\theta) &= C^*\sum_{i=l+1}^{l+2u} H_s(y_i g(x_i)) \end{aligned}$$

Parameter vector θ comprises of f_k $(1 \leq k \leq m)$, bias term b and coefficients d_k. From this point we can write: $\langle \theta, \nabla_\theta J_2(\theta^t)\rangle = C^* \sum_{i=l+1}^{l+2u} \langle \theta, \nabla_\theta H_s(y_i g^t(x_i))\rangle$ where $\nabla_\theta H_s(y_i g^t(x_i))$ is the derivative taken at the current decision function $g^t(x)$. As $J_2(\theta)$ is independent of d_k, it should suffice to calculate $\langle \theta, \nabla_\theta H_s(yg^t(x))\rangle$ which involves terms relative to f_k and the bias b. Recalling definition of $H_s(z)$ and using reproducing property of Hilbert space $(f(x) = \langle f, k(x,\cdot)\rangle_{\mathcal{H}_k})$, we obtain the relations: $\nabla_b H_s(yg^t(x)) = \nu y$ and $\nabla_{f_k} H_s(yg^t(x)) = \nu y \kappa_k(x,\cdot)$. The scalar ν is the gradient $\partial H(z)$ at $z = yg^t(x)$:

$$\nu = \begin{cases} -1 & \text{if } \; yg^t(x) < s \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

It is worth mentionning that Hinge loss function is differentiable everywhere except in $z = s$. To be consistent, we should consider the subgradient. However, following [5] we arbitrary set $\nu = 0$ at $z = s$. Gathering all those informations, the affine approximation takes the form: $\langle \theta, \nabla_\theta H_s(yg^t(x))\rangle = \nu yb + \nu y \sum_{k=1}^{m} f_k(x) = \nu y g(x)$ and hence: $\langle \theta, \nabla_\theta J_2(\theta^t)\rangle = C^* \sum_{i=l+1}^{l+2u} \nu_i y_i g(x_i)$.

With all these elements, the application of DC programming to TSVM-MKL leads to Algorithm 1. One can notice that this problem simply turns out to solve iteratively a fully supervised multiple kernel SVM with additionnal balancing constraint which does not harm the solution. So we can benefit from any efficient off-the-shelf MKL solver as those presented in [6, 7].

3.2 Solving each iteration of TSVM-MKL

For completeness sake, we present in the sequel a adaptation of SimpleMKL [6] to handle (4). Natively, the approach is iterative and can be summarized as follows: assume d_k is fixed, (4) turns to be a classical SVM problem. Let $\tilde{J}(\mathbf{d})$ is the minimum according to f_k and b of (4) which explicitely depends on $\mathbf{d}$. The coefficients d_k are therefore derived by solving the following convex problem [6]:

$$\min_{\mathbf{d}} \tilde{J}(\mathbf{d}) \;\text{ s.t. }\; d_k \geq 0, \; \forall k = 1, \cdots, m \;\text{ and }\; \sum_{k=1}^{m} d_k = 1.$$

Algorithm 1 Iterative procedure to solve TSVM-MKL

```
Set an initial estimation $d^0$, $b^0$, $f_k^0$ and $t = 0$
repeat
    Calculate the terms $\nu_i, i = 1, \ldots, l + 2u$ using (3).
    Determine $d^{t+1}, b^{t+1}, f_k^{t+1}, k = 1, \ldots, m$ solution of
        $$\min_{f_k, b, \mathbf{d}} \quad \frac{1}{2}\sum_{k=1}^{m} \frac{a_k}{d_k}\|f_k\|_{\mathcal{H}}^2 + C\sum_{i=1}^{l} H_1(y_i g(x_i)) + C^* \sum_{i=l+1}^{l+2u} H_1(y_i g(x_i)) - C^* \sum_{i=l+1}^{l+2u} \nu_i y_i g(x_i) \quad (4)$$
        $$\text{s.t.} \quad \sum_{k=1}^{m} d_k = 1, \quad d_k \geq 0, \quad \forall k = 1, \ldots, m, \quad \frac{1}{u}\sum_{i=l+1}^{l+u} g(x_i) = \frac{1}{l}\sum_{i=1}^{l} y_i.$$
until a convergence criterion is satisfied.
```

This optimization can be achieved by gradient descent ($\mathbf{d} \leftarrow \mathbf{d} - \tau\nabla_{\mathbf{d}}\tilde{J}(\mathbf{d})$) projected on the simplex to ensure its feasibility. The new solution $\mathbf{d}$ is therefore plugged into (4) which is solved for f_k and b. This procedure alternates between the calculation of $\mathbf{d}$ and f_k (b). we deem it reaches convergence when $\mathbf{d}$ does not evolve anymore. Assume the sub-differential of Hingle loss is defined as:

$$\partial H_1(z)/\partial z = \begin{cases} 0 & \text{if } \; z > 1 \\ -1 & \text{if } \; z < 1 \\ -\tilde{\eta} & \text{if } \; z = 1 \quad \text{with } 0 \leq \tilde{\eta} \leq 1. \end{cases}$$

when d_k is fixed, we will attain the solution for f_k:

$$f_k(x) \;=\; \frac{d_k}{a_k}\sum_{i=0}^{l+2u}(\alpha_i y_i + C^*\gamma_i)\kappa_k(x_i, x) \tag{5}$$

with these notations: (1) $\alpha_i = C\eta_i$, $0 \leq \alpha_i \leq C$, for $i = 1, \cdots, l$. (2) $\alpha_i = C^*\eta_i$, $0 \leq \alpha_i \leq C^*$, for $i = l+1, \cdots, l+2u$. (3) $y_0 = 1$ and $\kappa_k(x_0, x) = \frac{1}{u}\sum_{i=l+1}^{l+u}\kappa_k(x_i, x)$. (4) $\gamma_i = 0$ for $i = 0, \cdots, l$; $\gamma_i = \nu_i y_i$ for $i = 0, \cdots, l+2u$. η stands for a subgradient and x_0 is a virtual sample used to encode easily the balancing constraint as in [5]. Following the same procedure, we get the optimal condition related to the bias term b as: $\sum_{i=0}^{l+2u}(\alpha_i y_i + C^*\gamma_i) = 0$. Finally, the problem turns to solving a particular SVM-type problem with the kernel $\kappa(x_1, x_2) = \sum_{k=1}^{m}\frac{d_k}{a_k}\kappa_k(x_1, x_2)$.

4 Experimental analysis

Five benchmark data sets (G50c, Text, Page, Link, and Pagelink) were selected from [8]. Before experiments, we define the pool of kernels that composed of the

Data set	G50c	Text	Link	Page	Pagelink
SVM	9.7	18.9	26.7	20.8	14.2
LapSVM	5.4(0.6)	10.4(1.1)	14.9(8.8)	10.5(0.7)	6.3(0.6)
TSVM	5.7(1.6)	**6.0(1.1)**	11.6(2.9)	10.6(8.5)	8.6(7.3)
TSVM-MKL	**4.4(0.7)**	6.2(1.6)	**10.0(6.4)**	**8.3(5.2)**	**5.6(5.8)**

Table 1: Transductive setting: misclassification rates on unlabeled data

basic kernels and the manifold kernels. Gaussian and linear kernel are popular choices for basic kernels. Manifold kernels are obtained by deforming basic kernels in the same way in [8]. Involved hyper-parameters are (γ, σ, p, N), where γ is the ratio that specify a trade-off between ambient regularization and deformation, σ is the width of similarity measure, N is the number of neighbored samples, and p specifies the degree of induced Laplacian of the similarity graph. Any tuple of parameters leads to a kernel.

Transductive setting The training set comprises of n samples, l of which are labeled. Quality of TSVM-MKL is assessed by predicting the labels of the $n - l$ unlabeled samples. In our experiments, we set $C = C^\star$, C and s are selected by grid search over [10 100 1000] and [0 : 0.2 : 0.6] respectively. Gaussian kernel and euclidean nearest neighbor graphs with gaussian weights were used on G50c and Text. Linear basic kernel and cosine nearest neighbor graphs with gaussian weights were used for the rest data sets. Table 1 shows the results of involved algorithms with standard deviation indicated in parantheses. Results of SVM and LapSVM on G50c and Text are taken from [8]. To compare with our proposed algorithm fairly, we redo all the other experiments in the same setup with the same data splits.

Inductive setting The training set consists of l labeled samples and u unlabeled samples, the test set consists of $n - l - u$ data points. Performance is evaluated by predicting the labels of unseen test set. It aims to assess the generalization ability of TSVM-MKL on the out-of-sample situation. We perform a 4-fold cross validation for each data set. Optimal C, C^* and s are attained according to the misclassification rates on unlabeled sets. The multiple kernels are defined in the same way with those in transductive setting. Table 2 reports the results of predicting the labels of unlabeled and test data. Results of SVM and LapSVM on G50C and Text are taken from [8]. We redo all the other experiments in the same experimental setting. Experiments of SVM are implemented in this way: train an SVM on labeled set, and test it on unseen test set.

From Table 1 and 2, we can see that TSVM-MKL achieves the best solution in most cases. This indicates that the combination of cluster and manifold assumption can improve the performance of semi-supervised algorithms.

Data set	G50c	Text	Link	Page	Pagelink
Algorithm	Unlab	Unlab	Unlab	Unlab	Unlab
	Test	Test	Test	Test	Test
SVM	9.7	20.9	24.8	23.8	25.1
	9.7	20.9	24.8	23.8	25.1
LapSVM	4.9	9.9	21.2(21.4)	14.1(7.1)	12.8(8.4)
	5.0	9.7	21.1(21.3)	15.5(6.1)	14.4(6.0)
TSVM	5.4(1.1)	6.5(1.1)	11.6(2.7)	11.5(8.3)	9.0(7.2)
	6.1(1.3)	6.8(1.0)	11.2(2.8)	11.6(8.6)	8.9(7.0)
TSVM-MKL	**4.5(5.0)**	**6.2(1.4)**	**9.6(6.0)**	**8.5(4.6)**	**5.6(5.7)**
	4.7(5.2)	**6.4(1.5)**	**9.4(6.1)**	**9.0(4.9)**	**6.2(5.7)**

Table 2: Inductive setting: misclassification rates on unlabeled and test data

5 Conclusion

This paper presents a multi-kernel framework for semi-supervised learning. It fuses the manifold and cluster assumption into one learning task to obtain an inductive inference model. Empirical studies on benchmark data sets demonstrate that SSL-MKL is more effective than the single assumption algorithms. Forthcoming work will be deserved to non-sparse kernel combination.

References

[1] O. Chapelle and A. Zien., Semi-supervised classification by low density separation, In Z. Ghahramani and R. Cowell, editors, *proceedings of the* 10^{th} *International Workshop on Artificial Intelligence and Statistics* (AISTATS 2005), pages 57-64, January 6-8, Barbados, 2005.

[2] M. Belkin, P. Niyogi, and V. Sindhwani, Manifold regularization: a geometric framework for learning from label and unlabeled examples, *Journal of Machine Learning Research*, 7:2399-2434, 2006.

[3] Z. Xu, R. Jin, J. Zhu, I. King, M. Lyu, and Z. Yang, Adaptive regularization for transductive support vector machine, In Y. Bengio et al., editors, *Proceedings of the* 23^{th} *Annual Conference on Neural Information Processing Systems* (NIPS 2009), pages 2125-2133, December 7-9, Vancouver (Canada), 2009.

[4] G. Dai and D. Yeung, Kernel selection for semi-supervised kernel machines, In Proceedings of the 24^{th} *International Conference on Machine Learning* (ICML 2007), pages 185-192, June 20-24, Oregon (USA), 2007.

[5] R. Collobert, F. Sinz, J. Weston, and L. Bottou, Large scale transductive svms, *Journal of Machine Learning Research*, 7:1687-1712, 2006.

[6] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, SimpleMKL, *Journal of Machine Learning Research*, 9:2491-2521, 2008.

[7] Z. Xu, R. Jin, H. Yang, I. King, and M. Lyu, Simple and efficient multiple kernel learning by group lasso, In *Proceedings of the 27th International Conference on Machine Learning* (ICML 2010), pages 1191-1198, Haifa (Israel), June 22-24, 2010.

[8] V. Sindhwani, P. Niyogi, and M. Belkin, Beyond the point cloud: from transductive to semi-supervised learning, In *Proceedings of the 22nd International Conference on Machine Learning* (ICML 2005), pages 824-831, Bonn (Germany), August 7-11, 2005.