Implicit Learning of Simpler Output Kernels for Multi-Label Prediction

Hanchen Xiong Sandor Szedmak Justus Piater
Institute of Computer Science, University of Innsbruck
Innsbruck, A-6020, Austria
{*hanchen.xiong, sandor.szedmak, justus.piater*}@*uibk.ac.at*

Abstract

It has been widely agreed that, in multi-label prediction tasks, capturing and utilizing dependencies among labels is quite critical. Therefore, a research tendency in multi-label learning is that increasingly more sophisticated dependency structures on labels (*e.g.* output kernels) are proposed. We show that, however, over-complex dependency structures will harm more than help learning when the underling dependency is relatively weak. To avoid overfitting on structures, a regularization on label-dependency is desirable. In this paper, we put forward a novel *joint-SVM* for multi-label learning. Compared to other discriminative learning schemes, joint-SVM has two strengths: at first, the complexity of training joint-SVM is almost the same as training a single regular SVM, which is quite efficient; secondly, in joint-SVM, a linear output kernel on multi-label is implicitly learned and a regularization on the output kernel is implicitly added, which enhances generalization ability. In our experimental results on image annotation, joint-SVM compares favorably state-of-the-arts methods.

1 Predict Multi-label as Structured Outputs

In the past two decades, support vector machines (SVMs) have displayed remarkable successes in various application domains. The achievements of SVMs mainly stems from its two advantageous components: *maximum margins* and *input kernels*. The maximum-margin principle is a reflection of statistical learning theory [12] on linear binary classification. Kernels provide powerful mechanisms enabling the linear classifier to separate highly non-linear data. The critical observation of kernel methods is that a kernel function can be defined on a pair of data instances to implicitly map them to a reproducing kernel Hilbert space (RKHS):

$$K_\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \rangle \tag{1}$$

where $\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in \mathbb{R}^d$ are two input training instances, ϕ is the feature map induced by kernel function K_ϕ, and $\phi(\mathbf{x}^{(i)})$ is the representation of $\mathbf{x}^{(i)}$ in the RKHS $\mathcal{H}_\phi$. Given the training dataset $\{\mathbf{x}^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{+1, -1\}\}_{i=1}^m$, the primal form of training SVM is:

$$\begin{array}{rl} \arg\min\limits_{\mathbf{w}\in\mathbb{R}^{\mathcal{H}_\phi}} & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^m \xi^{(i)} \\ \text{s.t.} & y^{(i)}\left(\mathbf{w}^\top \phi(\mathbf{x}^{(i)})\right) \geq 1 - \xi^{(i)}, \xi^{(i)} \geq 0, i \in \{1, \ldots, m\} \end{array} \tag{2}$$

where $\mathbf{w}$ is a linear hyperplane in $\mathcal{H}_\phi$, $\xi^{(i)}$ are slack variables for the tolerance of noise, and C is a trade-off parameter. (2) differs from usual SVM formulation slightly at the absence of a bias term. Here we ignore the bias since it can be absorbed in $\mathbf{w}$. The computational advantage of kernels become obvious when the primal form of SVM (2) is reformulated to its dual form:

$$\begin{array}{rl} \arg\min\limits_{\alpha_1,\alpha_2,\ldots,\alpha_m} & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j y^{(i)} y^{(j)} K_\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{s.t.} & \forall i, 0 \leq \alpha_i \leq C \end{array} \tag{3}$$

The dual representation of $\mathbf{w}$ is $\sum_{i=1}^{m} \alpha_i y^{(i)} \phi(\mathbf{x}^{(i)})$, and thus the prediction of a test instance $\hat{\mathbf{x}}$ is

$$\hat{y} = \text{sgn}\left(\mathbf{w}^\top \phi(\hat{\mathbf{x}})\right) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y^{(i)} K_\phi(\mathbf{x}^{(i)}, \hat{\mathbf{x}})\right). \tag{4}$$

We can denote $y^{(i)}\left(\mathbf{w}^\top \phi(\mathbf{x}^{(i)})\right)$ in the constraints of (2) as a score function $F(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w})$, then for binary outputs $y^{(i)}$, $F\left(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w}\right) - F\left(\mathbf{x}^{(i)}, -y^{(i)}; \mathbf{w}\right) = 2 \times F\left(\mathbf{x}^{(i)}), y^{(i)}; \mathbf{w}\right)$. Also, a distance function between binary outputs can be denoted as $d(y^{(i)}, -y^{(i)}) = |y^{(i)} - (-y^{(i)})| = 2$. Then by replacing C with $\frac{C}{2}$, (2) can be rewritten as:

$$\begin{aligned} \arg \min_{\mathbf{w} \in \mathbb{R}^{\mathcal{H}_\phi}} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + C \textstyle\sum_{i=1}^{m} \xi^{(i)} \\ \text{s.t.} \quad & \forall i, \underbrace{F\left(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w}\right) - F\left(\mathbf{x}^{(i)}, -y^{(i)}; \mathbf{w}\right)}_{\Delta_F(y^{(i)}, -y^{(i)})} \geq d(y^{(i)}, -y^{(i)}) - \xi^{(i)}, \xi^{(i)} \geq 0 \end{aligned} \tag{5}$$

which is a binary-output case of structural SVM [11] (see later). By using hinge-loss representation for $\xi^{(i)}$, (5) is:

$$\arg \min_{\mathbf{w} \in \mathbb{R}^{\mathcal{H}_\phi}} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{m} \max\{0, d(y^{(i)}, -y^{(i)}) - \Delta_F(y^{(i)}, -y^{(i)})\} \tag{6}$$

Structural SVM [11] is an extension of SVM for structured-outputs, in which, however, the margin to be maximized is defined as the score gap between the desired output and the runner-up. Assume that structured outputs $\mathbf{y} \in \mathcal{Y}$, and the score function is linear in some *combined feature representation* of inputs and outputs $\mathbf{\Psi}(\mathbf{x}, \mathbf{y})$: $F(\mathbf{x}, \mathbf{y}; \mathbf{W}) = \langle \mathbf{W}, \mathbf{\Psi}(\mathbf{x}, \mathbf{y}) \rangle$, then the objective function of structural SVM is:

$$\arg \min_{\mathbf{W} \in \mathbb{R}^{\mathbf{\Psi}}} \frac{1}{2}||\mathbf{W}||^2 + C \sum_{i=1}^{m} \max_{\mathbf{y}' \in \mathcal{Y}} \left\{ d(\mathbf{y}^{(i)}, \mathbf{y}') - \Delta_F(\mathbf{y}^{(i)}, \mathbf{y}') \right\} \tag{7}$$

where $\Delta_F(\mathbf{y}^{(i)}, \mathbf{y}') = F(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}; \mathbf{W}) - F(\mathbf{x}^{(i)}, \mathbf{y}'; \mathbf{W})$ and $d(\mathbf{y}^{(i)}, \mathbf{y}')$ is a distance function defined on structured outputs. In multi-label scenario, given a set of T labels, then outputs are T-dimensional binary vector $\mathbf{y} = [y_1, \cdots, y_t, \cdots, y_T]^\top \in \mathbb{B}^T$. When we define the score function $F\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}; \mathbf{W}\right) = \langle \mathbf{W}, \phi(\mathbf{x}^{(i)}) \otimes \mathbf{y}^{(i)} \rangle$, and use *Hamming distance* on outputs, then because of linear decomposability, (7) can be rewritten as:

$$\begin{aligned} \arg \min_{\mathbf{W} \in \mathbb{R}^{\mathcal{H}_\phi} \times \mathbb{R}^T} \quad & \tfrac{1}{2}||\mathbf{W}||_F^2 + C \textstyle\sum_{i=1}^{m} \sum_{t=1}^{T} \max_{y'_t = \{-1, +1\}} \left\{ d(y_t^{(i)}, y'_t) - \Delta_F(y_t^{(i)}, y'_t) \right\} \\ & \Downarrow \\ \arg \min_{\mathbf{w}_1, \cdots, \mathbf{w}_T \in \mathbb{R}^{\mathcal{H}_\phi}} \quad & \textstyle\sum_{t=1}^{T} \left\{ \tfrac{1}{2}||\mathbf{w}_t||^2 + C \sum_{i=1}^{m} \max\left\{ 0, d(y_t^{(i)}, -y_t^{(i)}) - \Delta_F(y_t^{(i)}, -y_t^{(i)}) \right\} \right\} \end{aligned} \tag{8}$$

where $\langle \cdot, \cdot \rangle_F$ denotes Frobenius product and $||\mathbf{W}||_F$ is the Frobenius norm of matrix $\mathbf{W}$.

2 Joint SVM

It can be seen (by linking (6) and (8)) that, with linearly decomposable score functions and output distances, using structural SVM on multi-label learning is equivalent to learning T SVMs jointly. This is closely related to multi-task learning frameworks [1], where different learning tasks are connected by summing up their objectives and constraints respectively:

$$\begin{aligned} \min \quad & \tfrac{1}{2} \textstyle\sum_{t=1}^{T} ||\mathbf{w}_t||^2 + C \sum_{t=1}^{T} \sum_{i=1}^{m} \xi_t^{(i)} \\ \text{w.r.t.} \quad & \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T \in \mathbf{R}^{\mathcal{H}_\phi \times 1} \\ \text{s.t.} \quad & \textstyle\sum_{t=1}^{T} y_t^{(i)} \left(\mathbf{w}_t^\top \phi(x^{(i)})\right) \geq T - \sum_{t=1}^{T} \xi_t^{(i)} \end{aligned} \tag{9}$$

By denoting $\mathbf{y}^{(i)} = [y_1^{(1)}, \ldots, y_T^{(i)}]$, and $\mathbf{W} = [\frac{\mathbf{w}_1^\top}{T}; \ldots; \frac{\mathbf{w}_T^\top}{T}]^\top$, we can rewrite (9) as:

$$\begin{aligned} \arg \min_{\mathbf{W} \in \mathbb{R}^T \times \mathcal{H}_\phi} \quad & \tfrac{1}{2}||\mathbf{W}||_F^2 + C \textstyle\sum_{i=1}^{m} \bar{\xi}^{(i)} \\ \text{s.t.} \quad & \left\langle \mathbf{y}^{(i)}, \mathbf{W}\phi(x^{(i)}) \right\rangle \geq 1 - \bar{\xi}^{(i)}, \xi_i \geq 0, i \in \{1, \ldots, m\} \end{aligned} \tag{10}$$

which is referred to as *joint SVM*. When linear output kernels ($K_\psi(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) = \langle \psi(\mathbf{y}^{(i)}), \psi(\mathbf{y}^{(j)}) \rangle$) [4, 7, 13] are applied on outputs, (10) will be:

$$\begin{array}{rl} \arg \min\limits_{\mathbf{W} \in \mathbb{R}^{H_\psi \times \mathcal{H}_\phi}} & \frac{1}{2} ||\mathbf{W}||_F^2 + C \sum_{i=1}^m \bar{\xi}^{(i)} \\ \text{s.t.} & \left\langle \psi(\mathbf{y}^{(i)}), \mathbf{W}\phi(x^{(i)}) \right\rangle \geq 1 - \bar{\xi}^{(i)}, \xi_i \geq 0, i \in \{1, \dots, m\} \end{array} \tag{11}$$

Since the linear decomposability of $\Delta_F(\mathbf{y}^{(i)}, \mathbf{y}')$ is still preserved, join SVM solves the same problem as structural SVM. However, one strength of joint SVM is that its training complexity is almost the same as a single SVM, by contrast to the exponential complexity in structural SVM. Similarly to regular SVM, joint SVM can be converted to its dual form

$$\begin{array}{rl} \arg \min\limits_{\alpha_1, \cdots, \alpha_m} & \sum_{i=1}^m \alpha_i - \sum_{i,j=1}^m \alpha_i \alpha_j K_\psi(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) K_\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{s.t} & \forall i, 0 \leq \alpha_i \leq C \end{array} \tag{12}$$

with $\mathbf{W} = \sum_i^m \alpha_i \psi(\mathbf{y}^{(i)}) \phi(\mathbf{x}^{(i)})^\top$. It can be seen that, with the kernel matrix on outputs precomputed, the computational complexity of joint SVM (12) is the same as the learning of one single SVM (3), which is a great advantage in efficiency. Meanwhile, when more general output kernels are used, then the linear decomposability of $\Delta_F(\mathbf{y}^{(i)}, \mathbf{y}')$ will be violated, then joint SVM becomes a special case of max-margin regression [10], which seeks to learn linear operators $\mathbf{W} : \mathcal{H}_\phi \to \mathcal{H}_\psi$ from general $\phi(\mathbf{x}) \otimes \psi(\mathbf{y})$.

Given a test input $\hat{\mathbf{x}}$, the prediction $\psi(\hat{\mathbf{y}})$ in $\mathcal{H}_\psi$ is

$$\psi(\hat{\mathbf{y}}) = \mathbf{W}\phi(\hat{\mathbf{x}}) = \sum_{i=1}^m \alpha_i \psi(\mathbf{y}^{(i)}) K_\phi(\mathbf{x}^{(i)}, \hat{\mathbf{x}}). \tag{13}$$

Meanwhile, there is no direct way (say, by inverting Eq.(13)) to map $\psi(\hat{\mathbf{y}})$ back to $\hat{\mathbf{y}}$. Therefore, we can find the optimal solution $\hat{\mathbf{y}}^*$, out of all possible $\mathbf{y} \in \{+1, -1\}^T$, such that its projection in $\mathcal{H}_\psi$ is closest to $\mathbf{W}\phi(\hat{\mathbf{x}})$:

$$\begin{array}{rcl} \hat{\mathbf{y}}^* & = & \text{argmax}_{\mathbf{y} \in \{+1,-1\}^T} \langle \psi(\mathbf{y}), \mathbf{W}\phi(\hat{\mathbf{x}}) \rangle \\ & = & \text{argmax}_{\mathbf{y} \in \{+1,-1\}^T} \sum_{i=1}^m \alpha_i \underbrace{K_\phi(\mathbf{x}^{(i)}, \hat{\mathbf{x}})}_{\beta_i} K_\psi(\mathbf{y}^{(i)}, \mathbf{y}) \end{array} \tag{14}$$

In general, there is no closed-form solution to Eq.(14), so here we use a similar neighbour-based label transferring theme as [9, 6]:

$$\hat{\mathbf{y}}^* = \left(\sum_{k=1}^K \mathbf{y}^{(k)} w_k \right) \Bigg/ \sum_{k=1}^K w_k \qquad w_j = \sum_{i=1}^m \alpha_i \beta_i K_\psi(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) \tag{15}$$

where $k = \{j \in [1, m] : w_j > 0\}$ and maximum $K = 10$ neighbours are taken into account. Since α_i are $K_\psi(\mathbf{y}^{(i)}, \mathbf{y}^{(j)})$ were already computed in the training phase, only the computation of $\{\beta_i\}_{i=1}^m$ is needed during testing. Thus, the complexity in predicting is $\mathcal{O}(m)$.

3 Implicit Learning and Regularization of Output Kernels

Assume that the statistics of tags' pairwise co-occurrence can be encoded in a $T \times T$ matrix $\mathbf{P}$[3, 4, 7, 13], via which the output vectors can be linearly mapped as $\psi(\mathbf{y}) = \mathbf{P}\mathbf{y}$, and thus the corresponding linear output kernel is:

$$K_\psi^{Lin}(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) = \mathbf{y}^{(i)\top} \mathbf{\Omega} \mathbf{y}^{(j)} \tag{16}$$

where $\mathbf{\Omega} = \mathbf{P}^\top \mathbf{P} = \mathbf{P}\mathbf{P}^\top$. By denoting $\mathbf{U} = \mathbf{P}^\top \mathbf{W}$, we can rewrite joint SVM (11) as:

$$\begin{array}{rl} \arg \min\limits_{\mathbf{W} \in \mathbb{R}^{H_\psi \times \mathcal{H}_\phi}} & \frac{1}{2} ||\mathbf{W}||_F^2 + C \sum_{i=1}^m \bar{\xi}^{(i)} \\ \text{s.t.} & \left\langle \mathbf{y}^{(i)}, \mathbf{U}\phi(x^{(i)}) \right\rangle \geq 1 - \bar{\xi}^{(i)}, \xi_i \geq 0, i \in \{1, \dots, m\} \end{array} \tag{17}$$

Meanwhile, we need to control the scale of $\mathbf{P}$, otherwise the constraints in (17) will be pointless. Different regularizations on $\mathbf{P}$ have been proposed in previous work. In [4] one extra regularization on $\mathbf{\Omega}$, $\frac{1}{2}||\mathbf{\Omega}||_F^2$, was added into the objective function, while $||P||_F = 1$ was used in [13]. By

	Corel5K			Espgame			Iaprtc12		
Method	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
MBRM [5]	24.0	25.0	24.0	18.0	19.0	18.0	24.0	23.0	23.0
JEC [9]	27.0	32.0	29.0	24.0	19.0	21.0	29.0	19.0	23.0
TagProp [6]	33.0	42.0	37.0	39.0	27.0	**32.0**	45.0	**34.0**	**39.0**
FastTag [3]	32.0	**43.0**	37.0	**46.0**	22.0	30.0	**47.0**	26.0	34.0
JSVM	**48.5**	38.0	**42.6**	32.7	**31.6**	**32.2**	42.2	29.4	34.6
JSVM+Pol(2)	46.6	37.0	41.3	32.6	24.4	27.9	37.9	26.6	31.2
JSVM+Pol(3)	41.5	31.3	35.7	28.5	21.3	24.4	38.0	26.1	31.0

Table 1: Comparison between different versions of joint SVM and other related methods on three benchmark databases. P, R and F1 denote precision, recall and F1 measure respectively.

contrast, a pseudo regularization on $\mathbf{P}$ is used in [3] via the re-construction loss from manually-corrupted data and $\mathbf{P}$. Similar to [4], we want to add a regularizer to control overfitting from output dependency-structures. Meanwhile, by merging regularization on $\mathbf{W}$ and $\mathbf{P}$, we obtain a more compact regularizer, $\frac{1}{2}\mathbf{W}^\top \boldsymbol{\Omega} \mathbf{W}$, resulting in:

$$\begin{aligned} \arg \min_{\mathbf{U}\in\mathbb{R}^{H_\psi \times \mathcal{H}_\phi}} \quad & \tfrac{1}{2}||\mathbf{U}||_F^2 + C \textstyle\sum_{i=1}^m \bar{\xi}^{(i)} \\ \text{s.t.} \quad & \left\langle \mathbf{y}^{(i)}, \mathbf{U}\phi(x^{(i)}) \right\rangle \geq 1 - \bar{\xi}^{(i)}, \xi_i \geq 0, i \in \{1, \ldots, m\} \end{aligned} \tag{18}$$

Remarkably, (18) is equivalent to (11) with $\mathbf{W}$ substituted by $\mathbf{U}$, which suggests that a linear output kernel is implicitly learned, and absorbed in $\mathbf{W}$, when we training a plain joint SVM with no explicit kernel on outputs. In addition, a regularization on the output kernel is also implicitly added.

4 Experiments

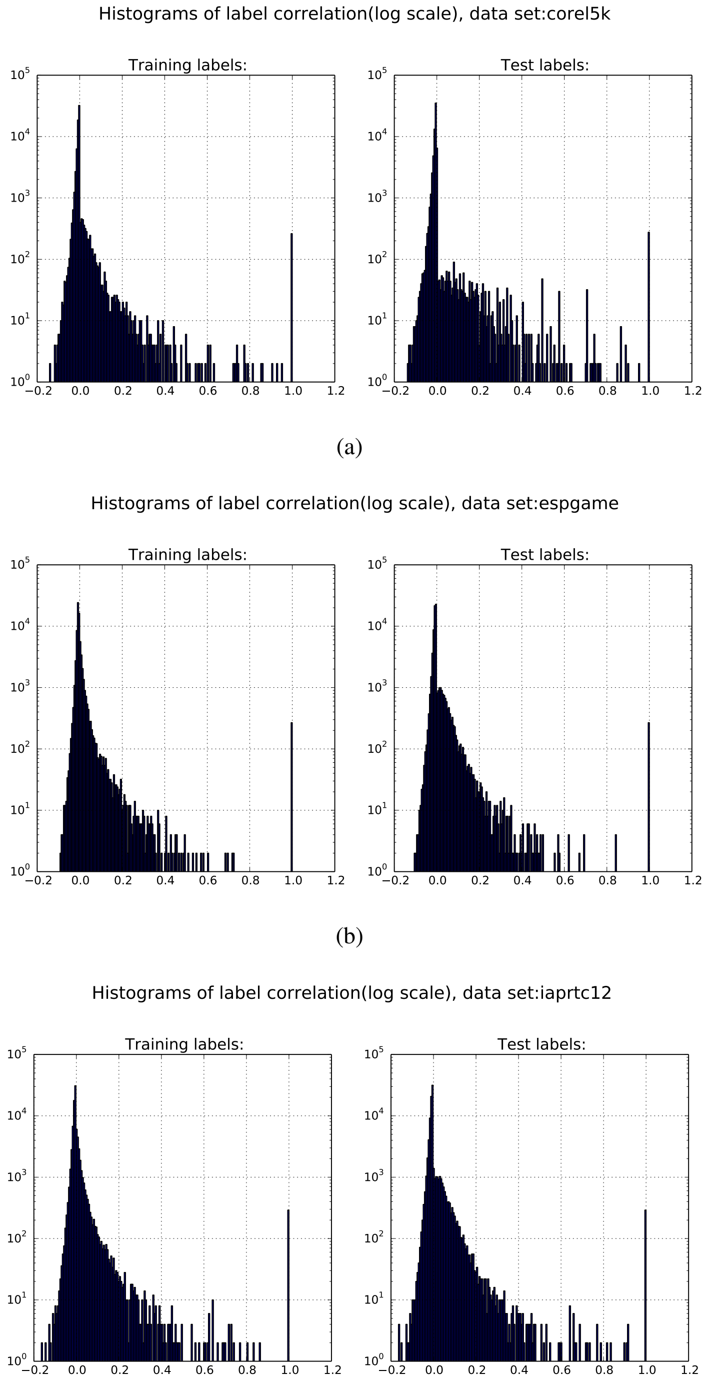
In our experiments, we evaluated the propose joint SVM on image annotation tasks. Here, we used three benchmark datasets, Corel5k, Espgame and Iaprtc12. These three datasets have been widely used in image annotation studies [8, 2, 5, 6, 9, 3] with performance evaluations reported therein. Therefore, we can easily compare our method with others. We used the same visual features as in [6, 3]. Three types of joint SVMs with different output kernels are tested: plain joint SVM (JSVM), 2-degree polynomial (JSVM+Pol(2)) and 3-degree polynomial (JSVM+Pol(3)).

The experimental results, together with the reported results from other related work, are presented in Table 1. We can see that plain joint SVM (JSVM) outperforms all other results on Corel5k and Espgame datasets. JSVM is also the second best result on Iaprtc12 dataset. JSVM+Pol(2) also worked better than some old methods [5, 9]. Meanwhile, JSVM+Pol(3) is worse than JSVM+Pol(2).

Discussions Based on our experiments, it seems that plain joint SVM (JSVM) works more robustly than the joint SVMs with explicit output kernels. In order to dig deeper to find an explanation, we can study the correlation matrices of output tag-sets in three datasets. In Figure 1, for each dataset, we plot the histograms (in log scale) of all correlation values in both training sets and testing sets. We found that most entries in correlation matrices are 0, which means that the pairwise correlation (or roughly speaking, dependencies) is rather sparse. Although JSVM, JSVM+Pol(2) both encode pairwise dependencies, it should be reminded that the implicit linear output kernel in JSVM is in regularization term, which implies that simpler output kernels (dependencies) are encouraged. However, JSVM+Pol(2) does not have this preference. Therefore, JSVM can implicitly learned most simple output kernels when no more complex ones are needed. Analogously, the same principle can explain why even JSVM+Pol(3) led to worse results.

5 Conclusions

A novel joint SVM was presented for multi-label learning. One benefit of using joint SVM is that the learning and regularization of a linear output kernel are implicitly conducted. Moreover, both training joint SVM and predicting with joint SVM are efficient. As a possible work direction, we might investigate more interesting output kernel regularization schemes to fit different applications.



(a)

(b)

(c)

Figure 1: The histograms (in log scale) of all correlation values in both training sets and testing sets: (a) Corel5k, (b) Espgame (c) Iartc12.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience.

References

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[2] David M. Blei and Michael I. Jordan. Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 2003.

[3] Minmin Chen, Alice Zheng, and Kilian Q. Weinberger. Fast image tagging. In *ICML*, 2013.

[4] Francesco Dinuzzo, Cheng Soon Ong, Peter V. Gehler, and Gianluigi Pillonetto. Learning output kernels with block coordinate descent. In *ICML*, 2011.

[5] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[6] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[7] Bharath Hariharan, S. V. N. Vishwanathan, and Manik Varma. Efficient max-margin multi-label classification with applications to zero-shot learning. *Machine Learning*, 88(1-2):127–155, 2012.

[8] Victor Lavrenko, R. Manmatha, and Jiwoon Jeon. A model for learning the semantics of pictures. In *NIPs*. 2004.

[9] Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar. Baselines for image annotation. *International Journal of Computer Vision*, 90:88–105, 2010.

[10] Sandor Szedmak and John Shawe-taylor. Learning via linear operators: Maximum margin regression. Technical report, University of Southampton, UK, 2005.

[11] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[12] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

[13] Yu Zhang and Dit-Yan Yeung. Multilabel relationship learning. *ACM Trans. Knowl. Discov. Data*, 7(2):1–30, August 2013.