This copy is for personal use, the final version will be publisehd in the proceedings of the 22th European Symposium on Artificial Neural Networks, Computation Intelligence and Machine Learning (ESANN 2014) ©*ESANN2014*

Joint SVM for Accurate and Fast Image Tagging

Hanchen Xiong   Sandor Szedmak   Justus Piater [*]

Institute of Computer Science, University of Innsbruck
Technikerstr.21a A-6020, Innsbruck, Austria

Abstract. This paper studies how joint training of multiple support vector machines (SVMs) can improve the effectiveness and efficiency of automatic image annotation. We cast image annotation as an output-related multi-task learning framework, with the prediction of each tag's presence as one individual task. Evidently, these tasks are related via correlations between tags. The proposed joint learning framework, which we call *joint SVM*, can encode the correlation between tags by defining appropriate kernel functions on the outputs. Another practical merit of the joint SVM is that it shares the same computational complexity as one single conventional SVM, although multiple tasks are solved simultaneously. According to our empirical results on an image-annotation benchmark database, our joint training strategy of SVMs can yield substantial improvements, in terms of both accuracy and efficiency, over training them independently. In particular, it outperforms many other state-of-the-art algorithms.

1 Introduction

Automatic image annotation is an important yet challenging machine learning task. The importance is based on the fact that the number of images grows exponentially on the internet, and most of them have no link to semantic tags. Therefore, automatic annotation is of great significance to generate meaningful metadata for image retrieval from textual queries. The challenges are usually considered from two perspectives: first, to directly apply mature binary classification methods, e.g. Support Vector Machines (SVMs), assumes the independence of the labels; secondly, the image data on internet is usually presented in large volumes, so the desired learning method should be capable of working on large-scale data with high learning and prediction efficiency. One straightforward yet naive strategy is to consider each tag's presence as a binary classification problem. Then, multiple SVMs can be trained independently for different tags. This method, however, will suffer from high computational complexity in both training and prediction phases when the number of tags is relatively large. Independently learning multiple SVMs is not expected to work well because it ignores the correlation between the presences of tags, which is a phenomenal characteristic of image annotation tasks (e.g., sky and cloud often co-occur). In this paper, we propose to interpret image annotation as the learning of multiple related tasks. However, different from most existing multi-task learning

[*] 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.

frameworks [1] in which tasks are related through their *inputs*, our joint learning method focuses on the relation between *outputs*. Our strategy is motivated by two intuitions. First, by connecting multiple SVM classifiers together, the correlation between their outputs (the presences of tags), presumably, can be more easily encoded. Secondly, if the outputs of multiple SVMs can be merged into a single vector entity, the optimization problem can be established and solved over vectors, greatly reducing the computational complexity. These two objectives, surprisingly, can be easily achieved by summing up the objectives and constraints in different SVMs, plus an appropriately designed kernel on outputs.

2 Joint Learning of Multiple SVMs

2.1 Support Vector Machines and Input Kernels

In the past two decades, support vector machines (SVMs) have seen remarkable successes in various 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 [2] 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 ith and jth 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 $\mathcal{H}_\phi$. Most popularly, a Gaussian (or radial basis function) kernel

$$K_\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(\frac{-\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{2\sigma^2}\right) \tag{2}$$

is employed because its corresponding RKHS is an unnormalized Gaussian density function

$$\phi(\mathbf{x}^{(i)}) \propto \mathcal{N}(\tau; \mathbf{x}^{(i)}, \sigma) \tag{3}$$

which is of infinite dimension, and thus greatly improves the representational capability of input data. Given the training dataset $\{\mathbf{x}^{(i)} \in \mathbf{R}^d, y^{(i)} \in \{+1, -1\}\}_{i=1}^m$ of one binary classification problem, the primal form of training SVM is written

$$\begin{array}{rl} \min & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^m \xi^{(i)} \\ \text{w.r.t.} & \mathbf{w} \in \mathbb{R}^{\mathcal{H}_\phi \times 1}, b \in \mathbb{R} \\ \text{s.t.} & y^{(i)}\left(\mathbf{w}^\top \phi(\mathbf{x}^{(i)}) + b\right) \geq 1 - \xi^{(i)}, \xi^{(i)} \geq 0, i \in \{1, \ldots, m\} \end{array} \tag{4}$$

where $\mathbf{w}$ is a linear hyperplane in $\mathcal{H}_\phi$, b is bias term, $\xi^{(i)}$ are slack variables for the tolerance of noise, and C is trade-off parameter between training error and max-margin regularization. The computational advantage of kernels become more obvious when the primal form of SVM (4) is reformulated to its dual form by introducing Lagrange multipliers α_i for each constraints:

$$\begin{array}{rl} \max & \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} K_\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{w.r.t.} & \alpha_1, \alpha_2, \ldots, \alpha_m \\ \text{s.t.} & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0; \forall i, 0 \le \alpha_i \le C \end{array} \tag{5}$$

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

$$\hat{y} = \operatorname{sgn}\left(\mathbf{w}^\top \phi(\hat{\mathbf{x}}) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y^{(i)} K_\phi(\mathbf{x}^{(i)}, \hat{\mathbf{x}}) + b\right). \tag{6}$$

It can be seen that the kernel function K_ϕ enables the learning of a infinite-dimensional $\mathbf{w}$ without explicit computation in $\mathcal{H}_\phi$. (6) shows that the kernel function yields a similarity measurement between two input instances.

2.2 Joint SVM

Automatic image annotation tasks seek to predict the presence of tags given an input image. If we consider prediction of each tag's occurrence as a binary classification problem, we can enlist as many SVMs as the number of tags. Similar to other multi-task learning frameworks [1, 3], we connect the learning tasks of different SVMs by simply summing up their objectives and constraints respectively in the primal form

$$\begin{array}{rl} \max & \frac{1}{2}\sum_{t=1}^{T} ||\mathbf{w}_t||^2 + C\sum_{t=1}^{T}\sum_{i=1}^{m} \xi_t^{(i)} \\ \text{w.r.t.} & \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T \in \mathbf{R}^{\mathcal{H}_\phi \times 1}, b_1, b_2, \ldots, b_T \in \mathbb{R} \\ \text{s.t.} & \sum_{t=1}^{K} y_t^{(i)} \left(\mathbf{w}_t^\top \phi(x^{(i)}) + b_t\right) \ge T - \sum_{k=1}^{T} \xi_t^{(i)} \end{array} \tag{7}$$

where t indexes different tags or learning tasks, and T is the total number of tags. By denoting $\mathbf{y}^{(i)} = [y_1^{(1)}, \ldots, y_T^{(i)}]$, $\mathbf{b} = [\frac{b_1}{T}, \ldots, \frac{b_T}{T}]$ and $\mathbf{W} = [\frac{\mathbf{w}_1^\top}{T}; \ldots; \frac{\mathbf{w}_T^\top}{T}]^\top$, we can rewrite (7) as a joint SVM:

$$\begin{array}{rl} \min & \frac{1}{2}||\mathbf{W}||_F^2 + C\sum_{i=1}^{m} \bar{\xi}^{(i)} \\ \text{w.r.t.} & \mathbf{W} \in \mathbb{R}^{K \times \mathcal{H}_\phi}, \mathbf{b} \in \mathbb{R}^K \\ \text{s.t.} & \left\langle \mathbf{y}^{(i)}, \mathbf{W}\phi(x^{(i)}) + \mathbf{b} \right\rangle \ge 1 - \bar{\xi}^{(i)}, \xi_i \ge 0, i \in \{1, \ldots, m\} \end{array} \tag{8}$$

where $||\mathbf{W}||_F$ is the Frobenius norm of matrix $\mathbf{W}$, and $\bar{\xi}^{(i)} = \frac{1}{T}\sum_{t=1}^{T} \xi_t^{(i)}$. One rationale of (7) is that within the joint form of objectives and constraints, learning easy tasks can help the learning of challenging tasks. For example, if training data $(\mathbf{x}^{(i,)}, y_p^{(i)})$ can be easily classified correctly in the pth task (i.e., $y^{(i)}(\mathbf{w}_p^\top \mathbf{x}^{(i)} + b)/T > \frac{1}{T}$), it can offer some 'freedom' to other challenging tasks before violating constraint $\left\langle \mathbf{y}^{(i)}, \mathbf{W}\phi(x^{(i)}) + \mathbf{b} \right\rangle_{\mathcal{H}} \ge 1$. In addition, a key functionality this joint form (8) can afford is that we can define kernel functions on outputs $\mathbf{y}$ to improve their representational power (e.g. correlations). Assume the kernel function defined on outputs are $K_\psi(\mathbf{y}^{(i)}, \mathbf{y}^{(j)})$ (the design of the output kernel will be explained later) and the corresponding feature map is $\psi : \mathbb{R}^K \rightarrow \mathcal{H}_\psi$, then (8) is modified to

$$\begin{array}{rl} \min & \frac{1}{2}||\mathbf{W}||_F^2 + C\sum_{i=1}^m \bar{\xi}^{(i)} \\ \text{w.r.t.} & \mathbf{W} \in \mathbb{R}^{H_\psi \times \mathcal{H}_\phi}, \mathbf{B} \in \mathbb{R}^{\mathcal{H}_\psi \times 1} \\ \text{s.t.} & \left\langle \psi(\mathbf{y}^{(i)}), \mathbf{W}\phi(x^{(i)}) + \mathbf{B} \right\rangle \geq 1 - \bar{\xi}^{(i)}, \xi_i \geq 0, i \in \{1, \ldots, m\} \end{array} \tag{9}$$

Interestingly, although derived from a rather different starting point, our joint SVM (9) is the same as Maximum Margin Regression (MMR) [4], wherein the motivation is to seek a linear operator in arbitrary tensor product space $\psi(\mathbf{y}^{(i)}) \otimes \phi(\mathbf{x}^{(i)})$. In addition, (9) is also related to structured-output learning [5] by sharing the same objective, yet with different constraints. Basically, the differences originate from two types of margins used in (9) and [5]. An empirical comparison of these two methods on structured-output learning is in [6]. The solution of the MMR stands close to the Minimum Description Length Principle, see for example in [7], by providing a highly compressed description to complex learning problems.

2.3 Output Kernels and Solutions

Similarly to a single conventional SVM, the joint SVM learning (9) can be converted to its dual form

$$\begin{array}{rl} \max & \sum_{i=1}^m \alpha_i - \sum_{i,j=1}^m \alpha_i\alpha_j K_\psi(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) K_\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{w.r.t} & \alpha_1, \cdots, \alpha_m \\ \text{s.t} & \sum_{i=1}^m \alpha_i \psi(\mathbf{y}^{(i)}) = 0; \quad \forall i, 0 \leq \alpha_i \leq C \end{array} \tag{10}$$

with $\mathbf{W} = \sum_i^m \alpha_i \psi(\mathbf{y}^{(i)})\phi(\mathbf{x}^{(i)})^\top$. It can be seen, with kernel matrix on outputs pre-computed, that the computational complexity of joint learning (10) is the same as the learning of one single SVM (5), which is a great advantage in efficiency. In this paper, the Gaussian kernel function (2) is used on $\mathbf{y}$, hence $\psi(\mathbf{y}^{(i)})$ corresponds to an unnormalized density function (which is non-negative everywhere), and the bias-induced constraint $\sum_{i=1}^m \alpha_i \psi(\mathbf{y}^{(i)}) = 0$ will lead to a trivial solution $\forall i, \alpha_i = 0$. Since the Gaussian kernel is translation invariant, the bias in output space $\mathbf{y}$ has no effect, and we can ignore the bias $\mathbf{B}$ in (9) and its corresponding constraint in (10). Therefore, given a test data $\hat{\mathbf{x}}$, the prediction $\phi(\hat{\mathbf{y}})$ in $\mathcal{H}_\psi$ is

$$\psi(\hat{\mathbf{y}}) = \mathbf{W}\phi(\hat{\mathbf{x}}) = \sum_{i=1}^m \alpha_i \psi(\mathbf{y}^{(i)}) K_\phi(\mathbf{x}^{(i)}, \hat{\mathbf{x}}). \tag{11}$$

With identical scalar σ, the Gaussian kernel (2) on $\mathbf{y}$ can be decomposed into independent Gaussian kernels on each element y_t. To preserve the correlation between every pair of tags, one simple remedy is to use a full covariance matrix Σ. Here, we use a scaled empirical covariance from outputs in training data. Another computational issue is that there is no direct way (say, by inverting (11)) to map $\psi(\hat{\mathbf{y}})$ back to $\hat{\mathbf{y}}$. Therefore, instead of finding a closed form solution, 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 $\mathbf{W}\phi(\hat{\mathbf{x}})$:

$$\begin{aligned}
\hat{\mathbf{y}}^* &= \operatorname{argmax}_{\mathbf{y}\in\{+1,-1\}^T} \langle \psi(\mathbf{y}), \mathbf{W}\phi(\hat{\mathbf{x}}) \rangle \\
&= \operatorname{argmax}_{\mathbf{y}\in\{+1,-1\}^T} \textstyle\sum_{i=1}^m \alpha_i K_\psi(\mathbf{y}^{(i)}, \mathbf{y}) \underbrace{K_\phi(\mathbf{x}^{(i)}, \hat{\mathbf{x}})}_{\beta_i} \\
&= \operatorname{argmax}_{\mathbf{y}\in\{+1,-1\}^T} \textstyle\sum_{i=1}^m \alpha_i \beta_i \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{y}^{(i)})^\top \Sigma^{-1} (\mathbf{y} - \mathbf{y}^{(i)})\right)
\end{aligned} \tag{12}$$

In general, there is no closed-form solution to (12), so usually approximate dynamic programming (ADP) is applied in searching for the optimum $\hat{\mathbf{y}}^*$. Here, we employ a simpler strategy. Since the number of tags associated with one image is relatively small, most of the $\mathbf{y}$ in $\{+1, -1\}^T$ space are bad solutions. Therefore, when the training data size is large, the most likely solutions of (12), presumably, are covered by the outputs in training data $\{\mathbf{y}\}_{i=1}^m$. Consequently, we can find the optimum $\hat{\mathbf{y}}^*$ via

$$\operatorname*{argmax}_{\{\mathbf{y}^{(j)}\}_{j=1}^m} \sum_{i=1}^m \alpha_i \beta_i \underbrace{\exp\left(-\frac{1}{2}(\mathbf{y}^{(i)} - \mathbf{y}^{(j)})^\top \Sigma^{-1} (\mathbf{y}^{(j)} - \mathbf{y}^{(i)})\right)}_{\gamma_{ij}} \tag{13}$$

where $\{\gamma_{ij}\}_{i,j=1}^m$ were already computed in the training phase, $\{\alpha_i\}_{i=1}^m$ are training results, and only the computation of $\{\beta_i\}_{i=1}^m$ is needed during testing.

3 Experiment

In our experiment, we used the Corel5K benchmark dataset with image features extracted as in [8]. The dataset contains 5,000 images of different scenarios and objects, out of which 4500 images are used as training data and 500 images are test data. In the database, there exist 260 tags, and on average each image is annotated with 3.5 tags. For all images in the database, 15 different features [8] were extracted.

In our experiment, we applied both a joint SVM and independent SVMs for comparison. To ensure fairness, in the learning phase, the optimization problems (5) and (10) were solved with the same coordinate descent method [9]. In addition, for both independent SVMs and the joint SVM, 500 instances of training data were taken out as validation data to find the best parameter C. In the testing phase, the performance was measured with precision, recall and F1 score. To measure the efficiency, training and testing times were recorded as well. We tried two learners on 15 different input visual features and found that the global “RgbV3H1” feature yields best results for both cases. All experiments were run on the same simulation and hardware conditions (Python, Intel Core i7). The comparison of accuracy and efficiency between independent SVMs and joint SVM is presented in Figure 1. While the learning and testing time of independent SVMs scale with the number of tags, the computation time of the joint SVM approximately equals a SVM for single-tag classification. At the same

	Training	Testing	Testing Performance		
	Time (sec)	Time (sec)	Precision	Recall	F1
Independent SVMs	6285.11	317.20	0.1049	0.1225	0.1130
Joint SVM (Gaussian)	80.68	**6.92**	**0.4078**	**0.3713**	**0.3887**
Joint SVM (Polynomial)	**76.48**	9.11	0.3908	0.3565	0.3728
The best result in [10]	–	–	0.27	0.32	0.292

Fig. 1: Performance of different algorithms.

time, in terms of accuracy, the joint SVM worked much better than independent SVMs. In addition, to test the higher-order dependency among tags, a 3rd-degree polynomial kernel function was also applied. The performance of this type kernel falls very close to that the Gaussian kernel provided. However, it is worth noting that the proposed joint SVM, with either Gaussian or polynomial kernel, outperforms many other state-of-the-art methods by a large margin (see a survey in [10]) on the same database.

4 Conclusions

A novel joint SVM was presented for automatic image tagging. Its superiority over conventional SVMs is obvious from our mathematical derivation and empirical results. Although, in this preliminary work, simple individual features and kernels already display good results, yet more improvements are expected when more features and sophisticated kernels, e.g. multi-kernel learning, are employed, which is a promising direction of future work.

References

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006.

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

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

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

[5] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *ICML*, 2005.

[6] Katja Astikainen, Liisa Holm, Esa Pitkänen, Sandor Szedmak, and Juho Rousu. Towards structured output prediction of enzyme function. In *BMC Proceedings, 2(4):S2*. 2008.

[7] Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

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

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

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