GSoC Proposal: A generic SO-learning framework in Shogun

Peter Romov

April 6, 2012

1 Structured Output Learning

For the structured output models we define a *discriminant function* $F(x, y)$ and use it to predict the answer $\hat{y}(x) = \arg\max_{y \in \mathcal{Y}} F(x, y)$. Structured output SVM (SO-SVM) [3] assume the discriminant function is from linear family, i.e.

$$F(x, y; w) = \mathbf{w}^\top \Psi(x, y),$$

here vector $\Psi(x, y)$ is called a *joint feature map*. For the structured output learning we also define a loss function $\Delta(y, \hat{y})$.

Given training examples $(x^1, y^1), \ldots, (x^N, y^N)$ and positive coefficient C. SO-SVM trains the weights vector $\mathbf{w}$ by solving

$$\frac{1}{2}\|w\|^2 + C\frac{1}{N}\sum_{n=1}^{N} l(w, x^n, y^n) \to \min_{\mathbf{w}}$$

where l is a generalization of hinge-loss constructed by margin-rescaling [3]

$$l(w, x^n, y^n) = \max_{y \in \mathcal{Y}} \Delta(y^n, y) + \mathbf{w}^\top \Psi(x^n, y) - \mathbf{w}^\top \Psi(x^n, y^n)$$

We could reformulate this in form of constrained quadratic program. Also we could use squared loss: l^2 instead of l.

SO-SVM has a kernelized dual formulation

$$\sum_{\substack{y \in \mathcal{Y} \\ n=1,\ldots,N}} \alpha_{ny}\Delta(y^n, y) - \frac{1}{2}\sum_{\substack{y \in \mathcal{Y} \\ n=1,\ldots,N}} \sum_{\substack{y' \in \mathcal{Y} \\ n'=1,\ldots,N}} \alpha_{ny}\alpha_{n'y'}\bar{K}^{nn'}_{yy'} \to \max_{\alpha \geq 0}$$

$$\text{s.t.:} \quad \sum_{y \in \mathcal{Y}} \alpha_{ny} \leq \frac{C}{N}, \quad n = 1, \ldots, N.$$

Here $\bar{K}$ is a structured analog of the kernel matrix computed using the positive defined *joint kernel function* $k(\cdot, \cdot)$

$$\bar{K}^{nn'}_{yy'} = K^{nn'}_{y^n y^{n'}} - K^{nn'}_{y^n y'} - K^{nn'}_{y y^{n'}} + K^{nn'}_{yy'}$$
$$K^{nn'}_{yy'} = k(\,(x^n, y), (x^{n'}, y')\,)$$

So the trained discriminant function can be expressed through the support pairs (x^n, y): $\alpha_{ny} > 0$

$$F(x,y) = \sum_{\substack{y' \in \mathcal{Y} \\ n=1,\ldots,N}} \alpha_{ny'} k(\,(x^n, y'), (x, y)\,)$$

As in binary $+1/-1$ classification the both primal and dual formulations are used. In some cases we need a non-linear decision rule but in other cases memorizing the support vectors is too hard. The linear SO-SVMs have been successfully used in many applications (e.g. in computer vision [2]).

2 Structured Models and SO-SVM Solvers

To build a generic structured output framework we need to know what is common between the different models. Also we need to know how different solvers use the model.

The well-known cutting plane methods use the following:

- *loss* value $\Delta(y, \hat{y})$,
- *loss-augmented prediction* (separation oracle)
  $y^*(\mathbf{w}) = \arg\underbrace{\max_{y \in \mathcal{Y}}\{\Delta(y^n, y) + \mathbf{w}^\top \Psi(x^n, y)\}}_{H(\mathbf{w})}$,
- if we do an approximate learning [1] we need *the bounds* (with their linear on $\mathbf{w}$ part) of the $y^*(w)$
  $\delta_{\text{under}} + \mathbf{w}^\top \psi_{\text{under}} \le H(\mathbf{w}) \le \delta_{\text{over}} + \mathbf{w}^\top \psi_{\text{over}}$,
- *joint feature map* $\Psi(x, y)$ for the linear SO-SVM
  or *joint kernel function* $k(x, y, \bar{x}, \bar{y})$ for the kernelized SO-SVM.

The structured model designed to learn with the cutting-plane solver should implement interface to these items.

3 Structured Output Machines in Shogun

Here is my thoughts about how to implement a general framework of structured learning for Shogun:

- The basic SO-learner class is CStructMachine (child of CSGObject).
- The class CLinearStructMachine inherits CStructMachine. It is intended to learn linear SO-SVM (so the inheritors solve the primal formulation of SO-SVM) and use instances of CJointFeatures as a train/test input).
- The class CKernelStructMachine is intended to solve dual formulation and use CJointKernel.

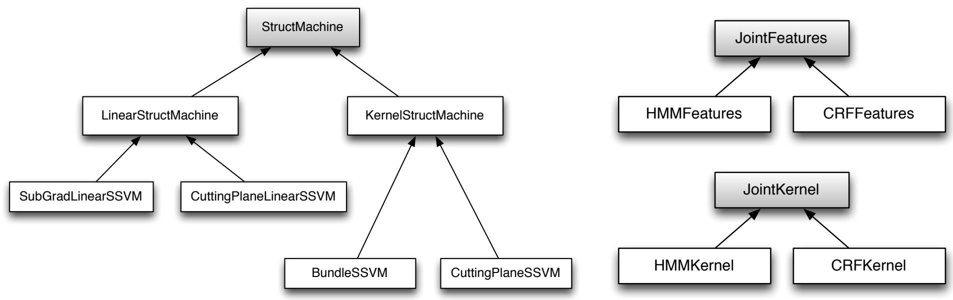

Figure 1: Proposed design of the structured learning framework.

- The classes CJointFeatures and CJointKernel implements the interface described in Section 2.

The proposed class hierarchy model is roughly illustrated on Fig. 1.

References

[1] Thomas Finley and Thorsten Joachims. Training structural svms when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 304–311, New York, NY, USA, 2008. ACM.

[2] Sebastian Nowozin and C.H. Lampert. *Structured Learning and Prediction in Computer Vision*. Now, 2011.

[3] I Tsochantaridis, T Joachims, T Hofmann, and Y Altun. Large margin methods for structured and interdependent output variables, 2005.