# Autoscanning for Coupled Scene Reconstruction and Proactive Object Analysis

Kai Xu[1,2] Hui Huang[1*] Yifei Shi[2] Hao Li[3] Pinxin Long[1] Jianong Caichen[2,1] Wei Sun[1] Baoquan Chen[3*]
[1]Shenzhen VisuCA Key Lab / SIAT [2]HPCL, National University of Defense Technology [3]Shandong University

## Appendix: Online cut cost learning with Multiple Kernel Learning

We first introduce the general method of Multiple Kernel Learning (MKL) [Bach et al. 2004]. Then, we show how to extend MKL for online learning, based on the passive-aggressive algorithm [Crammer et al. 2006]. Finally, we list the kernels and features we employed for cut cost learning.

## 1 Multiple Kernel Learning

Given a training data set:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^n,\, y_i \in \{-1, 1\}\}_{i=1}^n ,$$

where $\mathbf{x}_i$ is feature vector of $i$-th data point, and $y_i$, being $-1$ or $+1$, the indicator of the data point's binary class. The objective of Multiple Kernel Learning is to learn a prediction function:

$$f(\mathbf{x}) = \mathbf{w}^\top \phi_{\mathbf{d}}(\mathbf{x}) + b,$$

where $\mathbf{w}$ and $b$ are the parameters of a Support Vector Machine (SVM). The function involves a kernel $k_{\mathbf{d}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\mathbf{d}}^\top(\mathbf{x}_i)\phi_{\mathbf{d}}(\mathbf{x}_j)$. The kernel represents the dot product in feature space $\phi$ parameterized by $\mathbf{d}$ and is used to measure the similarity of data points. The goal in SVM learning is to learn the globally optimal parameters, the weight vector $\mathbf{w}$ and the bias $b$, from training data $\mathcal{D}$. This is achieved by solving the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i l(y_i, f(\mathbf{x}_i)),$$

where $l$ is loss function of the form $l = C\max(0, 1 - y_i f(\mathbf{x}_i))$ ($C$ is a constant). In MKL, besides the two parameters, we also need to optimize the kernel parameters $\mathbf{d}$, by solving the following minimization problem:

$$\min_{\mathbf{w},b,\mathbf{d}} \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i l(y_i, f(\mathbf{x}_i)) + r(\mathbf{d}) \quad \text{s.t.} \quad \mathbf{d} \geq 0$$

where both the regularizer $r$ and the kernel can be any general differentiable functions of $\mathbf{d}$ with continuous derivative. We use the $\ell_1$ regularizer proposed in [Chan et al. 2007].

## 2 Online Multiple Kernel Learning

The main feature of online learning is that the training data points arrive in a sequential manner. Let us denote the data point presented to the learning algorithm on round $t$ by $\mathbf{x}_t \in \mathbb{R}^n$, which is associated with an observed label $y_t \in \{-1, +1\}$. We refer to such instance-label pair $(\mathbf{x}_t, y_t)$ as online example. Online learning aims to make predictions using a prediction function learned incrementally with the online examples. In order to update the prediction function, which was learned based on historical examples, with the newly coming example, a guiding principle is to make the new prediction function "fits" the new example, while making minimal change to the original prediction function. By fitting, we mean that the new prediction function should give correct prediction for the new example.

The passive-aggressive algorithm proposed in [Crammer et al. 2006] is designed for SVM, aiming to find a new SVM prediction function based on a single example (corresponding to an online example) while ensuring it to remain as close as possible to the original one. This is achieved by constraining the weight vector $\mathbf{w}$ in SVM prediction function. Specifically, we set the weight vector $\mathbf{w}_{t+1}$ in round $t+1$, given the new example $(\mathbf{x}_t, y_t)$, to be the solution to the following constrained optimization problem:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad l(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0$$

The optimization has a simple closed form solution:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t \quad \text{with} \quad \tau_t = \frac{l_t}{\|\mathbf{x}_t\|^2}.$$

To extend the passive-aggressive algorithm to the MKL setting, we solve the following optimization problem:

$$\begin{aligned} \mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} & \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + \sum_i l(y_i, f(\mathbf{x}_i)) + r(\mathbf{d}) \\ \text{s.t.} \quad & \mathbf{d} \geq 0 \end{aligned} \tag{1}$$

The optimization of such problem can be solved by reformulating it as an interleaving optimization [Chapelle et al. 2002]. In the outer loop, the kernel is learned by optimizing over $\mathbf{d}$. In the inner loop, the kernel is held fixed and the $\mathbf{w}$ is optimized. See more details in [Chapelle et al. 2002].

## 3 Kernels and features

Our MKL uses *fourteen* kernels, which include eleven Gaussian kernels and three Polynomial kernels. The eleven Gaussian kernel we used are:

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}},$$

where $\sigma = 2^n$ with $n \in \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$. The three Polynomial kernels are:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^d,$$

with $d \in \{1, 2, 3\}$.

We use *six* features to describe the relations between two adjacent patches $P_u$ and $P_v$:

- **Dihedral angle**: We compute the dihedral angle between the two patches using their average normals:

$$x_1 = \cos\theta_{uv} = \mathbf{n}_u \cdot \mathbf{n}_v,$$

where $\mathbf{n}_u$ and $\mathbf{n}_v$ are the average normals of patch $P_u$ and $P_v$, respectively.

- **Dihedral angle convexity**: We measure the local convexity of the dihedral angle between $P_u$ and $P_v$:

$$x_2 = \kappa_{uv} = [(\mathbf{n}_u \times \mathbf{n}_v) \times (c_u - c_v)] \cdot \mathbf{n}_u$$

where $c_u$ and $c_v$ are the centers of patch $P_u$ and $P_v$, respectively.

*Corresponding authors: {hhzhiyan,baoquan.chen}@gmail.com

- **Difference of normal variance**: We compute the difference of the variance of normals in the two patches:

$$x_3 = |\Sigma_n(P_u) - \Sigma_n(P_v)|,$$

where $\Sigma_n$ is the variance of the normal directions of all points in a patch.

- **Difference of patch planarity**: We compare the patch planarity:

$$x_4 = |\pi(P_u) - \pi(P_v)|,$$

where patch planarity $\pi$ is measured as the average distance from all patch point to the least-square fitting plane of the patch.

- **Difference of patch size**: We compare the area of the two patches:

$$x_5 = |s(P_u) - s(P_v)|,$$

where $s(\cdot)$ counts the number of points in a patch, as an approximation to its area.

- **Difference of patch color distribution**: We compare the color distribution of the two patches:

$$x_6 = \chi^2(H_u - H_v) = \frac{1}{2}\sum_{k=1}^{K} \frac{[H_u(k) - H_v(k)]^2}{H_u(k) + H_v(k)},$$

where $H_u$ and $H_v$ are the color histograms of patch $P_u$ and $P_v$, respectively.

## References

BACH, F., LANCKRIET, G., AND JORDAN, M. 2004. Multiple kernel learning, conic duality, and the smo algorithm. In *Proc. ICML*, 1–6.

CHAN, A. B., VASCONCELOS, N., AND LANCKRIET, G. 2007. Direct convex relaxations of sparse svm. In *Proc. ICML*, 145–153.

CHAPELLE, O., VAPNIK, V., BOUSQUET, O., AND MUKHERJEE, S. 2002. Choosing multiple parameters for support vector machines. *Machine Learning 46*, 131–159.

CRAMMER, K., DEKEL, O., KESHET, J., SHALEV-SHWARTZ, S., AND SINGER, Y. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res. 7* (Dec.), 551–585.