Randomized Support Vector Forest

Xutao Lv[1]
xutao.lv@sri.com
Tony X. Han[2]
hantx@missouri.edu
Zicheng Liu[3]
zliu@microsoft.com
Zhihai He[2]
hezhi@missouri.edu

[1] SRI International
Princeton, NJ, USA
[2] University of Missouri
Columbia, MO, USA
[3] Microsoft Research
Seatle, WA, USA

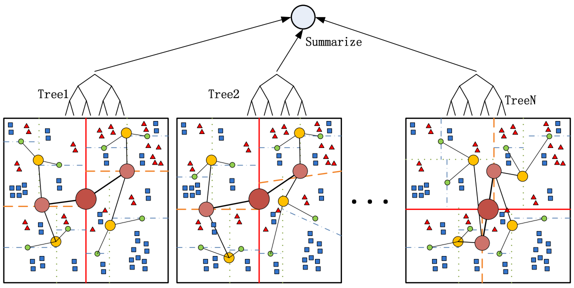

Figure 1: The structure of RSVF. This figure shows a RSVF with N trees. Each tree, with depth 5, is demonstrated in the last row of the figure. The small green dots are the LSVM classifiers; the other dots are the binary classifiers. Note, the binary classifiers mentioned in this paper represent decision nodes, which use a threshold to split the data into two child nodes.

Based on the structural risk minimization principle, the linear SVM aiming at finding the linear decision plane with the maximal margin in the input space has gained increasing popularity due to its generalizability, efficiency and acceptable performance. However, rarely training data are evenly distributed in the input space [1], which leads to a high global VC confidence [3], downgrading the performance of the linear SVM classifier. Partitioning the input space in tandem with local learning may alleviate the unevenly data distribution problem. However, the extra model complexity introduced by partitioning frequently leads to overfitting.

To solve this problem, we proposed a new supervised learning algorithm, Randomized Support Vector Forest (RSVF): Many partitions of the input space are constructed with partitioning regions amenable to the corresponding linear SVMs.

As illustrated in Figure 1, the RSVF consists of many Support Vector Trees (SVT). Each SVT represents a scheme of data partition and the corresponding local classifier. The final classification result of RSVF is a pooling from all the SVTs. After comparing various pooling methods including the majority voting, and max voting, i.e., taking the prediction from the SVT with the maximal confidence, we use majority voting from all of the trees in the forest for its simplicity and efficacy. We grow the RSVF through a procedure similar to growing the Classification And Regression Trees (CART) in random forest [7]. The steps of building RSVF is shown in Algorithm 1.

Input: Training dataset $\mathcal{X}$ and the number of trees N_{tree} in RSVF
Output: RSVF
for $t \leftarrow 1$ **to** N_{tree} **do**
  Randomly sample the bootstrap dataset $\mathcal{X}^*$ from $\mathcal{X}$; the Out-Of-Bag data will be $\mathcal{X} \backslash \mathcal{X}^*$;
  Train the SVTs $\mathcal{T}$ with both dataset $\mathcal{X}^*$ and $\mathcal{X} \backslash \mathcal{X}^*$;
end

Algorithm 1: Building RSVF

The generalization of the RSVF benefits from the randomness injected through random feature selection and bagging, which is also essential to the generalization of random forests [2].

The randomness of the partitions is injected through random feature selection and bagging. This partition randomness prevents the overfitting introduced by the over-complicated partitioning. With the injected randomness, the generalization error of RSVF can be proved to converge almost surely using the Law of Large Numbers when the number of SVTs increases. As the number of trees in RSVF increases, for almost surely all Θ, the generalization error e_g of RSVF converges to,

$$P_{\mathbf{X},Y}(P_\Theta(\mathcal{T}(\mathbf{X},\Theta)=Y)-\max_{j\neq Y} P_\Theta(\mathcal{T}(\mathbf{X},\Theta)=j<0) \quad (1)$$

where $\mathcal{T}$ is an SVT; $\mathbf{X}$ is feature matrix; Y is the label of $\mathbf{X}$; and Θ is a set of parameters ϕ^* associated with the SVT $\mathcal{T}$.

We extensively evaluate the performance of the RSVF on several benchmark datasets, originated from various vision applications, including the four UCI datasets, the letter dataset, the KTH and the UCF sports dataset, and the Scene-15 dataset. The performance is shown in Table 1 and Table 2. The proposed RSVF outperforms linear SVM, kernel SVM, Random Forests (RF), and a local learning algorithm, SVM-KNN, on all of the evaluated datasets. The classification speed of the RSVF is comparable to linear SVM.

Method	LSVM	RF	RSVF	SVM-KNN	χ^2-KSVM	RBF-KSVM
KTH*	92.59%	91.67%	**93.98%**	87.04%	92.59%	92.13%
UCF	65.7 ± 5.8%	61.5 ± 7.3%	**72.2 ± 5.4%**	48.4 ± 5.6%	66.3 ± 6.6%	62.3 ± 6.7%
Scene15	75.1 ± 0.3%	63.3 ± 0.9%	**78.3 ± 0.4%**	59.9 ± 0.9%	76.9 ± 0.4%	75.7 ± 0.6%

Table 1: Recognition accuracy on KTH, Scene-15 and UCF sports datasets. *Note: since the training and the testing sets are fixed in the KTH dataset, we just follow the standard setup so that our result can be compared with [4, 5, 6, 9].

Type	Best in [8]	Linear SVM	RBF-SVM	RSVF	RF
dna	0.059 ± 0.005	0.088 ± 0.017	0.054 ± 0.010	**0.052 ± 0.008**	0.056 ± 0.011
wine	0.030 ± 0.029	0.023 ± 0.024	0.016 ± 0.022	**0.002 ± 0.007**	0.014 ± 0.016
iris	0.057 ±0.022	0.038 ± 0.026	0.032 ± 0.025	**0.029 ±0.048**	0.041 ±0.029
glass	0.232 ±0.047	0.408 ± 0.091	0.300 ± 0.059	**0.223 ± 0.068**	0.234 ± 0.055

Table 2: Performance comparison on UCI datasets. The results in the first column is obtained from [8].

[1] Leon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural Computation*, 4:888–900, 1992.

[2] Leo Breiman and E. Schapire. Random forests. In *Machine Learning*, volume 45, pages 5–32, 2001.

[3] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72, 2005.

[5] Andrew Gilbert, John Illingworth, and Richard Bowden. Action recognition using mined hierarchical compound features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):883–897, 2011.

[6] Nazli Ikizler-Cinbis and Stan Sclaroff. Object, scene and actions: combining multiple features for human action recognition. In *Proceedings of the 11th European conference on Computer vision: Part I*, ECCV'10, pages 494–507, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15548-0, 978-3-642-15548-2.

[7] Daniel F. Schwarz, Inke R. König, and Andreas Ziegler. On safari to random jungle: a fast implementation of random forests for high-dimensional data. *Bioinformatics*, 27(3):439, 2011.

[8] Chunhua Shen and Zhihui Hao. A direct formulation for totally-corrective multi-class boosting. In *CVPR*, pages 2585–2592, 2011.

[9] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.