Discriminative tree-based feature mapping

Miroslav Kobetski
http://www.csc.kth.se/~kobetski/
Josephine Sullivan
http://www.csc.kth.se/~sullivan/

Computer Vision Group, CVAP
KTH, Royal Institute of Technology
Teknikringen 14,
114 28 Stockholm

We approach the classification and detection problem by viewing each step in the classification pipeline as a feature mapping. Good discriminative features $x(I)$ (such as HOG or LBP) can be seen as functions that map from pixel intensities I to a feature space where classes are more easily separable. Mixture components and parts [1] can be thought of as an additional feature mapping layer $\Phi(x,z)$ where the mapping function is also parametrised by component and part positions z. Also a non-linear kernel-based classifier can be seen as a linear classifier in an implicitly mapped space $\Phi(x)$. Finite explicit approximations to a number of kernels have been proposed by other works [3, 4, 5].

Rather that trying to approximate a kernel, we present explicit approximate mappings of tree-based classifiers learnt specifically for a particular problem and feature set. Tree-based classifiers such as boosted ensembles or random forests are powerful and fast, but are not easy to integrate with other well-developed methods that rely on explicit feature representations. In this paper we propose a more direct approach for finding a low-dimensional feature mapping $\Phi(x)$ to a space where the object and background classes are more easily separated by a linear hyperplane; having a well-generalizing underlying classifier is important for this separability to translate to unseen examples. Rather than finding a kernel with these properties and approximating it, we directly learn a non-linear decision boundary using boosted decision trees. We then induce the new feature dimensions from the decision rules of the trees. We present two of our methods for inducing $\Phi(x)$ from trees. We show that they increase discriminative performance compared to our baseline, at a small cost of evaluating the $\Phi(x)$ function. Linear SVMs in the tree-mapped space even outperform the original tree-based classifiers.

One main reason for inducing a feature mapping rather than using the trees directly is that many existing methods and frameworks (many generative models, clustering methods, detection frameworks) require an explicit feature representation. Hard negative mining is another example of a very important technique that is well defined for linear SVMs [1], but more of an unstable art-form for tree ensembles.

Each feature dimension $z_j = \phi_j(x)$ of our mapped feature vector $z = \Phi(x)$ encodes the decision path S_j taken to one leaf node of the tree. Starting at the root, the path to a leaf node can be seen as a sequence of decisions based on a number of feature values x_i and thresholds α_i, where x_i is the i:th dimension of feature vector x. We define

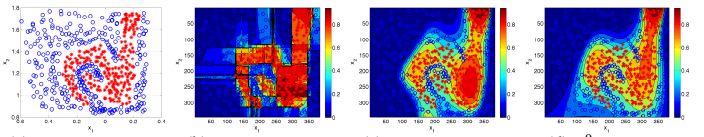
$$S_j(x) = \prod_{k=1}^{n} \delta_{jk}(x_{j_k}, \alpha_{j_k}), \tag{1}$$

where $\delta_{jk} \in \{\delta^+, \delta^-\}$ encode binary decisions $\delta^+(z_1,z_2) = [z_1 \geq z_2]$ and $\delta^-(z_1,z_2) = [z_1 < z_2]$, described using Iverson bracket notation. Such rules are inherently binary, so we enrich our feature mappings by also considering the distance to the decision boundary or rule margin $m_j(x)$. Introducing the concept of rule margin adds some ambiguity to the mapping - since for a rule j, $m_j(x)$ is the accumulated score of a number of decisions and can be envisioned and implemented in a number of ways. We present two different ones - *Leaf node decision mapping* (LDM), that most closely resembles the space partitioning of the underlying trees and *Polynomial cross-term mapping* (PCM) that can be related to sparse polynomial cross terms.

LDM is the most straight-forward addition of $m_j(x)$ to the otherwise discrete tree mapping. For a leaf node of depth n, the mapping simply becomes

$$\phi_j(x) = S_j(x)(x_{j_n} - \alpha_{j_n}). \tag{2}$$

The non-linearity is encoded through the binary decisions made following the path $S_j(x)$. An example receives value zero if it fails any of the binary decisions. If all decisions are true the example gets a value signifying its signed distance to the decision threshold made just prior to the leaf node. Each added dimension can be seen as a partition of space where similar examples are ordered according to feature x_{j_n} and dissimilar examples (as they have failed earlier similarity tests) are ignored. Due to this partitioning property LDM closely resembles the decision rules from which it is induced.



(a) Original 2D problem (b) LDM + linear SVM (c) PCM + linear SVM (d) $\Phi^9_{poly}(x)$ L1-reg

Figure 1: 2D toy problem where the 2D points have been mapped using tree-based feature mapping, after which a linear SVM was learned and projected back to 2D. $\Phi^d_{poly}(x)$ is the feature map of a d-degree polynomial kernel where each dimension is a polynomial term. The color values represent the scores of the linear SVM.

Mapping	mAP	improvement
Linear	64.85	0.00
χ^2-mapped	69.10	4.25
Polynomial kernel	70.09	5.24
Boosted Trees	72.68	7.83
LDM	72.76	7.91
PCM	72.88	8.02
CTM	74.01	9.16

Table 1: VOC2007 patch classification performance for a single HOG template and linear SVM with various feature mappings. The proposed mappings LDM, PCM and CTM outperform the other methods, including the boosted tree classifier they were induced from.

PCM accumulates the signed distance to each of the splits along $S_j(x)$

$$\phi_j(x) = \prod_{k=1}^{n} (2\delta_{j_k} - 1)(x_{j_k} - \alpha_{j_k}). \tag{3}$$

This can be seen as selecting a subset of all cross terms of a polynomial kernel, where the depth of the tree determines the degree of the polynomial. This mapping is also much sparser than a polynomial kernel since only the cross terms of features that jointly produce discriminative decisions for the particular class are selected.

We perform image patch classification experiments on the VOC2007 and INRIAPerson datasets and image classification on VOC2007. Table 1 shows that both PCM and LDM mappings improve patch classification results significantly and that the best performance is obtained by combining the tree mappings (CTM). For image classification the kernel-approximating mappings (χ^2 and Hellinger) that we used are theoretically well suited, which is reflected in the results - χ^2 improves mAP by 5.13% where while the combined CTM mapping reaches 4.68% improvement.

[1] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), September 2010.

[2] J.H. Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, 28, 2000.

[3] Subhrasnu Maji. Linearized smooth additive classifiers. In *eccv, Workshop on Web-scale Vision and Social Media*, 2012.

[4] Subhrasnu Maji, Alexander C. Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[5] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3), 2011.