Supplementary Materials: Weighted Substructure Mining for Image Analysis

Sebastian Nowozin, Koji Tsuda
Max Planck Institute for Biological Cybernetics
Spemannstrasse 38, 72076 Tübingen, Germany
{sebastian.nowozin,koji.tsuda}@tuebingen.mpg.de

Takeaki Uno
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan
uno@nii.jp

Taku Kudo
Google Japan Inc., Cerulean Tower 6F, 26-1 Sakuragaoka-cho, Shibuya-ku, Tokyo, 150-8512, Japan
taku@google.com

Gökhan Bakır
Google GmbH, Freigutstrasse 12, 8002 Zurich, Switzerland
ghb@google.com

1. LPBoost

Some details have been omitted from the main presentation for space reasons and clarity of presentation. Here we additionally provide a description of the LPBoost algorithm and a detailed motivation and derivation of the 1.5-class ν-LPBoost variant.

We also provide a larger image of the unsupervised ranking results.

1.1. LPBoost Algorithm

The LPBoost algorithm is summarized in Algorithm 1. We use $\mathcal{H}$ to denote the space of possible hypothesis. For weighted substructure mining applications this is $\mathcal{H} = \{h(\cdot; \boldsymbol{t}, \omega) | (\boldsymbol{t}, \omega) \in \mathcal{T} \times \Omega\}$. We denote by h_i the hypothesis selected at iteration i.

Algorithm 1 Linear Programming Boosting (LPBoost)

Input: Training set $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_\ell\}, \boldsymbol{x}_i \in \mathcal{X}$,
labels $Y = \{y_1, \dots, y_\ell\}, y_i \in \{-1, 1\}$.
convergence threshold $\theta \geq 0$.
Output: The classification function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$.

$\lambda_n \leftarrow \frac{1}{\ell}, \forall n = 1, \dots, \ell$
$\gamma \leftarrow 0, J \leftarrow 1$
loop
  $\hat{h} \leftarrow \text{argmax}_{h \in \mathcal{H}} \sum_{n=1}^{\ell} y_n \lambda_n h(\boldsymbol{x}_n)$
  if $\sum_{n=1}^{\ell} y_n \lambda_n \hat{h}(\boldsymbol{x}_n) \leq \gamma + \theta$ **then**
    break
  end if
  $h_J \leftarrow \hat{h}$
  $J \leftarrow J + 1$
  $(\boldsymbol{\lambda}, \gamma) \leftarrow$ solution to the dual of the LP problem, where γ is the objective function value.
  $\boldsymbol{\alpha} \leftarrow$ Lagrangian multipliers of solution to dual LP problem
end loop
$f(\boldsymbol{x}) := \text{sign}\left(\sum_{j=1}^{J} \alpha_j h_j(\boldsymbol{x})\right)$

1.2. 1.5-class LPBoost

In Figures 1-3 the behaviour of the 1-class, 2-class and 1.5-class classifiers is shown schematically for a 2D toy example.

Given a set of positive samples $X_1 = \{\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,N}\}$

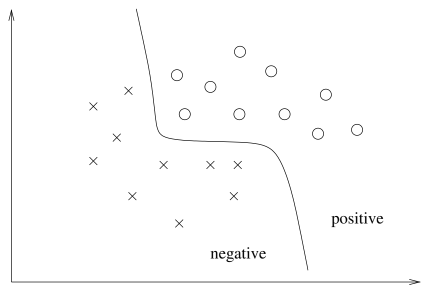

Figure 1. 2-class classifier: Learning a separation of positive and negative samples in feature space.

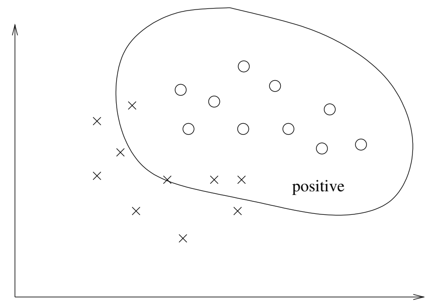

Figure 2. 1-class classifier: Learning a description of the positive class in feature space using the positive training samples only.

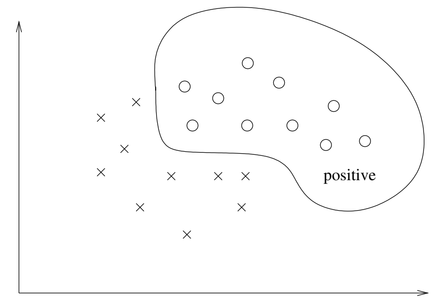

Figure 3. 1.5-class classifier: Learning a description of the positive class in feature space using both positive and negative training samples.

Figure 4. Separation as achieved by the 1.5-class LPBoost formulation: the positive samples are separated from the origin by ρ_1, while the output for negative samples is kept below ρ_2.

and a set of negative samples $X_2 = \{\boldsymbol{x}_{2,1}, \ldots, \boldsymbol{x}_{2,M}\}$, we derive the following new "1.5-class LPBoost" formulation.

$$
\begin{aligned}
\min \quad & \rho_2 - \rho_1 + \frac{1}{\nu N}\sum_{n=1}^{N} \xi_{1,n} + \frac{1}{\nu M}\sum_{m=1}^{M} \xi_{2,m} && (1)\\
\text{sb.t.} \quad & \sum_{\boldsymbol{t}\in\mathcal{T}} \alpha_{\boldsymbol{t}} h(\boldsymbol{x}_{1,n};\boldsymbol{t}) \geq \rho_1 - \xi_{1,n},\ n = 1,\ldots,N \\
& \sum_{\boldsymbol{t}\in\mathcal{T}} \alpha_{\boldsymbol{t}} h(\boldsymbol{x}_{2,m};\boldsymbol{t}) \leq \rho_2 + \xi_{2,m},\ m = 1,\ldots,M \\
& \sum_{\boldsymbol{t}\in\mathcal{T}} \alpha_{\boldsymbol{t}} = 1, \\
& \boldsymbol{\alpha} \in \mathbb{R}_+^{|\mathcal{T}|}, \rho_1, \rho_2 \in \mathbb{R}_+, \boldsymbol{\xi}_1 \in \mathbb{R}_+^N, \boldsymbol{\xi}_2 \in \mathbb{R}_+^M
\end{aligned}
$$

where we directly maximize a soft-margin $(\rho_1 - \rho_2)$ that separates positive from negative training samples as illustrated in Figure 4. The hypotheses are decision stumps that reward the presence of a pattern:

$$
h(\boldsymbol{x};\boldsymbol{t}) = \begin{cases} 1 & \boldsymbol{t} \subseteq \boldsymbol{x} \\ 0 & \text{otherwise.} \end{cases} \qquad (2)
$$

The class decision function is given by thresholding at the margin's center $(\rho_1 + \rho_2)/2$, such that

$$
f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{\boldsymbol{t}\in\mathcal{T}} \alpha_{\boldsymbol{t}} h(\boldsymbol{x};\boldsymbol{t}) - \frac{\rho_1 + \rho_2}{2} \right). \qquad (3)
$$

Problem (1) can be solved by the LPBoost algorithm [1] using the following dual LP problem.

$$
\begin{aligned}
\max_{\boldsymbol{\lambda},\boldsymbol{\mu},\gamma} \quad & -\gamma && (4)\\
\text{sb.t.} \quad & \sum_{n=1}^{N} \lambda_n h(\boldsymbol{x}_{1,n};\boldsymbol{t}) - \sum_{m=1}^{M} \mu_m h(\boldsymbol{x}_{2,m};\boldsymbol{t}) \leq \gamma, \\
& \qquad \boldsymbol{t} \in \mathcal{T} && (5)\\
& \sum_{n=1}^{N} \lambda_n \geq 1 \\
& \sum_{m=1}^{M} \mu_m \leq 1 \\
& 0 \leq \lambda_n \leq \frac{1}{\nu N}, \quad n = 1,\ldots,N \\
& 0 \leq \mu_m \leq \frac{1}{\nu M}, \quad m = 1,\ldots,M.
\end{aligned}
$$

For solving Problem (4) we again use column-generation techniques, incrementally adding the most violated constraint.

Similarly to the original 2-class LPBoost formulation we derive the gain function from the constraints on the hypotheses outputs of the dual of (1) to obtain

$$
\hat{h} = \operatorname*{argmax}_{h\in\mathcal{H}} \left[\sum_{n=1}^{N} \lambda_n h(\boldsymbol{x}_{1,n}) - \sum_{m=1}^{M} \mu_m h(\boldsymbol{x}_{2,m}) \right], \qquad (6)
$$

which is the same as for the 2-class case, except that the set of samples are explicitly split into two sums. For performing weighted substructure mining efficiently we need a bound on the gain for a pattern $\boldsymbol{t}'$. The bound shall be evaluated using only $\boldsymbol{t}$, where $\boldsymbol{t} \subseteq \boldsymbol{t}'$ is subpattern of $\boldsymbol{t}'$; this allows efficient pruning in the mining algorithm. For the new formulation we derive the following new bound on the gain function. Using the anti-monotonicity property [2] for any $\boldsymbol{t} \subseteq \boldsymbol{t}'$ we have

$$\begin{aligned}
gain(\boldsymbol{t}') &= \sum_{n=1}^{N} \lambda_n h(\boldsymbol{x}_{1,n}; \boldsymbol{t}') - \sum_{m=1}^{M} \mu_m h(\boldsymbol{x}_{2,m}; \boldsymbol{t}') \\
&= \sum_{n=1}^{N} \lambda_n I(\boldsymbol{t}' \subseteq \boldsymbol{x}_{1,n}) - \sum_{m=1}^{M} \mu_m I(\boldsymbol{t}' \subseteq \boldsymbol{x}_{2,m}) \\
&\leq \sum_{n=1}^{N} \lambda_n I(\boldsymbol{t}' \subseteq \boldsymbol{x}_{1,n}) \\
&\leq \sum_{n=1}^{N} \lambda_n I(\boldsymbol{t} \subseteq \boldsymbol{x}_{1,n}).
\end{aligned}$$

A drawback of the new formulation (1) is the violation of the closed-under-complementation assumption of Demiriz et al. [1], hence we are not guaranteed to obtain the optimal solution $(H, \boldsymbol{\alpha})$ among all possible sets and weightings. In practice this never caused any problems and the convergence behavior measured by an independent test error is very similar to the 2-class case.

The 1.5-class formulation (1) is a generalization of 1-class ν-Boosting in Rätsch et al. [3] and we recover the original formulation when $M = 0$, that is, when no negative samples are available.

References

[1] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Journal of Machine Learning*, 46:225–254, 2002.

[2] S. Morishita. Computing optimal hypotheses efficiently for boosting. In S. Arikawa and A. Shinohara, editors, *Progress in Discovery Science*, volume 2281 of *Lecture Notes in Computer Science*, pages 471–481. Springer, 2002.

[3] G. Rätsch, B. Schölkopf, S. Mika, and K.-R. Müller. SVM and boosting: One class. Technical report, 2000.

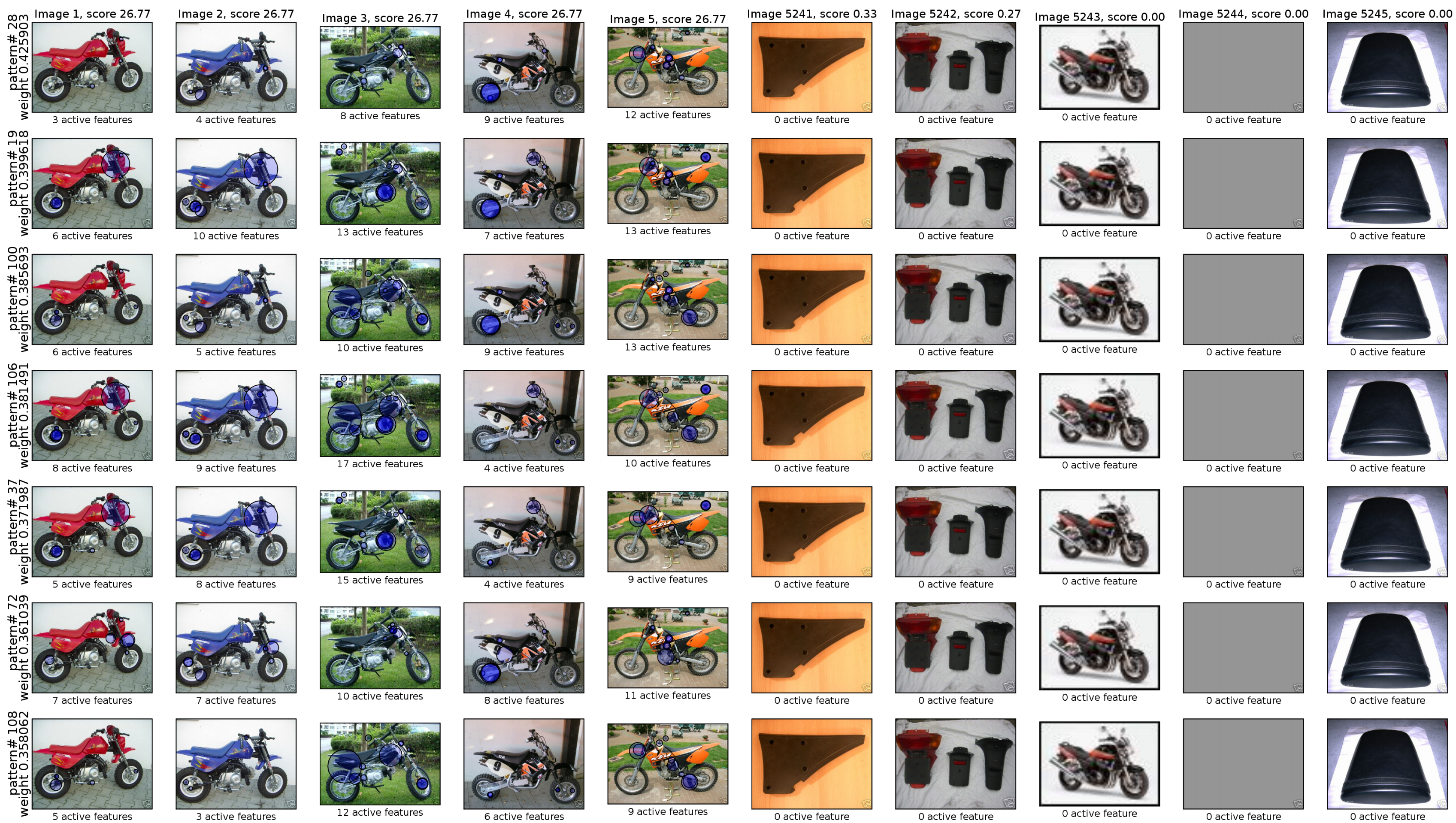

Figure 5. Top seven most influential patterns for the unsupervised ranking task. (This is an enlarged version of the figure in the paper.)