Unsupervised One-Class Learning for Automatic Outlier Removal

Wei Liu[†] Gang Hua[†‡] John R. Smith[†]
[†]IBM T. J. Watson Research Center [‡]Stevens Institute of Technology
{weiliu,jsmith}@us.ibm.com ghua@stevens.edu

Abstract

Outliers are pervasive in many computer vision and pattern recognition problems. Automatically eliminating outliers scattering among practical data collections becomes increasingly important, especially for Internet inspired vision applications. In this paper, we propose a novel one-class learning approach which is robust to contamination of input training data and able to discover the outliers that corrupt one class of data source. Our approach works under a fully unsupervised manner, differing from traditional one-class learning supervised by known positive labels. By design, our approach optimizes a kernel-based max-margin objective which jointly learns a large margin one-class classifier and a soft label assignment for inliers and outliers. An alternating optimization algorithm is then designed to iteratively refine the classifier and the labeling, achieving a provably convergent solution in only a few iterations. Extensive experiments conducted on four image datasets in the presence of artificial and real-world outliers demonstrate that the proposed approach is considerably superior to the state-of-the-arts in obliterating outliers from contaminated one class of images, exhibiting strong robustness at a high outlier proportion up to 60%.

1. Introduction

A lot of recent vision research has exploited a massive number of images from the Internet as a source of training data for building learning models, including learning object categories [8], query-related visual classifiers [12], query-adaptive graph rankers [18], query-specific semantic features [28], *etc*. In a typical Internet image powered application, one can crawl abundant images corresponding to a textual query (*e.g*., an object name or a semantic concept) by querying into web image search engines such as Google and Bing, or photo sharing websites such as Flickr, and then build a model for the target object/concept associated with the image collection. However, such images gathered via web crawling are often noisy, which could compromise the learning model. Therefore, pruning the irrelevant images, *i.e*., the *outliers*, becomes necessary.

Existing methods for outlier removal or detection either construct a profile about normal data examples and then identify the examples not conforming to the normal profile as outliers [6, 10, 24, 26], or explicitly isolate outliers based on statistical or geometric measures of abnormality [7, 13, 17]. A variety of methods can be found in the survey [32]. It is noticeable that most prior methods adopted the "few and different" assumption about the outlierness nature, so a relatively large fraction of outliers approaching 50% may lead to impaired performance.

To better mitigate this issue, we propose a novel approach for outlier removal, which automatically eliminates the outliers from a corrupted dataset so that the remaining samples belong to one or multiple dense regions, *e.g*., manifolds or clusters. Our approach attempts to cope with the high outlier level through equally concerning normal examples and outliers, and then formulates an elegant learning model to unify normal characterization, achieved by one-class classifier learning, and outlier detection.

As a tight connection, our approach falls into the *One-Class Learning* paradigm [24, 26], but differs in the input data configuration. One-class learning (or classification) is engaged in distinguishing one class of target data objects from all other possible objects, through learning a classifier with a training set merely containing the examples from the target class. Such a learning problem is more difficult than a conventional supervised learning problem due to the absence of negative examples. Since negative examples may often be insufficient due to difficult acquisition or costly manual labeling, one-class learning is usually favored and has found its usage in broad applications including image retrieval [5], document classification [20], web page classification [31], and data stream mining [16].

Nonetheless, existing one-class learning methods such as the representative *One-Class Support Vector Machine* [24] did not explicitly handle uncertain input data, though they can tolerate a small quantity of outliers. Another well-known method *Support Vector Data Description* [26] found that when integrating outlier examples into one-class learning, classification accuracy will be boosted. However, more often we are not aware of the outliers beforehand for a learn-

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.483

ing task. Assuming no special input configuration, our approach deals with an uncertain data mixture, where neither positive samples nor outliers are labeled in advance.

Our approach brings a new insight into the commonly investigated outlier detection problem from a learning perspective. The core idea is to tailor an *unsupervised* learning mechanism to tackle uncertainty of input data, where the dubious outliers are gradually discovered via a self-guided labeling procedure and then separated from the trustable positive samples by training a large margin one-class classifier. In doing so, our approach not only enables effective outlier removal but also yields a confidence value for any asserted "positive" sample.

Consequently, the proposed one-class learning approach can readily be applied to two emerging Internet vision applications: web image tag denoising [19] and web image search re-ranking [12, 18]. For tag denoising, the identified outlier images should not take the respective tag; for re-ranking, the entire images are re-ranked according to their confidence values produced by our approach. Extensive experiments carried out on three public image databases and a new web image database collected by ourselves with artificial and real-world outliers show remarkable performance gains of our approach over the state-of-the-arts in outlier removal and one-class learning.

2. Related Work

There has been substantial previous work concentrating on outlier detection or removal, investigated by various areas ranging from computer vision to data mining. We summarize these methodologies into three major categories.

From a geometric point of view, the first category of methods exploit sample reconstruction to do outlier detection. In particular, one can reconstruct a sample using the principal subspace acquired by PCA, Kernel PCA, Robust PCA [4, 14, 29], and Robust Kernel PCA [22, 29], or the representatives [7] which summarize the entire dataset. The outliers are thus identified as the samples taking on high reconstruction residues. This style of methods still follow the basic "few and different" assumption about outliers.

The second category of methods treat the outlier detection problem as a probabilistic modeling process, resulting in an outlierness measure based on a probability density function. Then the samples with low probability densities are judged to be outliers. Complying with this principle, tremendous probability density estimation schemes have been explored, including parametric estimators [32] and nonparametric estimators such as the kernel density estimator (KDE) (*i.e*., the classical *Parzen-Rosenblatt window* method [23]) and the more recent robust kernel density estimator (RKDE) [11].

Instead of exploring the characteristics of outliers, the third category describes normal objects through learning a compact data model such that as many as possible normal samples are enclosed inside. Two commonly used models are hyperplane and hypersphere. The former was proposed by one-class support vector machines (OC-SVMs) [24], and the latter was advocated by support vector data description (SVDD) [26]. It has been proven that OC-SVMs and SVDD are essentially equivalent when stationary kernels ($k(\boldsymbol{x}, \boldsymbol{x})$ is a constant) are adopted [24]. Though having shown advantages, they both need a clean training set consisting of known normal examples, labeled by "positive", to learn the hyperplane and hypersphere models. If the training set is corrupted with a relatively large fraction (*e.g*., 50%) of outliers, the performance of both OC-SVMs and SVDD is very likely to deteriorate because they do not explicitly handle outliers during model training.

Let us use a toy example to illustrate the outlier issue of OC-SVMs and SVDD. Fig. 1(a) shows a 2D toy dataset, where the outliers stem from uniformly distributed random noise and the groundtruth positive (*i.e*., normal) samples are within a big and dense cluster located at the center. Note that the outlier proportion is as high as 50%. Taking this corrupted dataset as a training set, OC-SVM is biased to the outliers near the boundary of the normal points, and comparable with RKDE in precision of the judged positive samples, as revealed by Figs. 1(b)(c). Note that we use a Gaussian kernel so OC-SVM and SVDD output the same result. In contrast, our proposed unsupervised one-class learning (UOCL) approach (see Fig. 1(d)) achieves the highest precision, thereby exhibiting strong robustness at the high outlier level.

In machine learning, there exist some approaches related to the uncertain learning paradigm discussed in this paper, including supervised binary class learning with label uncertainty [3, 30], semi-supervised one-class learning with positive and unlabeled examples [15, 21], unsupervised learning of two normally distributed classes [1], *etc*. All these are beyond the scope of this paper.

3. Unsupervised One-Class Learning

In this section, we tackle the outlier present one-class learning problem by proposing a reliable unsupervised learning model to automatically obliterate outliers from a corrupted training dataset. The proposed model is built upon two intuitive assumptions: 1) outliers originate from low-density samples, and 2) neighboring samples tend to have consistent classifications. We specially term this model *Unsupervised One-Class Learning* (UOCL) and design a provably convergent algorithm to solve it.

3.1. Learning Model

Given an unlabeled dataset $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$, we pursue a classification function $f : \mathbb{R}^d \mapsto \mathbb{R}$ similar to OC-SVMs. By leveraging a kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$

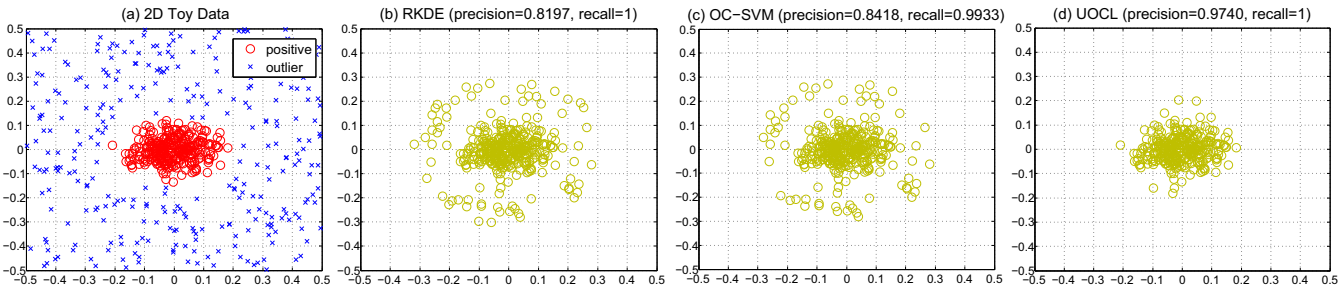

Figure 1. The outlier removal results on a 2D toy dataset where the percentage of outliers is 50%.

that induces the *Reproducing Kernel Hilbert Space* (RKHS) $\mathcal{H}$, the Representer theorem [25] states that the target classification function is in the following expression:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \kappa(\boldsymbol{x}, \boldsymbol{x}_i)\alpha_i, \tag{1}$$

where α_i is the expansion coefficient contributed by the functional base $\kappa(\cdot, \boldsymbol{x}_i)$. Let us introduce a *soft* label assignment $\mathcal{Y} = \{y_i \in \{c^+, c^-\}\}_{i=1}^n$ to the input data $\mathcal{X}$, which takes on a positive value c^+ for the positive samples whereas a negative value c^- for the outliers. Let $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$ be the vector representation of $\mathcal{Y}$. The use of soft labels will help handle a high fraction of outliers, as validated later.

Now we establish the UOCL model as minimizing the following objective:

$$\begin{aligned} \min_{f\in\mathcal{H}, \{y_i\}} & \sum_{i=1}^{n} \big(f(\boldsymbol{x}_i) - y_i\big)^2 + \gamma_1 \|f\|_{\mathcal{M}}^2 - \frac{2\gamma_2}{n^+} \sum_{i, y_i>0} f(\boldsymbol{x}_i) \\ \text{s.t.} \quad & y_i \in \{c^+, c^-\}, \ \forall i \in [1:n], \\ & 0 < n^+ = \big|\{i | y_i > 0\}\big| < n, \end{aligned} \tag{2}$$

where $\gamma_1, \gamma_2 > 0$ are two trade-off parameters controlling the model. Note that we impose the constraint $0 < n^+ < n$ to discard two extreme cases: no positive sample and full positive samples. In order to remove the influence of varying $\|\boldsymbol{y}\|^2 = \sum_{i=1}^n y_i^2$ on the optimization in Eq. (2), we design the values (c^+, c^-) of soft labels such that $\|\boldsymbol{y}\|^2$ is constant. For instance, $(1, -1)$, $(\sqrt{\frac{n}{2n^+}}, -\sqrt{\frac{n}{2(n-n^+)}})$, or $(\sqrt{\frac{n-n^+}{n^+}}, -\sqrt{\frac{n^+}{n-n^+}})$ satisfies this requirement. It is worthwhile to point out that optimizing Eq. (2) using other soft labels of unfixed $\|\boldsymbol{y}\|^2$ could lead to trivial solutions, *e.g.*, the soft labels $(\frac{1}{n^+}, -\frac{1}{n-n^+})$ always result in $n^+ \approx \lfloor n/2 \rfloor$.

Since neither positive samples nor outliers are known before the learning task, UOCL is fully unsupervised and hence uses the squared loss $(f(\boldsymbol{x}_i) - y_i)^2$ instead of a hinge loss used by many (semi-)supervised learning models where exact labels are needed.

The term $\|f\|_{\mathcal{M}}^2$ in Eq. (2) is the manifold regularizer [2], which turns out to endow f with the smoothness along the intrinsic manifold structure $\mathcal{M}$ underlying the data cloud $\mathcal{X}$. We construct this term by making use of a neighborhood graph G whose affinity matrix is defined by

$$W_{ij} = \begin{cases} \exp\left(-\dfrac{\mathcal{D}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\varepsilon^2}\right), & i \in \mathcal{N}_j \text{ or } j \in \mathcal{N}_i, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathcal{D}(,)$ is a distance measure in $\mathbb{R}^d$, the set $\mathcal{N}_i \subset [1:n]$ contains the indices of k nearest neighbors of $\boldsymbol{x}_i$ in $\mathcal{X}$, and $\varepsilon > 0$ is the bandwidth parameter. Let us define a diagonal matrix $\mathbf{D}$ with diagonal elements being $D_{ii} = \sum_{j=1}^n W_{ij}$, and compute the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [2]. Then, we can write the manifold regularizer as follows

$$\|f\|_{\mathcal{M}}^2 = \frac{1}{2} \sum_{i,j=1}^{n} \big(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\big)^2 W_{ij} = \boldsymbol{f}^\top \mathbf{L} \boldsymbol{f}, \tag{4}$$

in which the vector $\boldsymbol{f} = [f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_n)]^\top \in \mathbb{R}^n$ is the realization of the function f, prescribed by Eq. (1), on the training dataset $\mathcal{X}$. For concise notations, we define the coefficient vector $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^\top \in \mathbb{R}^n$, the kernel matrix $\mathbf{K} = [\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$, and the vectorial kernel mapping $\boldsymbol{k}(\boldsymbol{x}) = [\kappa(\boldsymbol{x}_1, \boldsymbol{x}), \cdots, \kappa(\boldsymbol{x}_n, \boldsymbol{x})]^\top \in \mathbb{R}^n$, so the target classification function f can be expressed as $f(\boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{k}(\boldsymbol{x})$ and $\boldsymbol{f} = \mathbf{K}\boldsymbol{\alpha}$.

The last term $-\sum_{i,y_i>0} f(\boldsymbol{x}_i)/n^+$ in the objective of problem (2) accounts for maximizing the margin averaged over the judged positive samples. Due to lacking in accurate labels, we consider an average margin, unlike SVMs and OC-SVMs which optimize a margin for individual examples. The strategy of average margin maximization over the positive samples is able to suppress the bias caused by the dubious outliers, pushing the majority of the true positive samples far away from the decision boundary $f(\boldsymbol{x}) = 0$. To avoid unbounded optimization, we further bound the range of $\{f(\boldsymbol{x}_i) = \boldsymbol{\alpha}^\top \boldsymbol{k}(\boldsymbol{x}_i)\}_{i=1}^n$ by fixing $\|\boldsymbol{\alpha}\| = 1$. Thus, $\sup\{f(\boldsymbol{x}_i) | 1 \le i \le n\} = \max_{1\le i\le n} \|\boldsymbol{k}(\boldsymbol{x}_i)\|$.

By incorporating Eq. (4) and ignoring the constant term $\|\boldsymbol{y}\|^2$, we rewrite problem (2) as

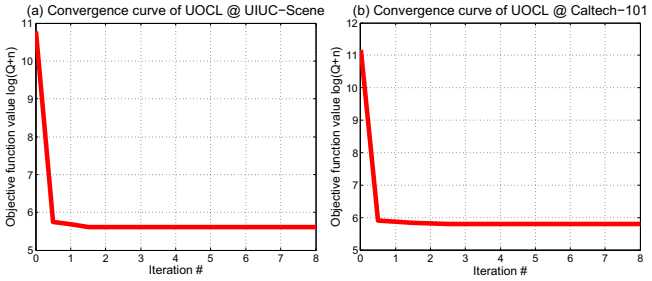

Figure 2. Convergence test of UOCL. At the t-th iteration, the objective function value is plotted as $\ln\big(Q(\boldsymbol{\alpha}_t, \tilde{\boldsymbol{y}}_t) + n\big)$. (a) On "MITcoast" category (360 samples) of **UIUC-Scene** with 30% outliers; (b) on "Faces" category (435 samples) of **Caltech-101** with 30% outliers.

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \tilde{\boldsymbol{y}}} \quad & Q(\boldsymbol{\alpha}, \tilde{\boldsymbol{y}}) := \boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{I} + \gamma_1 \mathbf{L})\mathbf{K}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \mathbf{K}\tilde{\boldsymbol{y}} \\ \text{s.t.} \quad & \|\boldsymbol{\alpha}\| = 1,\ \tilde{\boldsymbol{y}} \in \left\{ c^+ + \frac{\gamma_2}{\|\tilde{\boldsymbol{y}}\|_+},\ c^- \right\}^{n\times 1}, \\ & 0 < \|\tilde{\boldsymbol{y}}\|_+ < n, \end{aligned} \tag{5}$$

in which $\|\boldsymbol{a}\|_+$ stands for the number of positive elements in vector $\boldsymbol{a}$, and the new label assignment vector $\tilde{\boldsymbol{y}}$ takes the same signs as $\boldsymbol{y}$. While the objective function Q of problem (5) is convex, the feasible solution set is not a convex set, making problem (5) a combinatorial optimization problem.

To highlight the uniqueness of the proposed UOCL model, we point out that UOCL formulated in Eq. (5) can work under a self-guided mechanism, leading to a large-margin and neighborhood-smooth one-class classifier f along with a soft label assignment $\tilde{\boldsymbol{y}}$ (equivalent to $\boldsymbol{y}$) that directly indicates inliers and outliers. Different from previous outlier removal and one-class learning methods, our UOCL model does not overly emphasize positive samples nor outliers. Instead, it treats inliers and outliers fairly and makes them compete against each other through optimizing the label assignment $\tilde{\boldsymbol{y}}$ with the soft labels (c^+, c^-).

3.2. Algorithm

Problem (5) that fulfills the UOCL model is not trivial to solve because it is a mixed program involving a continuous variable $\boldsymbol{\alpha}$ and a discrete variable $\tilde{\boldsymbol{y}}$. Here we devise an alternating optimization algorithm which bears a theoretical foundation for convergence and obtains a good solution.

First, we consider the $\boldsymbol{\alpha}$-subproblem of problem (5) with $\tilde{\boldsymbol{y}}$ fixed:

$$\min_{\|\boldsymbol{\alpha}\|=1} \quad \boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{I} + \gamma_1 \mathbf{L})\mathbf{K}\boldsymbol{\alpha} - 2(\mathbf{K}\tilde{\boldsymbol{y}})^\top \boldsymbol{\alpha}, \tag{6}$$

which falls into a constrained eigenvalue problem that has been well studied in [9]. Subject to a fixed $\tilde{\boldsymbol{y}}$, the global minimizer $\boldsymbol{\alpha}$ to subproblem (6) is solved as

$$\boldsymbol{\alpha}^*(\tilde{\boldsymbol{y}}) = \big(\mathbf{K}(\mathbf{I} + \gamma_1 \mathbf{L})\mathbf{K} - \lambda^* \mathbf{I}\big)^{-1} \mathbf{K}\tilde{\boldsymbol{y}}, \tag{7}$$

in which λ^* is the smallest real-valued eigenvalue of the matrix $\begin{bmatrix} \mathbf{K}(\mathbf{I} + \gamma_1 \mathbf{L})\mathbf{K} & -\mathbf{I} \\ -(\mathbf{K}\tilde{\boldsymbol{y}})(\mathbf{K}\tilde{\boldsymbol{y}})^\top & \mathbf{K}(\mathbf{I} + \gamma_1 \mathbf{L})\mathbf{K} \end{bmatrix}$.

Second, we deal with the $\tilde{\boldsymbol{y}}$-subproblem of problem (5) with $\boldsymbol{\alpha}$ fixed, that is

$$\begin{aligned} \max_{\tilde{\boldsymbol{y}}} \quad & (\mathbf{K}\boldsymbol{\alpha})^\top \tilde{\boldsymbol{y}} \\ \text{s.t.} \quad & \tilde{\boldsymbol{y}} \in \left\{ c^+ + \frac{\gamma_2}{\|\tilde{\boldsymbol{y}}\|_+},\ c^- \right\}^{n\times 1}, \\ & 0 < \|\tilde{\boldsymbol{y}}\|_+ < n. \end{aligned} \tag{8}$$

This discrete optimization problem seems daunting but can be exactly solved in $O(n \log n)$ time. The theorem below addresses a simpler case under a given integer $m = \|\tilde{\boldsymbol{y}}\|_+ \in [1, n-1]$.

Theorem 1. *Given an integer $m \in [1, n-1]$ and a vector $\boldsymbol{f} \in \mathbb{R}^n$, an optimal solution to the problem*

$$\begin{aligned} \max_{\tilde{\boldsymbol{y}}} \quad & \boldsymbol{f}^\top \tilde{\boldsymbol{y}} \\ \text{s.t.} \quad & \tilde{\boldsymbol{y}} \in \left\{ c^+ + \frac{\gamma_2}{m},\ c^- \right\}^{n\times 1}, \\ & \|\tilde{\boldsymbol{y}}\|_+ = m \end{aligned} \tag{9}$$

satisfies $\tilde{y}_i > 0$ if and only if f_i is among m largest elements of $\boldsymbol{f}$.

Proof. We prove this theorem by contradiction.

Suppose that such an optimal solution is $\tilde{\boldsymbol{y}}$. Accordingly, we define its positive support set $\mathcal{C} = \{i | \tilde{y}_i > 0, i \in [1:n]\}$ of $|\mathcal{C}| = m$, and the complement of $\mathcal{C}$ is $\overline{\mathcal{C}} = [1:n] \setminus \mathcal{C}$.

If the conclusion does not hold, then there exist $i \in \mathcal{C}$ and $j \in \overline{\mathcal{C}}$ such that $f_i < f_j$. Now we construct another feasible solution $\tilde{\boldsymbol{y}}'$ corresponding to a new positive support set $\mathcal{C}' = (\mathcal{C} \setminus \{i\}) \cup \{j\}$. Then we derive

$$\begin{aligned} (\tilde{\boldsymbol{y}}')^\top \boldsymbol{f} &= \Big(c^+ + \frac{\gamma_2}{m}\Big) \sum_{s\in\mathcal{C}'} f_s + c^- \sum_{s\in\overline{\mathcal{C}'}} f_s \\ &= \Big(c^+ + \frac{\gamma_2}{m}\Big) \Big(\sum_{s\in\mathcal{C}\setminus\{i\}} f_s + f_j \Big) + c^- \Big(\sum_{s\in\overline{\mathcal{C}}\setminus\{j\}} f_s + f_i \Big) \\ &> \Big(c^+ + \frac{\gamma_2}{m}\Big) \Big(\sum_{s\in\mathcal{C}\setminus\{i\}} f_s + f_i \Big) + c^- \Big(\sum_{s\in\overline{\mathcal{C}}\setminus\{j\}} f_s + f_j \Big) \\ &= \Big(c^+ + \frac{\gamma_2}{m}\Big) \sum_{s\in\mathcal{C}} f_s + c^- \sum_{s\in\overline{\mathcal{C}}} f_s = \tilde{\boldsymbol{y}}^\top \boldsymbol{f}, \end{aligned}$$

which indicates that $\tilde{\boldsymbol{y}}'$ results in a larger objective value than $\tilde{\boldsymbol{y}}$, so $\tilde{\boldsymbol{y}}$ is not optimal. By contradiction we conclude that the theorem holds. □

Theorem 1 uncovers that one optimal solution to problem (9) can be simply attained by sorting $\boldsymbol{f}$ in a descending order and then cutting off at the m-th sorted element before

Algorithm 1 UOCL

Input: The kernel and graph Laplacian matrices $\mathbf{K}, \mathbf{L} \in \mathbb{R}^{n\times n}$, model parameters $\gamma_1, \gamma_2 > 0$, and soft labels $c^+ > 0, c^- < 0$.

Initialize

$\boldsymbol{\alpha}_0 = \frac{\mathbf{1}}{\sqrt{n}}, m_0 = \arg\max_{m\in[1:n-1]} (\mathbf{K}\boldsymbol{\alpha}_0)^\top \boldsymbol{q}(\mathbf{K}\boldsymbol{\alpha}_0, m)$,

$\tilde{\boldsymbol{y}}_0 = \boldsymbol{q}(\mathbf{K}\boldsymbol{\alpha}_0, m_0)$, $\mathbf{T} = \mathbf{K}(\mathbf{I} + \gamma_1\mathbf{L})\mathbf{K}$, $t = 0$;

repeat

$\boldsymbol{b}_t := \mathbf{K}\tilde{\boldsymbol{y}}_t$, $\lambda_t :=$ smallest eigenvalue of $\begin{bmatrix} \mathbf{T} & -\mathbf{I} \\ -\boldsymbol{b}_t\boldsymbol{b}_t^\top & \mathbf{T} \end{bmatrix}$,

$\boldsymbol{\alpha}_{t+1} := \left(\mathbf{T} - \lambda_t\mathbf{I}\right)^{-1}\boldsymbol{b}_t$,

$m_{t+1} := \arg\max_{m\in[1:n-1]} (\mathbf{K}\boldsymbol{\alpha}_{t+1})^\top \boldsymbol{q}(\mathbf{K}\boldsymbol{\alpha}_{t+1}, m)$,

$\tilde{\boldsymbol{y}}_{t+1} := \boldsymbol{q}(\mathbf{K}\boldsymbol{\alpha}_{t+1}, m_{t+1})$, $t := t + 1$,

until convergence.

Output: The one-class classifier $f^*(\boldsymbol{x}) = \boldsymbol{\alpha}_t^\top \boldsymbol{k}(\boldsymbol{x})$ and the soft label assignment $\tilde{\boldsymbol{y}}^* = \tilde{\boldsymbol{y}}_t$ over the training dataset.

and including which $\tilde{y}_i > 0$ while after which $\tilde{y}_i < 0$. We write this optimal solution as $\boldsymbol{q}(\boldsymbol{f}, m)$. Note that if some identical elements appear in the vector $\boldsymbol{f}$, there may exist multiple optimal solutions $\tilde{\boldsymbol{y}}^*$ that yield the same objective value. Subsequently, we go back to the original $\tilde{\boldsymbol{y}}$-subproblem (8) whose optimal solution is obtained by

$$\tilde{\boldsymbol{y}}^*(\boldsymbol{\alpha}) = \boldsymbol{q}\big(\mathbf{K}\boldsymbol{\alpha}, m^*(\boldsymbol{\alpha})\big), \tag{10}$$

in which

$$m^*(\boldsymbol{\alpha}) = \arg\max_{m\in[1:n-1]} (\mathbf{K}\boldsymbol{\alpha})^\top \boldsymbol{q}(\mathbf{K}\boldsymbol{\alpha}, m). \tag{11}$$

It is worth mentioning that if a tie arises in determining the optimal $m^*(\boldsymbol{\alpha})$, we always choose the largest integer as m^* in order to include as many as possible inliers, *i.e.*, the potential positive samples.

So far, we have exactly solved two subproblems (6)(8) stemming from the raw problem (5) which is too difficult to optimize directly. As such, we can devise an alternating optimization algorithm to find a good solution to problem (5). By taking advantage of Eqs. (7)(10)(11), we describe this optimization algorithm, still dubbed UOCL, in Algorithm 1 and prove its convergence by Theorem 2. After reaching a convergent label assignment $\tilde{\boldsymbol{y}}^*$, the positive samples or the outliers are simply determined by checking whether $\tilde{y}_i^* > 0$ or not.

Theorem 2. *The optimization algorithm alternating between $\boldsymbol{\alpha}$ and $\tilde{\boldsymbol{y}}$ converges.*

Proof. Because of the alternating optimization strategy, we accomplish $\boldsymbol{\alpha}_{t+1} = \arg\min_{\|\boldsymbol{\alpha}\|=1} Q(\boldsymbol{\alpha}, \tilde{\boldsymbol{y}}_t)$ and $\tilde{\boldsymbol{y}}_{t+1} = \arg\min_{\tilde{\boldsymbol{y}}} Q(\boldsymbol{\alpha}_{t+1}, \tilde{\boldsymbol{y}})$ for any iteration t. Thus, we can derive

$$Q(\boldsymbol{\alpha}_t, \tilde{\boldsymbol{y}}_t) \geq Q(\boldsymbol{\alpha}_{t+1}, \tilde{\boldsymbol{y}}_t) \geq Q(\boldsymbol{\alpha}_{t+1}, \tilde{\boldsymbol{y}}_{t+1}),\ \forall t \in \mathbb{Z}.$$

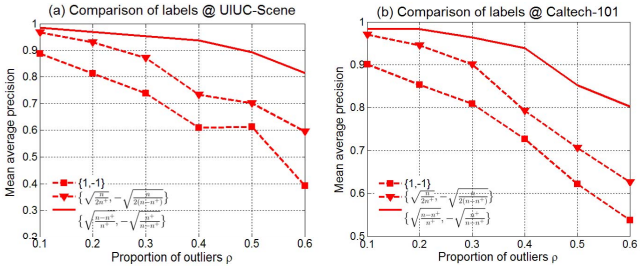

Figure 3. Comparison of labels (hard and soft) used in UOCL on the **UIUC-Scene** and **Caltech-101** datasets.

As Q is bounded from below, the nonincreasing sequence $\{Q(\boldsymbol{\alpha}_t, \tilde{\boldsymbol{y}}_t)\}_t$ must converge to $Q^* = \lim_{t\to\infty} Q(\boldsymbol{\alpha}_t, \tilde{\boldsymbol{y}}_t)$. □

Remarks: i) The UOCL algorithm essentially employs the kernel density estimate function as the warm start $\boldsymbol{f}_0 = \mathbf{K}\mathbf{1}/\sqrt{n}$ to launch its alternating procedure. This implies that the outliers are initially sought as those low-density samples and later gradually separated from the coherent high-density regions which give rise to the confident positive samples. ii) The target one-class classifier f is trained in an iterative and self-guided manner: at each iteration t, it absorbs a noisy label assignment $\tilde{\boldsymbol{y}}_t$ but yields a smooth yet discriminative output $\boldsymbol{f}_{t+1} = \mathbf{K}\boldsymbol{\alpha}_{t+1}$ by enforcing graph Laplacian regularization and average margin maximization; a refined labeling $\tilde{\boldsymbol{y}}_{t+1}$ is achieved via quantizing $\boldsymbol{f}_{t+1}$ and then serves re-training of f at the next iteration. When convergence is reached, the locations of the positive entries in both $\boldsymbol{f}$ and $\tilde{\boldsymbol{y}}$ will concentrate on a coherent high-density subset of the input dataset $\mathcal{X}$. iii) The time complexity of the UOCL algorithm is bounded by $O(n^3)$. In practice, UOCL usually converges rapidly within a few iterations, which is illustrated by Fig. 2 where only three iterations are needed for UOCL to converge.

3.3. Discussions

While UOCL can work with any soft labels (c^+, c^-) such that $c^+ > 0$, $c^- < 0$, and $\|\boldsymbol{y}\|^2$ is constant, we wish to investigate the impact of the choice of (c^+, c^-) on the performance of UOCL. In particular, we evaluate three kinds of labels $(1, -1)$, $(\sqrt{\frac{n}{2n^+}}, -\sqrt{\frac{n}{2(n-n^+)}})$, and $(\sqrt{\frac{n-n^+}{n^+}}, -\sqrt{\frac{n^+}{n-n^+}})$ all satisfying $\|\boldsymbol{y}\|^2 = n$. Fig. 3 plots mean average precision of the one-class classifier f^* learned by UOCL using different label settings on the **UIUC-Scene** and **Caltech-101** datasets. The results in Fig. 3 reveal that the soft labels $(\sqrt{\frac{n}{2n^+}}, -\sqrt{\frac{n}{2(n-n^+)}})$ and $(\sqrt{\frac{n-n^+}{n^+}}, -\sqrt{\frac{n^+}{n-n^+}})$ are better than the hard labels $(1, -1)$, and that $(\sqrt{\frac{n-n^+}{n^+}}, -\sqrt{\frac{n^+}{n-n^+}})$ surpasses the other two significantly, especially at increasing outlier propor-

tions. The reason is that $(\sqrt{\frac{n-n^+}{n^+}}, -\sqrt{\frac{n^+}{n-n^+}})$ incorporates the number n^+ of the positive samples to accomplish an adaptive balance of the labeling, *i.e.*, $\sum_{i=1}^{n} y_i = 0$, so that the positive samples and outliers are treated in a more balanced way. As a result, the adaptively balanced soft labels $(\sqrt{\frac{n-n^+}{n^+}}, -\sqrt{\frac{n^+}{n-n^+}})$ can enable our UOCL model to discover and suppress a high fraction of outliers.

Note that our learning objective in the raw problem (2) does not leverage the RKHS regularization term $\|f\|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}$ that appears in most (semi-)supervised kernel machines such as SVMs, OC-SVMs, and LapSVMs [2]. We find that including $\|f\|_{\mathcal{H}}^2$ into the objective of Eq. (2) almost does not affect the performance of UOCL. Moreover, we argue that the main purpose of UOCL is not to gain the generalization capacity on test data, but to tackle the corrupted training data and clean up the outliers.

In running UOCL: the assumption "outliers from low-density samples" has been used as the initialization, where the low-density samples found by the kernel density estimator $\boldsymbol{f}_0 = \mathbf{K}\mathbf{1}/\sqrt{n}$ are initially judged to be the outliers; the number n^+ of the asserted inliers will change during the optimization until the label assignment $\tilde{\boldsymbol{y}}$ converges; the inlier configuration of multiple modes can be handled as long as these modes have comparable high densities.

4. Experiments

We evaluate the proposed UOCL approach in terms of two tasks, outlier image removal and image re-ranking.

We use three public image datasets: **UIUC-Scene**[1], **Caltech-101**[2], and the **INRIA** web image dataset [12], and our gathered **Google-30** dataset which consists of 30 categories (*e.g.*, 'accordion', 'butterfly', 'clownfish', 'eagle', 'elephant', 'ewer', 'gecko', 'hat', 'horse', 'panda', *etc.*) of crawled web images using the Google image search engine. On **UIUC-Scene**, we use all 15 object categories, and for a single category simulate outlier images with a proportion $0.1 \leq \rho \leq 0.6$ as the images randomly sampled from the other categories. On **Caltech-101**, we choose 11 object categories each of which contains at least 100 images, and also simulate outlier images with a proportion $0.1 \leq \rho \leq 0.6$ as uniformly randomly sampled images from the other categories besides a respective category. On **INRIA**, we select 200 text queries each of which incurs an outlier proportion $0.136 \leq \rho \leq 0.6$ and contains $14 \sim 290$ images. On **Google-30**, 15 categories incur outlier proportions $0.0197 \leq \rho \leq 0.5599$ and contain $326 \sim 596$ images. In **INRIA** and **Google-30**, outliers are realistic, which are those irrelevant images with respect to the text queries. In all datasets, groundtruth labels for inliers and outliers are available. In **INRIA** each image is represented by an ℓ_2 normalized $5 * 1024$-dimensional sparse-coding feature vector [27], while in the other datasets every image is represented by an ℓ_2 normalized $21 * 1024$-dimensional sparse-coding feature vector.

We compare UOCL with a variety of competing methods, including five reconstruction-based outlier detection methods PCA, High-dimensional Robust PCA (HR-PCA)[3] [29], Kernel PCA, Kernel HR-PCA (KHR-PCA) [29], and Sparse Modeling Representative Selection (SMRS) [7], two density-based methods Kernel Density Estimator [23] and Robust Kernel Density Estimator (RKDE) [11], along with the traditional one-class learning method One-Class SVM (OC-SVM) [24]. Note that OC-SVM is an essentially supervised method, but in this paper it is made to work under the unsupervised setting.

Now we characterize outlierness measures for these methods. The measure used by the subspace methods is the squared reconstruction residue and the outliers are thus the samples incurring high residues. The measure of SMRS is the row-sparsity index (rsi) [7] calculated upon a sparse reconstruction coefficient, which only applies to the representatives. The representatives with high rsi are decided to be outliers. For KDE/RKDE, the outlierness measure is the estimated probability density function, and the samples with low densities are outliers. We adopt the same Gaussian kernel for KDE/RKDE, and its bandwidth is chosen via least square cross validation. We follow [11] to set the other parameters of RKDE. For OC-SVM/UOCL, the output of the learned one-class classifier f directly indicates the outliers, *i.e.*, $\boldsymbol{x}$ such that $f(\boldsymbol{x}) < 0$. We feed the same Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/2\sigma^2)$ to OC-SVM/UOCL, and estimate $\sigma^2 = \sum_{i,j=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/n^2$. The model rejection rate parameter ν associated with OC-SVM is chosen via a max-margin principle; in a similar way we choose the model parameters γ_1, γ_2 in UOCL such that the maximum average margin of the judged inliers is obtained. On all datasets, UOCL uses the soft labels $(\sqrt{\frac{n-n^+}{n^+}}, -\sqrt{\frac{n^+}{n-n^+}})$. To construct kNN graphs, we define $\mathcal{D}(,)$ as the squared Euclidean distance and fix $k = 6$.

To acquire the cut-off thresholds for the seven outlier detection methods, we perform binary clustering over the outlierness measure values via deterministic seeding (two initial seeds are the largest and smallest values, respectively). For each method, the cut-off threshold is set to the mean of two cluster centers. In summary, all compared methods can return a subset that contains the "asserted" positive (normal) data examples. Except SMRS which cannot obtain outlierness measure values for full samples, the others can re-rank full samples according to the outlierness measure values or classifier outputs.

[1] http://www-cvr.ai.uiuc.edu/ponce_grp/data/

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[3] Because the tried datasets are all in high dimensions, we run this latest version of Robust PCA (also Robust KPCA).

Table 1. **UIUC-Scene** & **Caltech-101** datasets: mean precision (mPre), mean recall (mRec), mean F_1 score (mF_1), mean average precision (mAP), and mean running time over the image categories of seven outlier detection methods and two one-class learning methods. All time is recorded in second. For each column, the best result is shown in boldface.

Method	**UIUC-Scene** (60% outliers)					**Caltech-101** (60% outliers)				
	mPre	**mRec**	**mF_1**	**mAP**	**Time**	**mPre**	**mRec**	**mF_1**	**mAP**	**Time**
Initial	0.4011	1.0000	0.5726	–	–	0.4019	1.0000	0.5734	–	–
PCA	0.5352	0.8626	0.6587	0.6957	0.63	0.5058	0.8465	0.6321	0.6483	0.22
HR-PCA [29]	0.5336	0.8623	0.6577	0.6948	0.70	0.5221	0.8710	0.6520	0.6591	0.60
KPCA	0.5619	0.8294	0.6580	0.6122	0.54	0.5428	0.8154	0.6504	0.6436	0.25
KHR-PCA [29]	0.4684	**0.8999**	0.6147	0.5910	0.65	0.5073	**0.8825**	0.6428	0.6346	0.72
SMRS [7]	0.4536	0.8612	0.5933	–	1.91	0.5394	0.8690	0.6531	–	4.32
KDE [23]	0.5086	0.8851	0.6448	0.6892	**0.46**	0.4949	0.8579	0.6266	0.6470	**0.18**
RKDE [11]	0.5475	0.8943	0.6760	0.7306	0.47	0.5003	0.8736	0.6346	0.6570	0.19
OC-SVM [24]	0.5816	0.6209	0.5934	0.6350	2.23	0.5290	0.7155	0.6012	0.5981	4.74
UOCL (our approach)	**0.7027**	0.8822	**0.7754**	**0.8157**	1.31	**0.6795**	0.8587	**0.7483**	**0.8027**	2.28

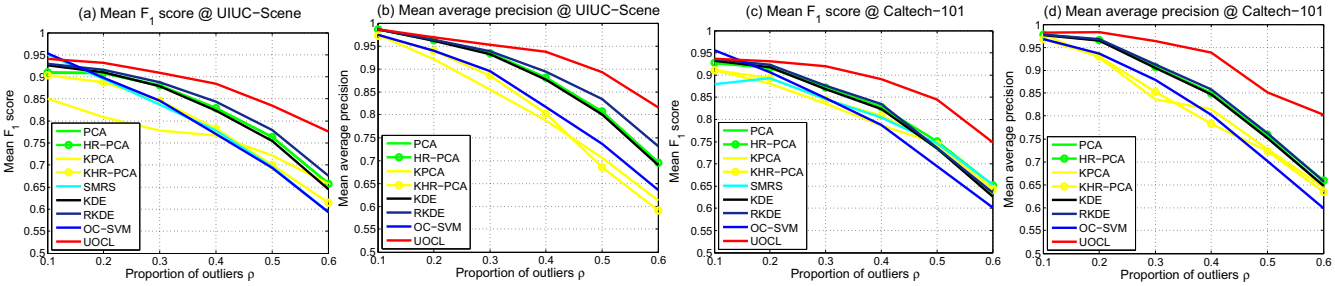

Figure 4. The results on the **UIUC-Scene** and **Caltech-101** datasets.

Since groundtruth labels are available on all datasets, we can compute precision, recall, and F_1 score for the outlier removal results achieved by all methods, and also compute average precision for the re-ranking results achieved by all methods except SMRS. The running time is also reported. Such results are shown in Tabs. 1 and 2 and Figs. 4 and 5.

Through these results, we can see that UOCL achieves the highest mean precision and mean F_1 score for most cases in terms of the outlier removal performance. It also consistently accomplishes the highest mean average precision and precision curve in terms of the re-ranking performance. The accuracy gains of UOCL over the other methods are more prominent when the proportion ρ of outliers increases. For example, on **UIUC-Scene** the gains in mPre, mF_1, and mAP of UOCL over the best competitor are respectively 21%, 15%, and 12% at $\rho = 0.6$; on **Caltech-101** the gains in mPre, mF_1, and mAP of UOCL over the best competitor are respectively 25%, 15%, and 22% at $\rho = 0.6$. While the accuracy gains on **INRIA** and **Google-30** are less sharp, UOCL still accomplishes the highest mean precision, mean F_1 score, mean average precision, and precision curve among all compared methods. The reason for decreased accuracy improvements may be that for some complicated query concepts the inlier images are less coherent and probably distributed in some isolated and sparse clusters, for which UOCL is likely to only capture the densest clusters and lose the inliers in the sparse clusters. All these experimental results disclose: 1) OC-SVM falls short at the high outlier level; 2) HR-PCA shows the robustness to some extent as it can remove some outliers in discovering the principal subspace; 3) UOCL exhibits the strongest robustness to the outlier images, producing the most coherent image subset from a contaminated image set with a high fraction of artificial or real-world outliers.

5. Conclusions

The proposed unsupervised one-class learning (UOCL) approach is highly robust to contamination of input training data and capable of suppressing outliers with a high proportion up to 60%. Extensive image outlier removal and image re-ranking results on four image datasets demonstrate that UOCL considerably outperforms the state-of-the-arts. The success of UOCL stems from three primary factors: 1) the self-guided learning mechanism jointly optimizes a large margin one-class classifier and a label assignment for inliers and outliers; 2) the adaptively balanced soft labels are exploited to handle the high outlier level; 3) the alternating optimization algorithm achieves rapid convergence.

Acknowledgements Dr. Wei Liu is partly supported by Josef Raviv Memorial Postdoctoral Fellowship. Dr. Gang Hua is partly supported by US National Science Foundation Grant IIS 1350763, China National Natural Science Foundation Grant 61228303, and GH's start-up funds from Stevens Institute of Technology.

References

[1] K. Balasubramanian, P. Donmez, and G. Lebanon. Unsupervised supervised learning ii: Margin-based classification without labels. *JMLR*, 12:3119–3145, 2011.

Table 2. The results on the **INRIA** and **Google-30** datasets.

Method	INRIA					Google-30				
	mPre	**mRec**	**mF_1**	**mAP**	**Time**	**mPre**	**mRec**	**mF_1**	**mAP**	**Time**
Initial	0.5734	1.0000	0.7221	0.6779	–	0.7727	1.0000	0.8645	0.8476	–
PCA	0.6714	0.7183	0.6749	0.7214	**0.02**	0.8568	0.8286	0.8299	0.8940	1.21
HR-PCA	0.7033	0.7273	0.6845	0.7264	0.23	0.8637	0.8238	0.8321	0.8956	2.83
KPCA	0.6584	0.5979	0.5806	0.6988	0.03	0.8162	0.7211	0.7518	0.8488	1.34
KHR-PCA	0.6720	0.6063	0.6015	0.6883	0.35	0.8048	0.8850	0.8308	0.8648	2.12
SMRS	0.6247	0.8453	0.7055	–	0.49	0.7812	**0.9932**	0.8672	–	9.79
KDE	0.6478	0.7837	0.6916	0.7186	**0.02**	0.8380	0.8670	0.8391	0.8912	**1.12**
RKDE	0.6533	0.7807	0.6931	0.7202	0.03	0.8425	0.8680	0.8405	0.8953	1.15
OC-SVM	0.5988	**0.9505**	0.7275	0.6912	1.03	0.8144	0.9220	0.8538	0.8838	11.27
UOCL (our approach)	**0.7371**	0.8489	**0.7671**	**0.7930**	0.32	**0.9123**	0.8795	**0.8804**	**0.9119**	3.65

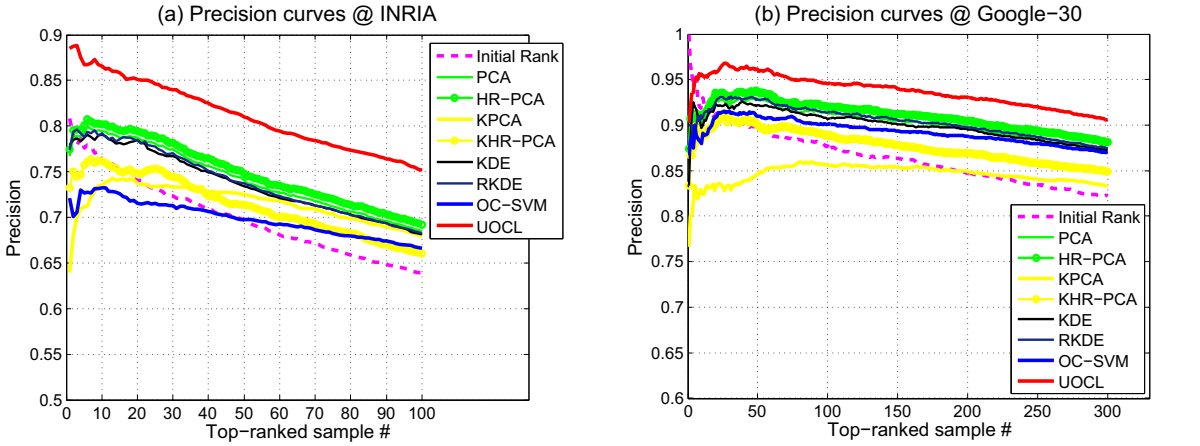

Figure 5. The results on the **INRIA** and **Google-30** datasets.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[3] J. Bi and T. Zhang. Support vector classification with input data uncertainty. In *NIPS 17*, 2004.

[4] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):Article 11, 2011.

[5] Y. Chen, X. S. Zhou, and T. S. Huang. One-class svm for learning in image retrieval. In *Proc. ICIP*, 2001.

[6] K. Crammer and G. Chechik. A needle in a haystack: Local one-class optimization. In *Proc. ICML*, 2004.

[7] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *Proc. CVPR*, 2012.

[8] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from internet image searches. *Proceedings of the IEEE*, 98(8):1453–1466, 2010.

[9] W. Gander, G. H. Golub, and U. von Matt. A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114/115:815–839, 1989.

[10] G. Gupta and J. Ghosh. Robust one-class clustering using hybrid global and local search. In *Proc. ICML*, 2005.

[11] J. Kim and C. D. Scott. Robust kernel density estimation. *JMLR*, 13:2529–2565, 2012.

[12] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web image search results using query-relative classifiers. In *Proc. CVPR*, 2010.

[13] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proc. KDD*, 2008.

[14] F. D. la Torre and M. J. Black. A framework for robust subspace learning. *IJCV*, 54(1/2/3):117–142, 2003.

[15] W. S. Lee and B. Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proc. ICML*, 2003.

[16] B. Liu, Y. Xiao, L. Cao, and P. S. Yu. One-class-based uncertain data stream learning. In *Proc. SIAM International Conference on Data Mining*, 2011.

[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proc. International Conference on Data Mining*, 2008.

[18] W. Liu, Y.-G. Jiang, J. Luo, and S.-F. Chang. Noise resistant graph ranking for improved web image search. In *Proc. CVPR*, 2011.

[19] W. Liu, J. Wang, and S.-F. Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

[20] L. M. Manevitz and M. Yousef. One-class svms for document classification. *JMLR*, 2:139–154, 2001.

[21] J. Muñoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camps-Valls. Semisupervised one-class support vector machines for classification of remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(8):3188–3197, 2010.

[22] M. H. Nguyen and F. D. la Torre. Robust kernel principal component analysis. In *NIPS 21*, 2008.

[23] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[24] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[25] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[26] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. CVPR*, 2010.

[28] X. Wang, S. Qiu, K. Liu, and X. Tang. Web image re-ranking using query-specific semantic signatures. *TPAMI*, 2014.

[29] H. Xu, C. Caramanis, and S. Mannor. Outlier-robust pca: The high-dimensional case. *IEEE Transactions on Information Theory*, 59(1):546–572, 2013.

[30] L. Xu, K. Crammer, and D. Schuurmans. Robust support vector machine training via convex outlier ablation. In *Proc. AAAI*, 2006.

[31] H. Yu, J. Han, and K. C. C. Chang. Pebl: web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004.

[32] A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, 2012.