USING SUPPORT VECTOR MACHINES FOR ANOMALOUS CHANGE DETECTION

Ingo Steinwart, James Theiler, and Daniel Llamocca

Los Alamos National Laboratory
Los Alamos, NM 87545

1. INTRODUCTION

Given two images of the same scene, taken at different times and (inevitably) under different conditions, we consider the problem of finding anomalous changes in the scene [1]. There will be pervasive differences between these two images, due to the different conditions under which the images were taken, but our working assumption is that the character of these differences will be the same over the whole image. By contrast, the anomalous changes will be small and/or rare, and their character will be different from the pervasive differences. The recasting of this problem in terms of binary classification enables the use of more sophisticated machine learning tools than have traditionally been employed for the change detection problem.

In this paper, we investigate the use of support vector machines (SVMs) with radial basis kernels for finding anomalous changes. Compared to typical applications of SVMs, we are operating in a regime of very low false alarm rate. This means that even for relatively large training sets, the data are quite meager in the regime of operational interest. This drives us to use larger training sets, which in turn places more of a computational burden on the SVM.

We take three different approaches to address this problem. The first is a standard SVM which is trained at one threshold (where more reliable estimates of false alarm rates are possible) and then re-thresholded for the low false alarm rate regime. The second uses the same thresholding approach, but employs a so-called least squares SVM; here a quadratic (instead of a hinge-based) loss function is employed, and for this model, there are good theoretical arguments in favor of adjusting the threshold in a straightforward manner. The third approach is a weighted support vector machine, where the weights for the two type of errors (false alarms and missed detection) are automatically adjusted to achieve the desired low false alarm rate. Our experiments show that in some cases the first two types work well, while in some other cases they do not. This renders both approaches unreliable for automated anomalous change detection. By contrast, the third approach reliably produces good results, but at the cost of larger computational requirements caused by the need to estimate very small false alarm rates. To address these computational requirements, we employ a recently developed in-house solver for SVMs that is significantly faster than freely available standard solvers.

But these computational issues are secondary to the larger question: do kernelized solutions provide better performance, in terms of detection rates and false alarm rates, than more traditional methods for change detection that effectively assume Gaussian data distributions? To this end, we will compare ROC curves obtained from the SVM with those from chronochrome [2], covariance equalization [3], and hyperbolic anomalous change detection [4].

This work was supported by the Los Alamos Laboratory Directed Research and Development (LDRD) program.

2. ANOMALOUS CHANGE DETECTION

A seeming difficulty with anomalous change detection, and with anomaly detection generally, is that anomalies tend to defy precise definition. We say that they are *not* normal or that they are not *not* typical, but we have more trouble trying to say what they *are*. As is usually the case with detection problems, however, the main technical challenge lies not in characterizing the targets, but in characterizing the background – in this case, the non-anomalous pervasive differences.

Let $x \in \mathbb{R}^{d_x}$ be a pixel in the "x" image, and $y \in \mathbb{R}^{d_y}$ be the pixel at the corresponding location in the "y" image. We write $P(x, y)$ as a joint probablity distribtuion in $d_x + d_y$ dimensional space that describes how x and y are correlated over the two images. Here, $P(x, y)$ corresponds to our *model* for pervasive differences.[1]

As a one-class problem, $P(x, y)$ describes the one "ordinary class;" data outside (or on the tails of) this distribution are candidates for anomalies. One way to find these anomalies is to recast anomaly detection as binary classification [5]. In this recasting, one identifies an "anomaly class" defined by a generic low-information distribution – the usual choice is a uniform distribution with support that extends well beyond the range of the data. In this uniform case, contours of the likelihood ratio (*i.e.,* the Bayes optimal classifier) correspond to the contours of $P(x, y)$. A *non*uniform anomaly class, introduced previously [6], provides a model that is tailored for anomalous *change*.

Let $P_x(x) = \int P(x, y)\, dy$ and $P_y(y) = \int P(x, y)\, dx$ be the marginal distributions of $P(x, y)$. Here $P_x(x)$ corresponds to the distribution of pixels in the x-image, regardless of what is going on in the y-image. And $P_y(y)$ is the distribution of pixels in the y-image. Our model for anomalous change considers the x and y pixels as individually ordinary, but the relationship between them to be unusual. Specifically we write the product $P_x(x)P_y(y)$ as our model for anomalous changes. This allows a likelihood ratio to be defined:

$$\mathcal{A}(x, y) = \frac{P_x(x)P_y(y)}{P(x, y)}. \tag{1}$$

Here $\mathcal{A}(x, y)$ is our measure of anomalousness, and when it is above a given threshold, then we declare the change at a pixel pair (x, y) to anomalous.

From a machine learning point of view, however, we do not want to work with distributions explicitly. Instead, we want to work directly with samples that are drawn from this distribution (namely, our data). We can effectively draw data from $P_x(x)P_y(y)$ by *resampling* from our data. A pair (x, y) is obtained by choosing x randomly from the x-image, and *independently* choosing y from the y-image. In fact, this can very efficiently be done by just scrambling the pixels in one of the images. These pairs define our anomalous change class; the original data defines our pervasive difference class. And we have all we need to employ our favorite binary classification algorithm.

It is important to note, however, that this binary classification has to operate in the low false alarm rate regime. From the point of view of likelihood ratio, this is a simple matter of adjusting a threshold. But for binary classification, one does not obtain a likelihood ratio, and must employ other techniques. In the next section, we describe the use of *our* favorite binary classifier, the support vector machine, with unequal weighting on the two classes, for solving the anomalous change detection problem.

[1] We remark that this model treats the pixels as i.i.d. samples from a parent distribution, and in particular neglects spatial correlations in the imagery. For hyperspectral imagery, this is often reasonable because there is so much detailed spectral information at each pixel.

3. A WEIGHTED SUPPORT VECTOR MACHINE APPROACH

In this work we decided to use support vector machines (SVMs), which are one of the best classification methods currently available, to solve these weighted binary classification problems. Therefore, let us briefly recall SVMS (see e.g. [7, 8] for a thorough introduction). The core ingredient of SVMs is a so-called kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, that is, a symmetric positive semi-defiite function. In the following we will solely focus on the so-called Gaussian RBF kernels that, for a given $\sigma > 0$, are defined by $k_\sigma := (x, x') := \exp(-\sigma^2\|x - x'\|_2^2)$, where $\|\cdot\|_2^2$ denotes the Euclidean norm on $\mathbb{R}^d$. In it well-known that to each such kernel there exists a unique reproducing kernel Hilbert spaces (RKHS) H_σ, which consists of functions $f : \mathbb{R}^d \to \mathbb{R}$. Now, given a so-called regularization parameter $\lambda > 0$, a kernel parameter $\sigma > 0$, and two classification weights $w_- > 0$ and $w_+ > 0$ with $w_- + w_+ = 1$, the corresponding SVM solves the optimization problem

$$f_{\sigma,\lambda,w_-} = \arg\min_{f\in H}\left(\lambda\|f\|_{H_\sigma}^2 + \frac{w_-}{n_-}\sum_{y_i=-1} L(-1, f(x_i)) + \frac{w_+}{n_+}\sum_{y_i=1} L(1, f(x_i))\right), \tag{2}$$

where $L(y, t) := \max\{0, 1 - yt\}$ is the so-called hinge loss and n_- and n_+ denotes the number of negatively and positively labeled samples, respectively. It is well-known that (2) is a strictly convex ptimization problem that always have a unique solution $f_{\sigma,\lambda,w_-} \in H_\sigma$, see e.g. [8, Chapter 5.1]. Moreover, this solution is of the form

$$f_{\sigma,\lambda,w_-} = \sum_{i=1}^{n} y_i\alpha_i^* k_\sigma(x_i, \cdot\,), \tag{3}$$

where $(\alpha_i^*, \ldots, \alpha_n^*)$ is a solution of the dual optimization problem, see e.g. [8, Chapter 11.1]. Unfortunately, there is, in general, no way to use suitable a-priori knowledge to determine the free parameters λ, σ and w_-, and thus they are often determined by a hold-out set in the following way. First one fixes sets Λ, Σ, and W of candidate values for λ, σ and w_-, respectively and splits the training set into two subsets D_1 and D_2. Then for each triple $(\lambda, \sigma, w_-) \in \Lambda \times \Sigma \times W$, the SVM optimization problem for the dataset D_1 is solved and the false alarm rate and the detection rate of the resulting f_{σ,λ,w_-} is estimated using D_2. Finally, the triple (λ, σ, w_-) is picked for which the false alarm rate is below the given threshold and the detection rate is maximized. Analogously to the unweighted classification case, see e.g. [8, Chapter 8.3], one can show that under suitable conditions on Λ, Σ, and W this approach asymptotically yields optimal decision functions f_{σ,λ,w_-}. In addition, this approach closely resembles many approaches recommended in practice for unweighted binary classification problems. Consequently, we followed this approach modulo two minor modifications.

Conceptionally, the SVM approach is quite straightforward, but when implemented by standard SVM packages such as LIBSVM [9] it is computationally almost infeasible on a single desktop. Indeed, the fact that we need to determine three hyperparameters λ, σ and w_- means that we have to solve the dual problem several thousand times, which is too time-consuming when done by such packages. To address this issue we developed our own SVM solver [10], which on most datasets is at least 40 times (sic!) faster than LIBSVM. In addition, our SVM package also carefully caches the kernel matrices $(y_i y_j k_\sigma(x_i, x_j))_{i,j=1}^n$, which also decreases the entire training time significantly. Another computational bottleneck comes from the fact that we are interested in very small false alarm rates, which can only be estimated by large hold-out sets D_2. Now (3) shows that a brute-force approach for estimating the false alarm and detection rate for a single triple requires (λ, σ, w_-) requires $D_1 \cdot D_2$ kernel computations and the same amount of additional multiplications and additions. With sample sizes of a few thousand for D_1 and

100-200 thousand for D_2, this becomes computationally intractable when done for several thousand triples (λ, σ, w_-), even if the sparsity, see [8, Chapters 8.4 and 8.6] of the representation (3) is taken into account. To address this issue, we combined the sparseness of (3) with the following strategies: *a)* caching the kernel matrix and changing σ in the most outer loop of the hyper-parameter determination, *b)* updating (3) only for those α_i^* that have changed from the previous value of λ, which are changed in the most inner loop of the hyper-parameter determination, *c)* implementing the remaining summation on a graphical processing unit (GPU). By this means, a typical computation of the false alarm rate and the detection rate for a single triple (λ, σ, w_-) currently takes about 5ms if $D_1 = 1,000$ and $D_2 = 100,000$, while without these strategies the same computation exploiting only the sparseness takes about one minute on one of the currently fastest desktop processors (Intel Core i7 Extreme). Similarly, the test phase in which the final decision function is applied to the entire image requires computing (3) very often (depending on the image size up to several million times). Again, this is computationally too expensive when done on a CPU, and hence we implemented this step on an GPU, too. The discussion above shows that a rigorous SVM approach for the anomalous change detection problem requires a significant implementation effort.

4. REFERENCES

[1] Michael T. Eismann, Joseph Meola, Alan D. Stocker, Scott G. Beaven, and Alan P. Schaum, "Airborne hyperspectral detection of small changes," *Applied Optics*, vol. 47, pp. F27–F45, 2008.

[2] A. Schaum and A. Stocker, "Long-interval chronochrome target detection," *Proc. 1997 International Symposium on Spectral Sensing Research*, 1998.

[3] A. Schaum and A. Stocker, "Hyperspectral change detection and supervised matched filtering based on covariance equalization," *Proc. SPIE*, vol. 5425, pp. 77–90, 2004.

[4] J. Theiler, "Quantitative comparison of quadratic covariance-based anomalous change detectors," *Applied Optics*, vol. 47, pp. F12–F26, 2008.

[5] I. Steinwart, D. Hush, and C. Scovel, "A classification framework for anomaly detection," *J. Machine Learning Research*, vol. 6, pp. 211232, 2005.

[6] J. Theiler and S. Perkins, "Proposed framework for anomalous change detection," *ICML Workshop on Machine Learning Algorithms for Surveillance and Event Detection*, pp. 7–14, 2006.

[7] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[8] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer, New York, 2008.

[9] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: a library for support vector machines," `http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.ps.gz`, 2004 – 2009.

[10] I. Steinwart, D. Hush, and C. Scovel, "Training SVMs without offset," *J. Mach. Learn. Res.*, accepted with minor revision, 2009.