Density-Based Similarity Measures for Content Based Search

Reid Porter, Christy Ruggiero, and Don Hush
Los Alamos National Laboratory
Los Alamos, NM 87545
Email: dhush@lanl.gov

***Abstract*—We consider the *query by multiple example* problem where the goal is to identify database samples whose content is similar to a collection of query samples. To assess the similarity we use a *relative content density* which quantifies the relative concentration of the query distribution to the database distribution. If the database distribution is a mixture of the query distribution and a background distribution then it can be shown that database samples whose relative content density is greater than a particular threshold ρ are more likely to have been generated by the query distribution than the background distribution. We describe an algorithm for predicting samples with relative content density greater than ρ that is computationally efficient and possesses strong performance guarantees. We also show empirical results for applications in computer network monitoring and image segmentation.**

I. Introduction

Consider the *query by multiple example* problem where we are given a collection of *query samples* $Q = (q_1, ..., q_k)$, $q_i \in \mathcal{X}$ and a collection of *database* samples $X = (x_1, ..., x_n)$, $x_i \in \mathcal{X}$, and asked to identify members of X that are similar in content to members of Q. The data space $\mathcal{X}$ will typically be a space of images, signals, documents, or feature vector representations of one of these data types. We define a similarity measure to be a function on $\mathcal{X}$ that computes the similarity between a point $x \in \mathcal{X}$ and the query Q. Assuming that the samples in Q and X are generated according to probability distributions P_q and P_x respectively we consider using the following density functions as similarity measures;

- the *content density* p_q that quantifies the absolute concentration of P_q and is given by the Radon-Nikodym derivative
$$p_q := dP_q/d\mu$$
where is the μ is the Lebesgue measure, and
- the *relative content density* p that quantifies the relative concentration of P_q to P_x and is given by the Radon-Nikodym derivative
$$p := dP_q/dP_x$$
where we assume that P_q is absolutely continuous with respect to dP_x (so that p is well-defined).

Members of X that fall into the high density regions of p_q are also guaranteed to be close to a nontrivial fraction of the samples in Q, and so p_q is analogous to common distance-based similarity measures. With a suitable choice of t we can say that the set $\{x \in X : p_q(x) > t\}$ contains samples that are *likely to be generated by* P_q.

In contrast the *relative content density* p has a different interpretation. Suppose that some of the samples in X are generated by P_q and the rest by another process P_{other}. Then P_x is a mixture distribution

$$P_x = \beta P_q + (1-\beta) P_{other}, \quad \text{for some } 0 < \beta < 1.$$

and with the appropriate choice of ρ the set $\{x \in X : p(x) > \rho\}$ contains samples that are *more likely to be generated by* P_q *than* P_{other}. In many applications the *relative content density* p will be preferred over the *content density* p_q. However retrieval algorithms for the *relative content density* may be more complicated since this density relies on both P_q and P_x, instead of just P_q.

We now mention a related problem whose formulation turns out to be identical. In this problem the raw data is assumed to be structured in a way that leads to a standard notion of "local regions" within the data. Examples include image data whose local regions might correspond to small spatial windows, or time series data whose local regions might correspond to small temporal windows. For this problem we are given a query sample q and a database sample x and asked to identify local regions of x that are similar (in content) to the local regions of q. If we define $\acute{X} = (\acute{x}_1, \acute{x}_2, ...)$, $\acute{x}_i \in \mathcal{X}$ be the collection of local regions of x, and $\acute{Q} = (\acute{q}_1, \acute{q}_2, ...)$ $\acute{q}_i \in \mathcal{X}$ to be the collection of local regions of q, and assume data generating distributions $P_{\acute{q}}$ and $P_{\acute{x}}$, then we can define density-based similarities that quantify the absolute concentration of $P_{\acute{q}}$, and the relative concentration of $P_{\acute{q}}$ to $P_{\acute{x}}$ just as we did above. Applying a threshold to these density-based similarities performs a *segmentation* of x into local regions that are similar to the local regions of q.

The common task in the formulations above is to identify the members of X (or $\acute{X}$) whose density value exceeds a threshold. To accomplish this we solve a slightly more general problem called the *density level detection* (DLD) problem where the goal is to identify the subset of the data space $\mathcal{X}$ where the density exceeds a threshold. For convenience we will develop solution methods for the relative content density p with threshold ρ, but it should be clear that these same methods can be used for the other formulations as well.

To solve the DLD problem we design a real valued function f that approximates the set $\{p > \rho\}$ with the set $\{f > 0\}$.

The quality of the approximation is assessed by the criterion

$$s(f) := P_x(\{f > 0\} \vartriangle \{p > \rho\})$$

where $\vartriangle$ denotes the symmetric difference. This criterion corresponds to the average number of mistakes made by f, i.e. it represents the fraction of time that f predicts that a sample is in the high density region when it is not, plus the fraction of time it predicts that a sample is not in the high density region when it is. The design problem can be stated as follows.

Retrieval Function Design: *Given a $\rho > 0$, query samples $(q_1, ..., q_k) \sim P_q$, and input samples $(x_1, ..., x_n) \sim P_x$, design a function $\hat{f}$ such that $s(\hat{f})$ is small.*

Remark: A close relationship exists between the DLD problem above and the binary classification problem. In the binary classification problem we assume a data generating distribution $P_x = p_1 P_{x,1} + p_0 P_{x,0}$ and we seek a real valued function f that minimizes the binary classification error $e(f) = p_1 P_{x,1}(f < 0) + p_0 P_{x,0}(f \geq 0)$. In ([1], Section 5) it is shown that if $P_x = \beta P_q + (1-\beta) P_{other}$ then with $\rho = \frac{1}{2\beta}$ any f that minimizes s also minimizes the binary classification error $e(f)$ for the binary classification problem where $p_1 = \beta$, $p_0 = 1 - \beta$, $P_{x,1} = P_q$ and $P_{x,0} = P_{other}$. Thus, the algorithms developed in this paper are directly applicable to binary classification problems whose input data consists of labeled samples from one class and unlabeled samples from the mixture. These problems are often referred to as *learning from only positive and unlabeled data* (LPU) problems.

II. Solution Methods

Numerous solution methods might be considered for computing $\hat{f}$, including methods based on various forms of density estimation, but first we consider the issue of validating the performance of $\hat{f}$ independent of how it is computed. Since P_q and P_x (and therefore p) are generally unknown there is no hope of computing the performance $s(\hat{f})$ directly. Furthermore there appears to be no reliable way of estimating $s(\hat{f})$ from empirical data since we have no ground truth for the samples in Q and X (i.e. we do not know if their density values exceed ρ). However, a general method for validating the performance of $\hat{f}$ without computing or estimating $s(\hat{f})$ has been described in [1]. It is based on the risk function r defined by

$$r(f) := \frac{1}{1+\rho}\ P_Q(f \leq 0)\ +\ \frac{\rho}{1+\rho}\ P_X(f > 0). \tag{1}$$

Under mild assumptions on the density p it is possible to show that the risk r is *calibrated* to s [1], i.e. r and s obey a relationship that tightly couples their behavior. The exact relationship is described in [1], but here it suffices to say that this relationship implies that all functions that minimize r also minimize s. Furthermore, all functions that *approximately* minimize r also *approximately* minimize s. Thus, we can use r instead of s as a performance criterion. This is important because, unlike s, the risk r can be estimated from sample data. In particular we can use the samples in Q and X to provide empirical estimates of the probabilities in (1) and thus estimate r using

$$\hat{r}(f) = \frac{1}{1+\rho}\,\frac{1}{k}\sum_{i=1}^{k} I(f(q_i) \leq 0) + \frac{\rho}{1+\rho}\,\frac{1}{n}\sum_{j=1}^{n} I(f(x_j > 0)$$

where $I(\cdot)$ is the identity function that takes the value 1 when its argument is true and 0 otherwise. This estimate can be used to validate a solution $\hat{f}$, compare the performance of different solution methods, and select the so-called tuning parameters that accompany solution methods.

The calibrated risk r also allows us to consider non-density based solution methods, in particular methods that choose $\hat{f}$ to minimize r more directly (e.g. by minimizing $\hat{r}$ or a surrogate version). The method proposed in [1], and the one adopted here, is based on *support vector machines* (SVMs). Simply put, the SVM method chooses $\hat{f}$ from a *reproducing kernel Hilbert space* (RKHS) of functions $\mathcal{F}$ to minimize the surrogate criterion R given by

$$\begin{aligned} R(f) := \lambda\|f\|_{\mathcal{F}}^2 \ &+\ \left(\frac{1}{1+\rho}\right)\frac{1}{k}\sum_{i=1}^{k}\big[1 - f(q_i)\big]_+ \\ &+\ \left(\frac{\rho}{1+\rho}\right)\frac{1}{n}\sum_{i=1}^{n}\big[1 + f(x_i)\big]_+ \end{aligned} \tag{2}$$

where $[\cdot]_+$ is a clipping operation that gives $[a]_+ = a$ when $a > 0$ and 0 otherwise. The criterion R is obtained by replacing the nonconvex r with a convex and *calibrated* (see [2]) r_c given by

$$\begin{aligned} r_c(f) :=& \frac{1}{1+\rho}\ P_Q\left(\big[1 - f(q)\big]_+\right) + \\ & \frac{\rho}{1+\rho}\ P_X\left(\big[1 + f(x_i)\big]_+\right), \end{aligned}$$

then forming the empirical estimate $\hat{r}_c$ of r_c and adding a regularization term $\lambda\|f\|_{\mathcal{F}}^2$ to control the finite sample effects.

Solutions to (2) take the form [3]

$$f(\cdot) = \sum_{i=1}^{k}\alpha_{q,i}k(\cdot, q_i) + \sum_{i=1}^{n}\alpha_{x,i}k(\cdot, x_i)$$

where $k(\cdot,\cdot)$ is the kernel function for the RKHS $\mathcal{F}$. When the data space is $\mathcal{X} \subseteq \mathbb{R}^d$ a common choice for $k(\cdot,\cdot)$ is the Gaussian RBF kernel

$$k(x_1, x_2) = \exp(-\sigma\|x_1 - x_2\|^2).$$

With this choice the SVM method method has been shown to be universally consistent [1]. Furthermore, under mild assumptions on the distributions (similar to the assumptions required for the calibration of r to s) the SVM method has been shown to possess fast rates of convergence to the Bayes optimal solution [4].

Practical algorithms for (approximately) minimizing R are common, but few are guaranteed to run in polynomial time. We employ a polynomial time algorithm from [5], [6]. This algorithm is a so-called decomposition algorithm that searches for the optimal α using an iterative procedure that optimizes two coefficients at a time until a stopping condition that guarantees $R(\hat{f}) - \min_f R(f) < \epsilon$ is satisfied. Assuming a data space $\mathcal{X} \subseteq \mathbb{R}^d$ and applying the run time analysis in [5] to the SVM formulation here gives the following run time bound

$$O\left((n+k)^2\left[d + \frac{(n+k)}{\lambda\epsilon k^2} + \log\frac{\lambda k^2}{(n+k)}\right]\right).$$

A complete algorithm that builds on the above algorithm and automatically chooses the regularization parameter λ and kernel width σ is described and analyzed in [7]. A version of this algorithm is used in the experiments in this paper.

To validate our results we compute empirical estimates of r on a "hold out" data set. We also assess performance in terms of the following components of r.

- $P_Q(f \leq 0)$ is called the *Q-missed detection rate* and represents the fraction of X samples generated by P_Q but not retrieved, and
- $P_X(f > 0)$ is called the *retrieval rate* and represents the rate at which samples from X are predicted to be similar to Q.

To assess performance we plot the Q-detection rate (i.e. 1 - the Q-missed detection rate) versus the retrieval rate as ρ is varied over a range of values (similar to an ROC curve).

III. Experiments

We now describe experimental results for two applications; computer network monitoring, and image segmentation. The network monitoring application is an instance of the *query by multiple example* problem and the image segmentation application corresponds to a problem where we *identify local regions with similar content*. In both cases we compare solutions obtained with the SVM method in Section II against solutions obtained using a more conventional approach.

A. Network Monitoring

In the network monitoring problem our goal is to identify a particular type of activity in *encrypted* network flows[1]. In the experiments below we attempt to identify flows associated with the "CHAT" protocol, but our approach applies to other types of activity as well. When a flow is *unencrypted* it is relatively easy to determine the flow type by examining the flow packet contents. However this is useless for encrypted flows and so determining their flow type is a difficult problem. We solve this problem using a *query by multiple example* approach where:

- The database samples $X = (x_1, ..., x_n)$ correspond to encrypted network flows. In our experiments we have 100,027 encrypted flows from a busy computer network.
- The query samples $Q = (q_1, ..., q_k)$ correspond to flows of a known activity type. In our experiments we have 3450 CHAT flows from unencrypted traffic on the same busy network.

In our experiments the flow samples are represented by finite dimensional feature vectors derived from the *packet size* and *wait time* flow sequences. Example sequences are shown in the table below where the packet sizes are in bytes, weight times are in milliseconds, and the packet direction (i.e. host-to-client or client-to-host) is encoded by the sign of the number.

Packet Sizes	132, -122, 43, 28, -27, 23
Wait Times	-0.081, 0.003, -0.183, 0.002

We compare the SVM method with a conventional *signature matching* method which designates a sample $x \in X$ to be "similar" to Q if its Euclidean distance to one of the *signature samples* from Q is below a threshold t. To provide an appropriate tuning for different values of ρ, the threshold t is chosen to minimize the risk $\hat{r}$. The signature samples are chosen as a random subset of Q, and the remaining samples in Q are used to estimate the risk $\hat{r}$, choose the threshold t, and estimate the Q-missed detection and retrieval rates. Since the regions identified by placing hyperspheres of radius t around the signature samples provides a crude approximation to the high density regions of p_Q, this signature matching method is analogous to retrieval based on *content density* (as opposed to *relative content density*).

The performance results are shown in Figures 1 and 2. The lower risk values in Figure 1 tell us that the SVM method is doing a much better job at minimizing the mistake rate (since r is calibrated to s) for all values of ρ. The superiority of the SVM method is even more pronounced in Figure 2. For example this figure tells us that if we want to detect approximately 90% of the samples generated by P_q the retrieval rate for the SVM method is approximately 4 orders of magnitude smaller than the signature match method.

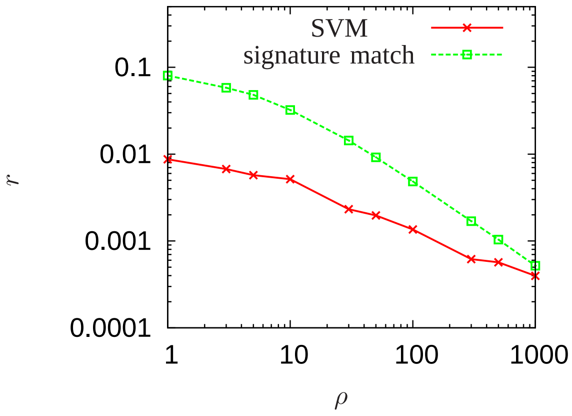

Fig. 1. Risk estimates for methods designed to extract CHAT flows from encrypted network traffic.

B. SAR Image Segmentation

Synthetic aperture radar (SAR) imaging has become an important surveillance tool for monitoring man–made targets such as buildings, manufacturing facilities, and military vehicles. The segmentation task is to identify regions of a SAR

[1] Network flows are sequences of packets with well-defined start and end points, and are the fundamental data unit processed by most network analysis tools.

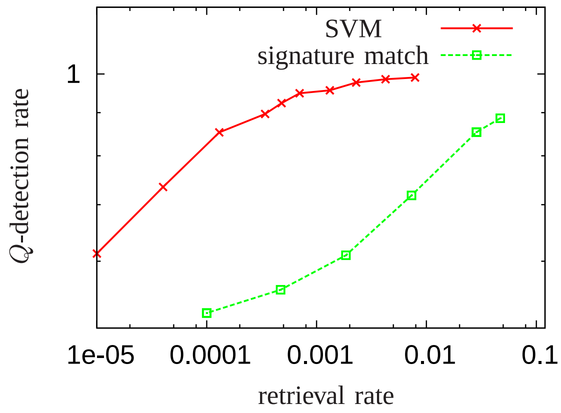

Fig. 2. Estimated performance curves for methods designed to extract CHAT flows from encrypted network traffic.

image that are likely to contain targets of interest. This is a challenging task because optimal segmentation is thought to require prior knowledge of both targets and clutter, but this knowledge is often not available because the deployed environments are not known ahead of time. We describe an approach that only requires target knowledge ahead of time, and (implicitly) gathers clutter information *in the field* at the time of deployment from the deployed SAR image that contains a mixture of target and clutter. In particular we identify local regions of the deployed SAR image that are similar in content to the local regions of a target rich query set constructed ahead of time. Using the results in [1] we can show that our approach provides optimal segmentation without the need for ground truth clutter information (see the remark at the end of Section I). Furthermore the solution is tailored to the statistics of each individual deployment.

The segmentation task is performed by a pixel classifier f that labels each pixel in the deployment image as either target or clutter, and then combines the target pixels to form the regions of interest. Our experiment uses one foot resolution, single–look, HH–polarized, X–band SAR magnitude data collected at a 15 degree depression angle as a part of DARPA's MSTAR program [8]. The target is a T-72 tank. To form the query set we selected 274 target images corresponding to a T-72 tank imaged at 274 different aspect angles over the range 0 to 360 degrees. Each target image was hand labeled as shown in Figure 3. Local "target regions" were represented by the (overlapping) 10–by–10 pixel windows surrounding the target pixels in each of these images. We extracted a random subset of 10,383 of these regions to form the query set $\acute{Q} = (\acute{q}_1, \ldots, \acute{q}_{10383}), q_i \in \mathbb{R}^{100}$. We show results against the deployment SAR image in Figure 7. This image contains 18 military vehicles (3 of them are T-72 tanks, 6 of them are other tanks) and 7 corner reflectors. A random subset of 41,670 local 10–by–10 pixel windows were extracted from this image and used in the SVM design of f.

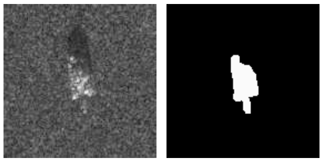

Fig. 3. A T-72 image (left) and a corresponding hand labeling of the target pixels (right).

The SVM solution is compared to a widely used SAR pixel classifier, the *cell averaging constant false alarm* (CA-CFAR) detector [9]. This detector computes the function

$$f(i,j) = \left(\frac{x(i,j) - \hat{\mu}(i,j)}{\hat{\sigma}(i,j)} \right) - \tau$$

at each pixel location (i,j) in the image where $x(i,j)$ is the pixel value at location (i,j), $\hat{\mu}(i,j)$ and $\hat{\sigma}(i,j)$ are the sample mean and standard deviation of pixel values from a stencil surrounding location (i,j) as illustrated in Figure 4, and τ is a threshold chosen to control the *false alarm rate* (i.e. the rate at which clutter pixels are labeled as target pixels)[2]. Pixel locations where $f > 0$ are labeled as target. Since this approach identifies pixel values in the complement of the high density regions of the clutter density it is analogous to retrieval based on *content density* (as opposed to *relative content density*). To provide a basis for comparison we chose the CFAR threshold τ to minimize the empirical risk R. In this way the threshold varies automatically with ρ.

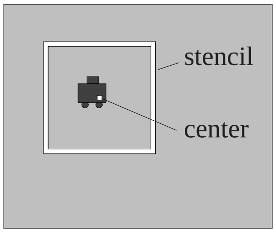

Fig. 4. Illustration of the stencil region used to estimate clutter statistics for the CA-CFAR detector.

The performance results are shown in Figures 5 and 6. Both figures support the conclusion that the SVM method is superior at minimizing the DLD mistake rate for all values of ρ (and therefore superior at identifying similar regions). These results are confirmed visually in Figure 7 where it is easy to see that the SVM method does a better job at identifying "on target" pixels and suppressing false alarms in the clutter regions.

Acknowledgment

The authors acknowledge support for this work from the LDRD program office at Los Alamos National Laboratory.

References

[1] I. Steinwart, D. Hush, and C. Scovel, "A classification framework for anomaly detection," *Journal of Machine Learning Research*, vol. 6, pp. 211–232, 2005.

[2] Pixels near the border of the image, where the stencil region is not well defined, must be treated differently. For the experiments in this paper they are simply ignored.

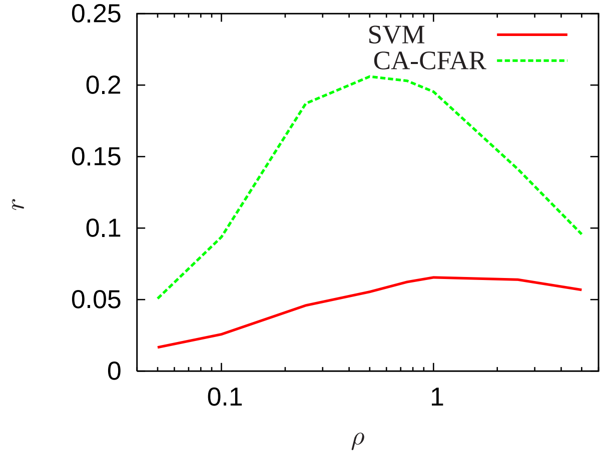

Fig. 5. Risk estimates for methods designed to find local regions of a deployment SAR image that are similar to local regions of a T-72 tank.

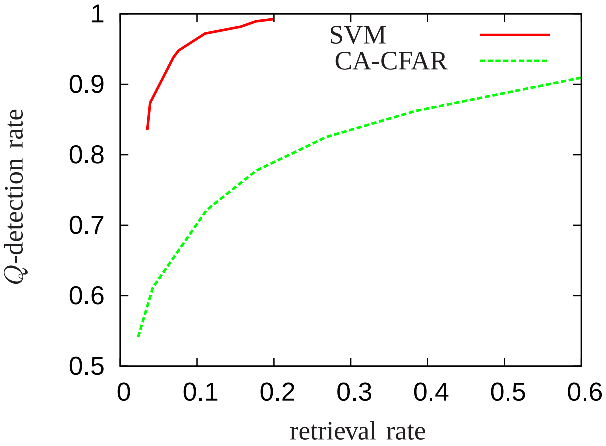

Fig. 6. Estimated performance curves for methods designed to find local regions of a deployment SAR image that are similar to local regions of a T-72 tank.

[2] P. Bartlett, M. Jordan, and J. McAuliffe, “Convexity, classification, and risk bounds,” *J. Amer. Statist. Assoc.*, vol. 101, pp. 138–156, 2006.

[3] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer, 2008.

[4] I. Steinwart, D. Hush, and C. Scovel, “Learning rates for density level detection,” *Analysis and Applications*, vol. 3, no. 4, pp. 356–371, 2005.

[5] D. Hush, P. Kelly, C. Scovel, and I. Steinwart, “QP algorithms with guaranteed accuracy and run time for support vector machines,” *Journal of Machine Learning Research*, vol. 7, pp. 733–769, 2006.

[6] I. Steinwart, D. Hush, and C. Scovel, “Training SVMs without offset,” *Journal of Machine Learning Research*, vol. *to appear*, 2009.

[7] ——, “An oracle inequality for clipped regularized risk minimizers,” in *Advances in Neural Information Processing Systems 19*. MIT Press, 2007, pp. 1321–2007.

[8] “Mstar public data collection,”
https://www.sdms.afrl.af.mil/request/data_request.htm, 1999.

[9] G. B. Goldstein, “False-alarm regulation in log-normal and Weibull clutter,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 9, no. 1, pp. 427–455, 1973.

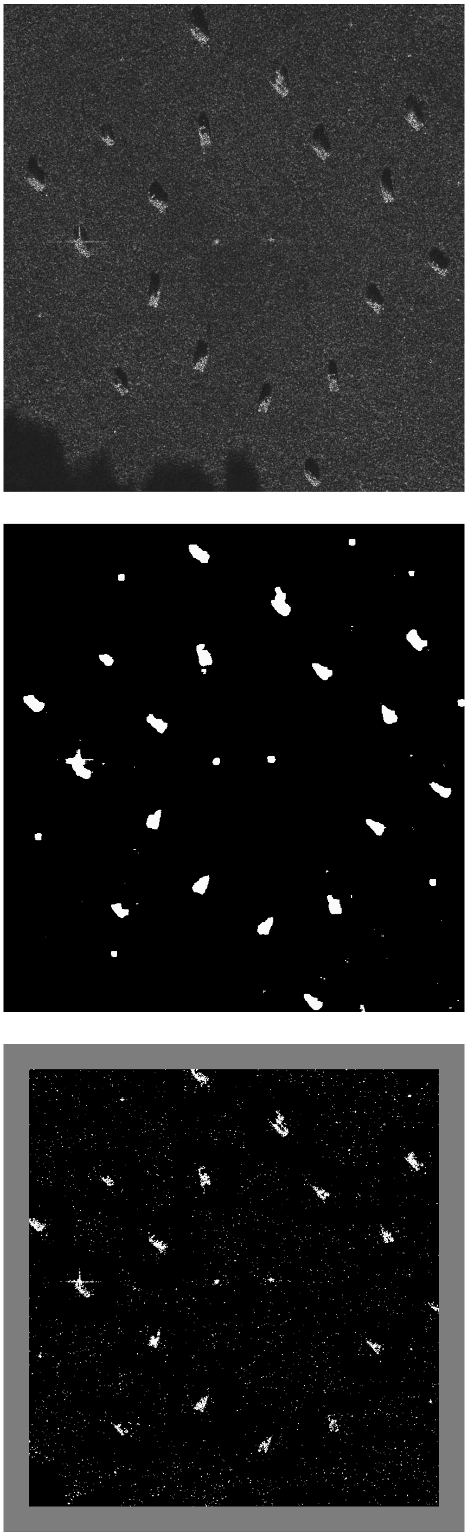

Fig. 7. The deployment SAR image (top), local regions identified by the SVM method (middle), and local regions identified by the CA-CFAR method (bottom).