Special Session on Hyperspectral Imaging

Maite Termenon[1]

[1]Computational Intelligence Group

2012 March 16

Article

IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 49, NO. 5, MAY 2011 1617

A Fast Cluster-Assumption Based Active-Learning Technique for Classification of Remote Sensing Images

Swarnajyoti Patra, and Lorenzo Bruzzone, *Fellow, IEEE*

Outline

1. Introduction
2. Proposed Method
3. Experimental Results

Outline

Resume

- Active-learning technique for solving remote sensing image classification problems with SVM classifiers.
- Main property: robustness to biased (poor) initial training sets.
- Considers the 1-D output space of the classifier to identify the most uncertain samples whose labeling and inclusion in the training set involve a high probability to improve the classification results.
- Histogram-thresholding algorithm is used to find out the low-density region in the 1-D SVM output space.

Active Learning

Algorithm 1: Active-learning process
Step 1: Train the classifier G with the training set L (which initially has few labeled samples).
Repeat
  Step 2: Select a set of samples from the unlabeled pool U using the query function Q.
  Step 3: Assign a class label to each of the queried samples by a supervisor S.
  Step 4: Add the new labeled samples to the training set L.
  Step 5: Retrain the classifier G.
Until the stop criterion is satisfied

Outline

Basic Concept

1) *Basic concept:* In the proposed approach, we estimate the uncertainty of each sample according to the output score of the SVM classifier [6]. Initially, the classifier is trained with the few available (and possibly biased) labeled samples. After training, a histogram is constructed in the 1-D output space of the classifier by considering the output scores of the samples in $[-1, +1]$. In the histogram, the region of interest is quantized into N mutually exclusive intervals called bins. We assume that all bins have equal widths (uniform quantization). The probability to have the output in a given bin is given by the number of samples whose output scores fall in that bin divided by the total number of samples in the histogram (i.e., the samples given as input to the classifier). Since the classifier ranks samples from the most likely members to the most unlikely members of a class, according to the cluster assumption (the decision boundary has to lie in low-density regions [4] of the kernel space), the samples whose output scores fall in the valley region of the histogram are the most uncertain. Thus, we can work in the 1-D output space of the classifier to identify the uncertain samples by finding a threshold on the histogram which is passing through this valley region,

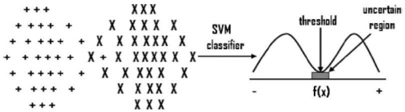

Fig. 1. Transformation of the original feature space into the 1-D classifier output space.

Entropy-based histogram thresholding

- In Kapur's method, an optimal threshold is determined based on the concept of entropy.
- Let ω_1 and ω_2 be two classes and H be the histogram of N bins generated by considering the output scores of the SVM classifier.

Let $p_i (i = 1, \ldots, N)$ be the probability of the ith bin. Assuming a threshold $t, t \in \{1, 2, \ldots, N\}$, the entropies of the classes ω_1 and ω_2 (denoted as $E_{\omega_1}(t)$ and $E_{\omega_2}(t)$, respectively) are computed as follows:

$$E_{\omega_1}(t) = -\sum_{i=0}^{t} \frac{p_i}{P_{\omega_1}(t)} \log_2 \left(\frac{p_i}{P_{\omega_1}(t)} \right)$$

$$E_{\omega_2}(t) = -\sum_{i=t+1}^{N} \frac{p_i}{P_{\omega_2}(t)} \log_2 \left(\frac{p_i}{P_{\omega_2}(t)} \right) \qquad (2)$$

where $P_{\omega_1}(t) = \sum_{i=0}^{t} p_i$ and $P_{\omega_2}(t) = 1 - P_{\omega_1}(t)$. To select a threshold on the histogram that separates the two classes ω_1 and ω_2 in the output space (i.e., that passes through the valley region of the histogram), we compute the entropy of classes ω_1 and ω_2 by assuming all possible values of the threshold t. Then, the optimal threshold t_0 is selected by maximizing the total entropy $E_{\omega_1}(t) + E_{\omega_2}(t)$, i.e.,

$$t_0 = \arg \max_{t \in (1,2,\ldots,N)} \{E_{\omega_1}(t) + E_{\omega_2}(t)\}. \qquad (3)$$

Multiclass active-learning algorithm

we consider each binary SVM classifier and separately select q (with q greater or equal to one) uncertain samples on the basis of the proposed query function. The q selected samples are those that, in U, have output scores closest to the detected threshold of the histogram generated by the output of the classifier. The threshold for each binary SVM is automatically detected by applying the entropy-based histogram-thresholding method described earlier. In greater detail, if we have n classes, n binary SVMs are initially trained with the current training set, and then, the functional distance $f_i(x)(i = 1, \ldots, n)$ is calculated for each binary SVM and for all the unlabeled samples $x \in U$. Then, the related histogram H_i is generated by considering the output score value in $[-1, +1]$. Thus, each binary SVM classifier generates a separate histogram considering its output score values. Then, a threshold t_i is selected for each histogram H_i by applying the entropy-based technique. Considering the ith binary SVM classifier, the q uncertain samples whose output score is the closest to the threshold t_i are selected. If, for a given classifier, there are no patterns whose output scores are in $[-1, +1]$, then the process of extraction of unlabeled pattern is stopped for that classifier. Thus, a total of $h \leq q \times n$ samples are chosen from the n binary SVM classifiers by considering only their uncertainty measure (h is lower than $q \times n$ if at least one sample is selected by more than one binary SVM or if there is at least one binary SVM which selects less than q samples).

Algorithm 2: Proposed fast cluster-assumption based active-learning technique

Step 1: Train n binary SVMs by using a small number of labeled samples. Let $f_i(.)$ be the decision function of the ith binary SVM classifier.

Repeat

  Step 2: $h = 0$.

  For $i = 1$ **to** n

   If $Cardinality(|f_i(x)| \leq 1) > q$

    Step 3: For the ith binary SVM classifier, generate the corresponding histogram H_i by considering the output score of the unlabeled samples $x \in U$, whose output value $f_i(x) \in [-1, +1]$.

    Step 4: Detect the threshold t_i from the histogram H_i by using the entropy-based histogram-thresholding technique.

    Step 5: For the ith binary SVM classifier, select the q samples from the pool U, whose output scores are closest to the threshold t_i.

    Step 6: $h = h + q$.

   Else

    Step 7: For the ith binary SVM classifier, select the samples from the pool U, whose output scores $f_i(x) \in [-1, +1]$.

    Step 8: $h = h + Cardinality(|f_i(x)| \leq 1)$.

   End if

  End for

  Step 9: Assign true labels to the h selected samples, and update the training set.

  Step 10: Retrain the n binary SVMs by using the updated training set.

Until the stop criterion is satisfied.

Outline

1. Introduction
2. Proposed Method
3. Experimental Results

Data Set Description

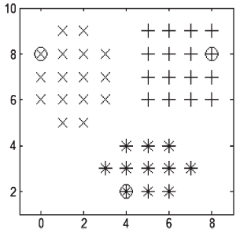

Fig. 2. Toy data set: The points represented with circles denote the initial training samples.

TABLE I
NUMBER OF SAMPLES OF EACH CLASS IN THE INITIAL TRAINING SET (L), IN THE TEST SET (TS) AND IN THE POOL (U) FOR THE PANEVEGGIO DATA SET

Classes	L	TS	U
Picea Abies	39	1135	1515
Larix Decidua	13	308	520
Pinus Mugo	6	160	234
Alnus Viridis	3	70	122
No Forest	40	1000	1560
Total	101	2673	3951

TABLE II
NUMBER OF SAMPLES OF EACH CLASS IN THE INITIAL TRAINING SET (L), IN THE TEST SET (TS) AND IN THE POOL (U) FOR THE PAVIA DATA SET

Classes	L	TS	U
Water	2	215	178
Tree areas	4	391	344
Grass areas	4	321	319
Road	12	613	975
Shadow	9	666	709
Red building	29	1620	2267
Gray building	7	427	590
White building	3	249	255
Total	70	4502	5637

Design of experiments

- We adopted an SVM classifier with radial basis kernel functions.
- We compared it with four other methods: 1) simple random sampling (RS); 2) MS; 3) MS-cSV; and 4) EQB.
- 5 Experiments:
    - Accuracy of the proposed technique with the other techniques by using 1 toy data set and 2 real data sets.
    - Robustness of the proposed approach when biased initial training samples are considered.
    - Computational load of the different methods.
    - Accuracy of the proposed technique varying the batch size.

Experiment 1 - Toy data

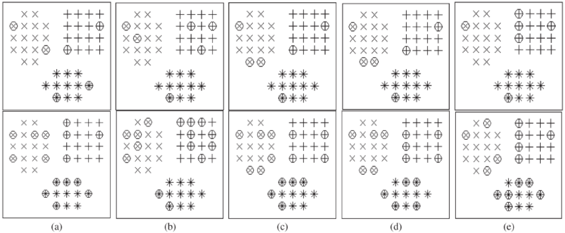

Fig. 3. Toy example showing a linear classification problem with three classes. The samples represented with circles denote the training samples selected by the (a) proposed, (b) RS, (c) MS, (d) MS-cSV, and (e) EQB methods after the (upper part of the figure) first and (lower part of the figure) fourth iterations.

TABLE III
OVERALL CLASSIFICATION ACCURACY ($\overline{OA}$) PRODUCED BY THE DIFFERENT TECHNIQUES AT DIFFERENT ITERATIONS (TOY DATA SET)

Itr No	**Training Samples**	$\overline{OA}$				
		Proposed	**RS**	**MS**	**MS-cSV**	**EQB**
0	3	97.43	97.43	97.43	97.43	97.43
1	6	100	92.30	94.87	94.87	89.74
2	9	100	94.87	100	100	100
3	12	100	92.30	100	100	100
4	15	100	97.43	100	100	100

Experiment 2 - Real data

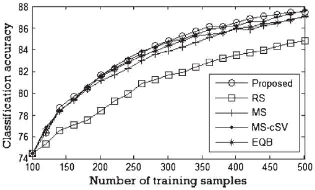

Fig. 4. Average classification accuracies over 20 runs provided by the proposed, RS, MS, MS-cSV, and EQB methods for the Paneveggio data set.

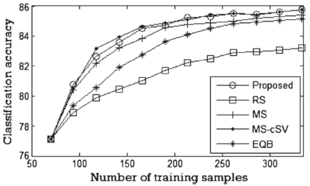

Fig. 5. Average classification accuracies over 20 runs provided by the proposed, RS, MS, MS-cSV, and EQB methods for the Pavia data set.

TABLE IV

AVERAGE OVERALL CLASSIFICATION ACCURACY ($\overline{OA}$), ITS STANDARD DEVIATION (s), AND KAPPA ACCURACY OBTAINED ON 20 RUNS FOR DIFFERENT TRAINING DATA SIZES (PANEVEGGIO DATA SET)

Methods	$\lvert L \rvert = 361$			$\lvert L \rvert = 421$			$\lvert L \rvert = 501$		
	$\overline{OA}$	s	kappa	$\overline{OA}$	s	kappa	$\overline{OA}$	s	kappa
Proposed	86.12	1.01	.792	86.87	0.91	.803	87.48	1.03	.812
RS	82.83	1.87	.742	83.74	1.90	.756	84.85	1.47	.772
MS	85.10	1.55	.778	85.87	1.92	.789	86.99	1.43	.806
MS-cSV	85.84	1.41	.788	86.56	1.75	.799	87.47	1.45	.812
EQB	85.39	1.57	.781	86.08	1.39	.792	87.07	1.33	.807

TABLE V

AVERAGE OVERALL CLASSIFICATION ACCURACY ($\overline{OA}$), ITS STANDARD DEVIATION (s), AND KAPPA ACCURACY OBTAINED ON 20 RUNS FOR DIFFERENT TRAINING DATA SIZES (PAVIA DATA SET)

Methods	$\lvert L \rvert = 166$			$\lvert L \rvert = 286$			$\lvert L \rvert = 334$		
	$\overline{OA}$	s	kappa	$\overline{OA}$	s	kappa	$\overline{OA}$	s	kappa
Proposed	84.50	1.00	.807	85.46	0.82	.819	85.75	0.62	.823
RS	81.00	2.29	.764	82.95	1.09	.788	83.23	1.02	.791
MS	83.86	1.74	.800	85.17	0.93	.816	85.43	0.70	.819
MS-cSV	84.65	1.12	.809	85.40	0.84	.818	85.76	0.77	.823
EQB	82.75	2.36	.786	84.94	1.26	.812	85.14	1.32	.815

Experiment 3 - Biased initial training samples

- Initial training sets were defined by taking two samples for each class (real data), respectively.

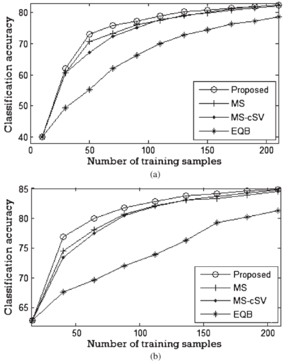

Fig. 6. Average classification accuracies provided by the proposed, RS, MS, MS-cSV, and EQB methods for the (a) Paneveggio and (b) Pavia data sets by starting with biased labeled samples.

Experiment 4 - Computational time

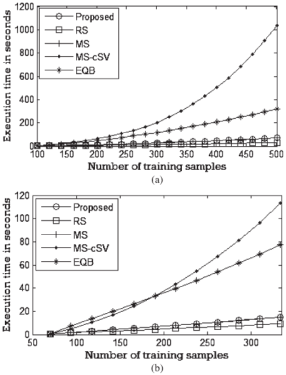

Fig. 7. Computational times taken by the proposed, RS, MS, MS-cSV, and EQB techniques at each iteration for the (a) Paneveggio and (b) Pavia data sets.

Experiment 5 - Varing batch size, h

- For each binary SVM, the number of selected uncertain samples q was varied in the range 2, 3, 4, and 5.

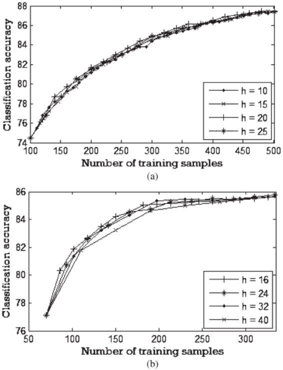

Fig. 8. Average classification accuracy provided by the proposed approach considering different values of batch size h for the (a) Paneveggio and (b) Pavia data sets.