ICA-based classification of sea ice SAR images

J. Karvonen & M. Simila
Finnish Institute of Marine Research, Finland

Keywords: SAR, sea ice, independent component, ICA, classification, texture

ABSTRACT: We have applied Independent Component Analysis (ICA) to sea ice SAR data to produce vector bases describing the elementary textural features in sea ice and use the projections of sea ice SAR image data to the sets of these basis vectors for classifying the SAR data. In this paper we solely concentrate on the classification based on the texture.

1 INTRODUCTION

The Finnish Institute of Marine Research (FIMR) and Finnish Ice Service (FIS), which is an operational part of FIMR, provide sea ice information for navigational purposes in the area of the Baltic Sea, especially in the Gulf of Bothnia and Gulf of Finland. During a normal ice winter, all the Finnish harbours are surrounded by sea ice, and navigational sea ice information is necessary. FIMR, in cooperation with the Swedish Meteorological and Hydrological Institute (SMHI), buys about 100 Radarsat-1 ScanSAR Narrow mode and ScanSAR Wide mode images over the Baltic Sea each winter, and also plans to utilize the Envisat ASAR instrument data exist. The images currently in operational use are mainly Radarsat-1 ScanSAR Wide mode images. The SAR images are used in making ice charts and also in producing an automated sea ice classification map (Karvonen, 1998). Because of the dark and cloudy winters, SAR images constitute a very important source of sea ice information.

The automated classification of the sea ice SAR data has gradually been developed. It is not possible to classify sea ice from SAR data based on intensity only very well (Simila 2001), and some further information is necessary. This paper describes an attempt to analyse the texture content of the data by applying the Independent Component Analysis (ICA). A special strength of the ICA-algorithm is that it computes a set of basis vectors that reflect regularly occurring patterns or texture primitives in the training data. Hence, e.g., the texture classification of Brodatz textures (Brodatz, 1966) can successfully be implemented by utilizing basis vectors computed for each texture separately. Our motivation to use the ICA-based approach was to find out if the adaptivity property of the ICA-algorithm could extract some useful features, which could be used to describe the morphology of ice fields from SAR images. In our experiments we noticed that this was possible for a fine resolution (1 m) optical Ikonos data of forested areas. However, straightforward applying of ICA cannot accomplish our goal. One can discern at least four major reasons for this. First, the echoes of a C-band radar are modified mainly by the ice surface roughness which is linked to the ice thickness (the most interesting property) only indirectly and in a complex manner. Second, the ground-truth available is not very accurate, and this complicates the training phase. Third, the ICA method is, due to its adaptive nature, sensitive to noise appearing in SAR images. Fourth, the resolution of the data (100 m) is relatively low when the main targets of interest (ice ridges) have only a width from a few metres to some tens of metres.

We have used four Radarsat-1 ScanSAR Wide mode SAR images from the years 2002 and 2003 for training the algorithm, and 10 Radarsat-1 ScanSAR Wide mode images from the winters 2002 and 2003 as a test set. Based on this test data, it is obvious that this method produces some additional information of the sea ice, although a detailed geophysical interpretation of the results is difficult.

2 INDEPENDENT COMPONENT ANALYSIS

The Independent Component Analysis (ICA) is a statistical data analysis method that has gained popularity during the last decade. In ICA the measured sample vectors are thought to be linear mixtures of
some underlying signals. These underlying signals (basis vectors) are determined adaptively from a large collection of sample signals of interest. The only constraint imposed is that the coordinates of a given signal in the ICA basis should be as independent as possible (Cardoso 1998, Hyvarinen 2000, Hyvarinen 2001). It has been noted that in practice applying an ICA basis usually leads to a sparse representation of a given signal, i.e. the representation contains only relatively few basis vectors. A characteristic feature of the ICA is that the set of basis vectors used in the representation change from one sample signal to another, unlike in the Principal Component Analysis (PCA).

The ICA model can be expressed as

\[ x = As \]  

(1)

where \( x \) is the measured data, \( A \) is the unknown basis vector matrix and \( s = (s_i) \) contains the unknown independent components which are the coordinates of \( x \) in the basis given by \( A \). It can also be put in the form \( s = Wx \), where \( W = A^{-1} \) is called the separating matrix. The estimation in the model (1) is performed by trying to find a solution or rather, an approximative solution, for the problem

\[ y = Bx \]  

(2)

where \( y = (y_i) \) are independent and, thus, the same as \( s_i \)’s.

The mutual information \( I \) is chosen as the measure of dependence. Mutual information is defined for vector \( y = (y_1, \ldots, y_m) \) as

\[ I(y) = \sum_{i=1}^{m} H(y_i) - H(y) \]  

(3)

where \( H \) refers to the differential entropy of a random variable. A direct computation of (3) shows that the joint distribution of \( y \) and the product distribution of the marginal distributions of \( y_i \) are equal if \( I = 0 \). In this case the random variables \( y_i \) are independent. Hence, one can base the ICA-algorithm on search of such mapping \( B \) that \( y = Bx \) and the mutual information between \( y_i \) is minimized. It can shown (Hyvarinen 2001) that to find such mapping \( B \) is equivalent to maximizing the non-gaussianity of the \( y_i \). The FastICA algorithm (Hyvarinen 1997), which we have been using, is based on this principle.

Whitening of the input data is performed as a preprocessing step, making the computations much easier by removing pairwise correlations and making the separating matrix \( W \) in the whitened coordinate system orthogonal (thus having only \( n(n-1)/2 \) free parameters for an \( n \times n \) orthogonal matrix).

It can be shown, e.g. by the well-known Lagrangian method, that for a function \( F(w) \), on the condition \( w \) is on the unit sphere, i.e. \( ||w||=1 \), the gradient at the maximum points to the same (or opposite) direction as \( w \). The derivation of this family of fixed point algorithms, known as FastICA, is based on this, and the algorithms simply iteratively update \( w \) to be the gradient of the selected measure of non-gaussianity, and then normalizing \( w \) at each iteration such that \( ||w||=1 \). For maximizing negentropy the update rule becomes:

\[ W \leftarrow E\{xg(w^T x)\} - E\{g'(w^T x)\} w \]  

(4)

where \( g \) is a function, which is the derivative of a nonquadratic function \( G \) used in the estimate of negentropy (Hyvarinen 1998). The expectation operations \( E \) should in practice be replaced with the corresponding sample means.

This algorithm has a cubic convergence and no adjustable parameters. This is a way to find one independent component (IC). To find multiple IC’s we first note that for whitened data the separating vectors \( w \), corresponding to different IC’s are mutually orthogonal. This leads to an implementation where after each iteration we orthogonalize the \( w \)'s. In practice we do this using the symmetric orthogonalization, in which no vectors are privileged over other (which is the case in e.g. Gram-Schmidt orthogonalization (Arfken 1985): let \( A^{{-1}} = W = (WW^T)^{-1/2}W \) where \( (WW^T)^{-1/2}W = (WW^T)^{-1/2}W \) i.e. we utilize the eigenvalue decomposition \( FAF^T \) of \( WW^T \) where \( A \) is a diagonal matrix.

3 DATA PREPROCESSING

First, the SAR images are rectified to Mercator projection and a land area mask is applied. Then an incidence angle normalization designed for Baltic Sea ice (Makynen 2002) is applied to the data. Then anisotropic diffusion filtering (Perona 1990) with some post filtering is applied to the SAR data and so-called featured areas, containing either pointwise features or edgelike features, are located in the image Karvonen 2003. Only these featured areas are used in the texture classification and training of the classification algorithm. In our case the input vectors to the ICA are sets of adjacent image pixel values around a featured location i.e. image windows scanned in a predefined order. We have used round-shaped data windows instead of square ones.

The noisy nature of SAR data requires that one selects the ICA training data carefully. We have tried to decrease the impact of the noise by utilizing in the training only data extracted from the featured areas of the images. From these candidates only the vectors having clearly multimodal (smoothed) histogram, with histogram peak-to-valley difference exceeding a given threshold, are used in the training. This kind of distributions indicate a presence of some sort of discontinuity.
We use our SAR data in 200 m resolution for the ICA classification, this down-sampling by averaging reduces the noise, and also reduces the computation time. Some visual inspection of the ICA training with our SAR data has shown that the basis vectors are similar in multiple resolutions, and even coarser resolution could be used for training and then applying these basis vectors in finer resolutions.

The texture features are computed as statistics for image segments, so a segmentation of the incidence angle corrected image is required. We currently use a simple intensity mean based segmentation of the images. The segmentation is performed using the isodata clustering algorithm (Ball, 1966), a variant of the k-means algorithm (Linde 1980). Instead of a predefined number of clusters of the k-means algorithm, isodata algorithm produces a data-dependent number of clusters adjusted by some input parameters (minimum distance between clusters, maximum variance inside a cluster, minimum number of samples in a cluster). In the isodata algorithm the clusters are merged and split according to these criteria during the iteration. The values used in the isodata segmentation are not the pixel values, but means computed in a round-shaped window with a given diameter D around each pixel. For each such window the gradient inside the window is computed and thresholded with a threshold relative to the maximum and minimum gradients inside the window, 

\[ T = \alpha (g_{\text{max}} - g_{\text{min}}) \]

where \( \alpha < 1 \) is a coefficient. The locations of the gradient values exceeding \( T \) are considered as (linear) edge pixel locations. The connected edge contour is formed as follows. The direction of the edge is defined to be the first principal component of the edge pixel locations. The location of the edge is determined by assigning the principal component vector to the center of mass of the gradient pixel locations. Finally, the mean in the window is computed only including those pixel values which are on the same side of the edge as the mid-pixel of the window, and if the mid-pixel happens to be at the edge, the mean is computed for the edge pixels only.

4 CLASSIFICATION ALGORITHM FOR SAR DATA

Image segments were chosen as basic units of image interpretation. To each segment an array of ICA statistics is assigned to describe the textural variation inside the segment. We can then perform a clustering of the ICA statistics computed for a training set, and produce a set of classes, described by the cluster (class) means.

4.1 Training

The first phase of the algorithm is to train the algorithm with a data set which covers the possible cases. The steps of the training phase are:

1. Perform incidence angle normalization for the SAR image.
2. Find featured areas in the image (i.e. areas containing non-gaussian data with reasonable variance).
3. Create ICA bases using the windows located in the locations of the areas found in the previous step. Bases can be computed in multiple resolutions.
4. Filter the basis vectors using an edge preserving filtering.

Figure 1. An example of a set of basis vectors computed from SAR data (top), and for comparison a set of basis vectors computed from optical data (bottom).

Figure 2. An example of basis vector filtering. Original vectors (top) and their filtered counterparts (bottom).
5. If multiple bases were computed, combine the bases into one set.

6. Prune duplicates that are very similar to some other vector(s) and also prune some small features located at the edge of the (2-D) window corresponding to a basis vector. The matching is done by cross-correlation, first locating the center of mass for thresholded gradient window of each window. The cross-correlation between two vectors is computed by rotating another of the vectors in increments of $\alpha_1$ (a parameter) and using the largest cross-correlation.

7. Cluster the basis vectors based on a large number of features (we just use a k-means type algorithm for that, with $k=10-15$). An practical example is shown in fig. 3.

8. Select basis vectors from the combined set that best fit a large number of samples drawn from the training data. The criterium is based on the sum of the absolute ICA coefficient values cumulated to each basis vector, but only cumulating the value when the absolute ICA coefficient value exceeds a given threshold. Also here each basis vector is rotated in increments of $\alpha_2$ and the largest ICA coefficient for each vector is used. A given number of basis vectors from each basis vector class are selected (see fig. 3).

9. Expanding the basis, by rotating each basis vector in increments of $\alpha_3$.

10. Classify each featured pixel location. The class is defined by the cluster of the best matching basis vectors, or if the absolute value of the cross-correlation is less than a threshold, the class is set to an "undefined" class, i.e. there are $k+1$ classes.

11. Segment the SAR image.

12. For each SAR segment attach a distribution of the featured pixel classes computed in the previous stage. The pixels at the segment edges are not used in the computation (segment edges may contain featured pixels). The distribution is normalized to one, and the number of featured pixels in each segment is used separately.

13. The segmentwise normalized distributions and numbers of featured pixels are clustered separately, e.g. using the isodata algorithm, producing two sets of cluster centers. This is performed for the segments of the whole training material, i.e. several images. These cluster means are used in the classification.

4.2 Classification

After the training has been performed any image can be classified utilizing the training data. The steps of the classification are as follows:

1. Perform incidence angle normalization for the SAR image.

2. Find the featured areas (as in training) for the image.

3. Classify each featured pixel location using the basis vector classes computed in the training.

4. Segment the SAR image.

5. For each SAR segment attach a distribution of the featured pixel classes.

6. Classify each segment, the class corresponds to the smallest euclidean distance between the cluster centres and the segment features. We get two classifications, one simply based on the number of featured segments in the segment divided by the segment size, and the other based on the normalized distribution of the elementary features. We here call these two features roughness and texture, respectively.

5 EXPERIMENTAL RESULTS

The sets of basis vectors derived from Radarsat-1 ScanSAR Wide mode data seem to be rather complicated in shape, there are e.g. no simple edges alone, but there are also some additional features. This makes interpretation of such a texture element complicated, because it usually contains several elementary features. The set of basis vectors, classified into 14 categories, we have used in the experiments presented here is shown in fig. 3 (top), and from this set only two vectors from each class were selected to be used in the final classification, this set is also shown in fig. 3 (bottom). The rotation angle step, $\alpha_3$, was 20 degrees producing 18 rotated versions of each basis vector.

Here we show the classification results for two Radarsat-1 ScanSAR Wide mode images, both were acquired on Feb 27th 2003, figs. 4 and 5. The results for the other test images were similar. For comparison, also the ice chart for this day is shown in fig. 6. The tone selection is such that tones close to each other represent similar textures in the texture feature space. And in the roughness image (lower left) the tone goes from blue (or black, low roughness) to red (or white, high roughness).
Figure 3. A set of basis vectors derived from Radarsat-1 ScanSAR Wide mode data (on the left), classified into 14 classes. The basis vector classes are separated by white horizontal lines, and a set of selected basis vectors (on the right). Two vectors from each class of the basis vectors shown on the left were selected.

By comparing the classification results and ice chart, it can be seen that the texture (second panels from top of the figures 4 and 5) of fast ice and new ice is rather similar, and the texture of consolidated close ice pack and compact or very close ice pack are also close to each other but different from the previous two. And also the texture of level ice differs from these two main classes. The texture of open ice and open sea depends on the amount of ice floes among the open sea and on wind conditions (waves). The roughness (third panels from top of the images 4 and 5) seems to be lowest for level ice, higher for close ice and even higher for consolidated, compact or very close ice. By combining the texture and roughness classifications we get a classification describing the ice surface inside segments as seen by the C-band SAR instrument. This description is usually in finer resolution than the symbols and segments drawn on the ice chart, making the verification difficult, especially for smaller segments.

Figure 4. A Radarsat ScanSAR Wide mode image over the Gulf of Finland (© Canadian Space Agency), acquired on Feb 27th 2003 (top), texture classification (second from top), roughness classification (third from top) and combined classification (bottom).
We have tested the use of elementary texture primitives derived from training SAR data for classification of sea ice SAR images. It seems that the method can distinguish between some elementary types of sea ice. By combining this method with intensity (segment mean intensity) classification and local autocorrelation classification, we can get an improved classification scheme compared to our earlier classifier (Karvonen 1998).

The method does not distinguish between rafted ice and ridged or hummocked ice very well. This may be because very often both these types of ice can co-exist inside same segments. Possibly some supervised training or selection of basis vectors could improve this situation.

The segmentation we have used in these experiments is based on the intensity only. Next step is to include the local image autocorrelation into the segmentation scheme, and also improve the precision of the segmentation.

The execution times including preprocessing and classification are typically about 20-30 minutes for a scene, on a 900 MHz AMD Athlon PC, the ICA
classification alone taking around 10 minutes per scene.

Verification of the classification results is currently based on visual comparison between ice charts and classification images. We have only ice thickness maps in digital form, which enables a direct comparison with the classification, but currently no ice type maps can be derived from the ice charts provided by FIS, and mean thicknesses of (usually relatively large) ice fields are not very descriptive for this purpose.

REFERENCES


