Chapter 9

I-FAC: Fuzzy Associative Classifier for Object Classes in Images

In this chapter we present our algorithm I-FAC [MCS10] which adapts fuzzy associative classification to fit the image classification perspective, by leveraging Speeded-Up Robust Features (SURF) that can be extracted from images [BETG08]. SURF is a fast scale and rotation-invariant interest point detector and descriptor for images. These interest points which can vary in number from image to image, can be used for further processing, like clustering and classification. Generally, obtaining the negative class set $C_N$ is an issue in image classification due to its ill-defined nature as compared to the positive class $C_P$. Effectively, the negative class set $C_N = U - C_P$, where $U$ is the universal set of all images. But, conventional classifiers need both positive and negative classes for training. Because $C_N$ is not well-defined, classifiers so trained (on a subset of the negative class) may not perform well on disparate test images. The advantage of I-FAC is that only positive class samples are required to train the classifier, with no reliance on negative class samples for training and without the need for unlabelled examples or outliers. In the literature, one-class classifiers which rely on unlabelled examples [FWE03] or treat outliers and noise as negative examples for training [MY01], have been proposed.

In the bag-of-words (BOW) approach, each SURF point belongs only to one of the clusters in the code-book, which is created by applying crisp clustering (like $k$-means) on a sizeable set of images. Using fewer number of clusters would avoid synonymy (two or more clusters having the same meaning), but would at the same time give rise to polysemy (one cluster with two or more distinct meanings). Thus, in BOW, deciding upon the number of clusters that should be used is an important, but difficult task, because of which $1000-3000$ clusters are generally used. But, I-FAC relies on fuzzy $c$-means (FCM) clustering [Dun73] and creates far less number of clusters ($\approx100$) using only the positive class training images, as compared to the number of clusters used for the code-book in BOW, thus avoiding synonymy. Due to the fuzzy nature of clusters, it is able to address polysemy as well.

The main contributions of this work are:

- Use of fuzzy sets and logic in object class classification in images. By doing so we can deal with polysemy and synonymy better as compared to crisp sets. This is reflected in the experiments section (Section 9.2) where usage of fuzzy sets yields better results than obtained using crisp sets.

- I-FAC is an associative classifier, which relies on frequent itemsets. Frequent itemsets capture all dominant relationships between items in a dataset. This helps in making the algorithm more resilient to noise.
9.1 I-FAC and Image Classification

This section describes key components of I-FAC for image classification. SURF points extracted from $C_P$ are clustered using FCM clustering, followed by fuzzy ARM. The fuzzy association rules are then transformed into fuzzy classification rules during training. For actual classification, membership values for the SURF points extracted from a test image are interpolated (using cosine-similarity) with the centers of the fuzzy clusters generated previously in the training phase. The classification is then done by calculating the cumulative fuzzy information gain, in conjunction with a threshold $\delta$.

9.1.1 SURF Point Generation

The first step in I-FAC extracts SURF points from images in the positive class training dataset, with no negative class images being used. Assuming we need $k$ fuzzy clusters, we run FCM (cosine distance metric is used) on all the $n$ SURF points extracted from the training images. From the $k$ fuzzy clusters, for each SURF point we have its membership value ($\mu$) in each of the $k$ fuzzy clusters. The $k$ membership values for each SURF point are then used to transform SURF-point-based representation of the images into a fuzzy-cluster-based representation. Each SURF point is represented as a separate record, with each record consisting of $k$ cluster (attribute) ids and corresponding $\mu$ pairs $\langle$cluster_id, $\mu$$\rangle$, followed by the positive class label, as shown in Figure 10.1.

\[
\begin{align*}
c_1 \mu_{1,1}, c_2 \mu_{1,2}, \ldots, c_k \mu_{1,k}, \text{positive_class_label} \\
\vdots \\
c_1 \mu_{n,1}, c_2 \mu_{n,2}, \ldots, c_k \mu_{n,k}, \text{positive_class_label}
\end{align*}
\]

Figure 9.1 Fuzzy-cluster based representation

9.1.2 Fuzzy Association Rule Mining

Subsequently we use FAR-HD, the fuzzy ARM algorithm (with appropriate minimum support) described in Chapter 5 to extract latent patterns in the form of fuzzy association rules from the fuzzy-cluster-based representation of the SURF points, as shown in Fig. 10.1. This algorithm is optimized to extract rules from very large datasets having many attributes (high dimensions), which is common in the image domain. The support $supp(I)$ of an itemset $I$, in the crisp domain, is defined as the proportion of transactions in the dataset which contain $I$. During fuzzy ARM, each of the $k$ dimensions corresponding to $k$ clusters is taken as an attribute. The membership values of a SURF point in each of the $k$ clusters provide the values for these $k$ attributes (Fig. 10.1). Moreover, support, as defined for crisp association rules, has been generalized in a suitable way for the fuzzy environment [CCK05, DHP06]. A t-norm $T$, given by Eq. [10.3] satisfies the condition $T(x, 1) = x, \forall x \in [0, 1]$, with fuzzy sets $A$ and $B$ (in a finite universe $D$) lying in the range $[0, 1]$. The cardinality of a fuzzy set in $D$ is defined by Eq. [10.4]. Using Equations 10.3 and 10.4 we get fuzzy support, defined in Eq. 10.5. $T_M (\min)$ t-norm, the most popular
The \textit{t}-norm, has been used in I-F AC to derive the rule-set \( R \) (with \( m' \) rules) from the fuzzy-cluster-based representation of SURF points.

\[
A(x) \cap_T B(x) = T(A(x), B(x)) \tag{9.1}
\]

\[
| A | = \sum_{x \in D} A(x) \tag{9.2}
\]

\[
sup(A \Rightarrow B) = \frac{1}{| X |} \sum_{x \in D} (A \cap_T B)(x) \tag{9.3}
\]

\[
H(Y) = -\sum_{i=0}^{z} p_i \log p_i \tag{9.4}
\]

\[
H(Y \mid X) = \sum \text{Prob}(X) H(Y \mid X) \tag{9.5}
\]

\[
IG(Y \mid X) = H(Y) - H(Y \mid X) \tag{9.6}
\]

### 9.1.3 Fuzzy Associative Classifier Training

Entropy and information gain are calculated for each rule in the rule set \( R \). Given a rule of the form \( X \Rightarrow Y_i \), where \( X \) is an itemset composed of varying number of attributes \( a_1, a_2, \ldots, a_l \), the entropy of \( X \) is given by Eq. \( 9.4 \) where \( z \) is the number of classes being considered. \( Y_i \) is the class label pertaining to the rule. In the crisp case, the fraction of records in the dataset where \( Y_i \) occurs is denoted by \( p_i \). But, in the fuzzy case, \( p_i \) of \( Y_i \) is calculated by taking the maximum membership value among all attributes (clusters) in each record in which \( Y_i \) exists. The average conditional entropy \( H(Y \mid X) \) for \( Y \), with respect to \( X \), is given by Eq. \( 9.5 \). In the fuzzy case, \( H(Y \mid X) \) is calculated using a \( t \)-norm (\( T_M \) \( t \)-norm in this case). The frequency for each record involving \( Y \) is a function of the minimum membership value of all attributes (fuzzy clusters) \( a_1, a_2, \ldots, a_l \) that are involved in \( X \). The information gain \( IG(Y \mid X) \) is given by Eq. \( 9.6 \).

ARM generates a large number of rules, most of which are redundant, and are pruned by I-FAC. The information gain of each rule and rule length \( i.e. \) number of attributes in each rule, is used for the pruning process. Each rule \( r_q \) is compared to all \( r_{q+1} \) to \( r_{m'} \) rules. A given rule \( r_q \) (with information gain \( IG_q \) and rule length \( rl_q \)) is pruned \( (R = R - r_q) \) if there exists another rule \( r_s \) (with information gain \( IG_s \) and rule length \( rl_s \)) which is a superset of \( r_q \), and \( rl_q < rl_s \) and \( IG_q < IG_s \). After pruning, the size of \( R \) reduces from \( m' \) to \( m'' \).

### 9.1.4 Image Classification

The actual classification stage is relatively straightforward, and is dependent upon the rule set \( R \) derived in the training stage. But before that, we identify all the \( n' \) SURF points in the image being classified. The centers of the \( k \) clusters generated during the training phase are used to calculate the fuzzy membership of each of the \( n' \) SURF points in each of the \( k \) clusters. For each SURF point \( s \), cosine similarity values are calculated between \( s \) and each of the \( k \) cluster centers. The normalized similarity between each cluster center \( c \) and \( s \) is denoted as fuzzy membership value \( \mu_{sc} \). Similar procedure
followed for all the $n'$ SURF points, at the end of which we have a record (in fuzzy cluster format representation) for each of the $n'$ SURF points. The membership value $\mu_{ij}$ for each cluster $c_i$ in each of the $n'$ records is aggregated to get one record $cr$ with cumulative membership values in each of the $k$ clusters of the whole image (Eq. 10.9).

Then, each rule $r$ in the rule set $R$ (with $m''$ rules) is applied to this cumulative record $cr$. When $r$ is applied, we identify each of the $t$ attributes (clusters) that are a part of the precedent (right hand side of the rule) of $r$. The cumulative fuzzy membership value ($c\mu_i$) for each of these $t$ clusters is extracted from $cr$. The product (Eq. 10.10) of the arithmetic mean of these cumulative fuzzy membership values and the information gain ($IG$) associated with $r$ is used to come up with a derived metric we call *fuzzy information gain* ($FIG$). The cumulative fuzzy information gain is calculated (Eq. 10.11) as each rule is applied on $cr$. If at the end $cumulative FIG \geq$ threshold $\delta$, then the image in question belongs to the positive class, or else it belongs to the negative class.

$$\mu_i = \sum_{j=1}^{n} \mu_{ij} \quad \text{where } i = 1 \text{ to } k \quad (9.7)$$

$$FIG = \prod (IG)(\frac{\sum_{i=1}^{t} c\mu_i}{t}) \quad (9.8)$$

$$cumulative FIG = \sum_{f=1}^{m''} FIG_f \quad (9.9)$$

### 9.2 Performance Study and Results

We have compared I-FAC (with minimum support $supp$ between 0.005 and 0.05, depending on dataset, fuzzification factor $m = 1.5$ and 100 fuzzy clusters) to two baseline approaches, namely BOW and SVM, both based on SURF points. The support value relies on how dense or sparse the dataset is, the number of items (singleton) involved in the dataset, and the average length of transactions in the dataset [ZKM01]. In BOW, we count how many times each visual word in the code-book occurs in an image. A feature vector consisting of weighted frequency of each ‘word’ from the bag of words is used for training and testing. The results for BOW have been taken from the baseline of [QFLG07], which uses 3000 clusters to create the code-book. To generate a single feature vector per image for SVM (libSVM implementation using RBF kernel) classification, SURF points from an image are combined using Latent Semantic Hashing. ROC Curve for SVM was calculated using a threshold for the probability of positive class. CALTECH Cars (Rear) background dataset was used as negative training set for BOW and SVM. I-FAC does not expect any negative class training set. The other datasets used are:

- **CALTECH Cars Rear**: The positive class training, positive class test, and negative class test datasets respectively are cars_markus, cars_brad, and the first 200 images from CALTECH-101 background class.

- **TUD Motorbikes**: CALTECH-4 motorbikes, TUD motorbikes, and 200 random images from CALTECH-256 clutter class were used for positive class training, positive class testing, and negative class testing respectively.
• **ETHZ Giraffes:** Training was done on 93 images of giraffes downloaded from Google Images. The positive class test and negative class test datasets were 87 giraffe images and the rest 168 images respectively from the ETHZ Shape Classes dataset.

• **GRAZ Bikes:** The positive class training and positive class test sets respectively are randomly picked 25 and 38 images from the GRAZ bikes dataset. The first 200 images from CALTECH-101 background class dataset were used as negative class test set.

• **CALTECH Faces:** 52 randomly picked images from the CALTECH Human Faces (Front) dataset were used for each of the positive class training and test sets. The first 200 images from CALTECH-101 background class dataset were used as negative class test set.

I-FAC consistently performs well on the basis of ROC Curve when compared to either BOW (by high margins on all five datasets) or SVM (by high margins on three datasets - Cars (Fig. 9.2), Faces(Fig 9.6), and Giraffes(Fig. 9.4), and by reasonable margins on the remaining two datasets (Figures 9.5 and 9.3). It especially performs very well at low false-positive-rates – FPRs (≤ 0.3), which is highly desirable for an image classifier. The performance of I-FAC can be attributed to two broad reasons. First, its fuzzy nature helps avoid polysemy and synonymy, which are common problems with BOW. Second, SVM has to deal with a lot of noise in the training images, which hampers the creation of a clear hyper-plane, affecting the assignment of probability with which the positive class occurs in a given image. This problem does not occur in I-FAC which uses only the positive class for training, and makes a classification decision based on cumulative FIG in conjunction with a threshold δ. For each dataset, δ has been determined by cross-validation on the respective positive and negative classes test sets. The variation in δ influences the variation of ROC curve for I-FAC in each dataset. Fig. 9.7 shows the variation in the ROC Curve as number of clusters and m of FCM is varied (for the Cars dataset), with best results achieved when 100 clusters and m = 1.5 were used. At m = 1.001 (m ≈ 1), FCM reduces to k-means, i.e. crisp clustering. Higher values of m (e.g. m = 2) gave worse results than those shown in Fig. 9.7.

![Figure 9.2 ROC Curves for CARS Dataset](image-url)
9.3 Summary of I-FAC

We present I-FAC, a novel fuzzy associative classification algorithm for object class detection in images using interest points. In object class detection, the negative class $C_N$ is generally vague ($C_N = U - C_P$; where $U$ and $C_P$ are the universal and positive classes respectively). But, image classification necessarily requires both positive and negative classes for training. I-FAC is a single-class image classifier that relies only on the positive class for training. Because of its fuzzy nature, I-FAC also handles polysemy and synonymy (common problems in most crisp (non-fuzzy) image classifiers) very well. As associative classification leverages frequent patterns mined from a given dataset, its performance as adjudged from its ROC curve is very good, especially at lower false-positive-rates when its recall is even better. I-FAC has the added advantage that the rules used for classification have clear semantics, and can be comprehended easily, unlike other classifiers, such as SVM, which act as black-boxes. From an empirical perspective (on standard public datasets), the performance of I-FAC is much better, especially at lower FPRs, than that of either bag-of-words (BOW) or SVM (both using interest points).
Figure 9.5 ROC Curves for GRAZ Bikes Dataset

Figure 9.6 ROC Curves for Faces Dataset
Figure 9.7 ROC Curves for varying clusters and $m$