1. Introduction

People can quickly recognize enormous number of rigid/non-rigid object, such as cars, faces, trees, regardless of the viewpoint, lighting, illumination and local deformation. How to recognize generic object has been a hard problem for long time in psychophysics, neurobiology and computation. Based on the research of psychophysics and neurobiology, human interprets the image scene (label class) depending on the shape of the objects, orientation the objects present and the context the objects locate [5]. Hence many object representations have been proposed capturing geometric and/or appearance information. In case of appearance based representations objects are represented holistically or in terms of parts and their spatial relationships.

The holistical appearance approach was first proposed by Turk and Pentland [54] in face recognition domain and by Murase and Nayar [44] in a more general domain. In this framework, the object is encoded via principal component analysis (PCA) and represented as an eigenspace - called eigenimage. The variation of object are encoded in the eigenfaces. Instead of operating the original image space, eigenimage can also work in some filter-bank feature space which characterizes the local statistics of the images with different orientation and scale. Because the eigenimage is computed using the whole image, it is very sensitive to partial occlusion, outlier, background clutter and segmentation. Leonardis et al [2, 3] further proposed a robust eigenimage recognition algorithm, which considered the partial occlusion. However, it is still hard to deal with occlusion, local deformation, disguise, clutter background, and object location and orientation robustly. It has poor class generalization ability.

To overcome the shortcomings of global appearance-based methods, many approaches developed recently represent objects in terms of local features [10, 45, 18, 52, 59, 51, 13, 55]. The approaches consider the appearance information of local features and/or the geometric spatial relationships between them and achieve the good performance on certain classification task and data base. However, those works either require the pre-segmented training images or have high computational complexity for model parameter estimation. For weakly supervised object recognition, the Bayesian framework in [13] only works with a small number of parts to build the model of objects due to the computational complexity of learning. It only has the discrimination power to distinguish the objects from background. The approach in [55] clustered all features
into clusters and was very sensitive the cluster algorithm and portion of the background features. Since image segmentation is NP-hard problem, large number of pre-segmented training images require huge human labor. In order to advance the state of the art of current object recognition methods and scale them to be able to deal with large number of objects and/or object categories, it is essential that the representations of objects can be learned directly from images, without prior segmentation. In such a scenario, the presence of the background features violate the assumption of many machine learning and classification algorithms. Even the best classification approach will fail because of the large number of irrelevant features. As a result feature selection is necessary first to discard the background features.

In this work, we study the generic object recognition with local features to deal with large variation of object appearance, even the clutter background. Compared with other methods, the central idea of our approach is to establish a simple and efficient feature selection algorithm to deal with irrelevant features and a new weak leaner in Boosting framework for object recognition where object models consist of a collection of local patches.

Specifically, the local features we use as a starting point for our research are the affine local region and difference-of-gaussian detector, recently developed by Mikolajczyk et al [42] and SIFT features proposed by David Lowe [35]. These features are represented by Scale Invariant Feature Transforms (SIFT) [35] descriptor, which is robust to large appearance changes. A strangeness measure is used to select the discriminative features from the detected feature set. The boosting learning framework is further used to select features and build the final classifier with strangeness as the non-parametric weak hypothesis. We apply and validate our approach on specific object recognition and general object category recognition. The training images for objects are either pre-segmented or weakly supervised.

The main scientific contribution expected from this work are:

- A new feature selection mechanism, which has the properties of both filter and wrapper approach. It has relatively low computational complexity and works even the training sample size is small.
- Incorporation of the new weak learner within the boosting learning framework, which can be used for both supervised object recognition and weakly supervised object category recognition, even with small number of training samples.

2. Background and Approach

As noted in the previous section, the central idea of this proposal is to establish a framework for generic object recognition where the training images contain the objects as well as the cluttered background. We adopt object models consisting of a collection of small local patches without considering their spatial relationship. The proposed method selects a “small set” of discriminative features, integrates the local discriminative evidence from them, and compensates for the repeatability of the feature detectors. This section briefly reviews previous work on local feature detectors and their associated descriptors. We describe SIFT features [35] and Affine Covariant Regions [41] that are used in this proposal.
At the end of this section we briefly outline our framework. The detail of our approach is relegated to Section 4 and 5.

2.1 Feature Detection

The central philosophy behind many feature detectors is to search for the “salient” regions and characterize them by a descriptor invariant to, brightness transformation, local deformation and illumination variation. One of the most popular feature detectors are Forstner corner [15] and Harris detector [21]. Forstner detected the corners by implicit least squares matching, using rectangular patches and Taylor expansion to re-express the accuracy in terms of the eigenvalues of the scatter matrix of the local gradients. Harris & Stephens [21] improved the localization performance by replacing the rectangular patches with Gaussian convolution window with a scale. Harris points are invariant to rotation and the support region is a fixed size neighborhood centered at the interest point.

Tuytelaars and Van Gool [53] construct small image regions depending on both corners and the intensity extreme. The intensity extreme are extracted based on the intensity profiles along rays emanating from the points. The nearby edges provide orientation and skew while the scale and stretch are given by the extreme of a 2D affine invariant function. Kadir and Brady [25] finds regions that are salient over both location and scale, which are invariant to rotation and scale changes. In this work, the saliency is defined in terms of local signal complexity or unpredictability, which is measured with Shannon entropy of local attributes. Matas et at [23] used a procedure to construct a maximally stable affine co-variant regions. The areas are selected from an intensity images using watershed image segmentation. The proposed regions are the connected components of pixels which are all brighter or darker than all pixels on the region’s contour, and are bounded by cycles of edge pixels.

To obtain invariance to scale changes, interest points can be extracted in the scale space of the image [33]. Mikolajczyk & Schmid [39] improved Harris Corner detector by introducing so called Harris-Laplace points, which are invariant to rotation and scale changes. The points are detected by a scale adapted Harris measure and selected in scale space by the Laplacian operator. The selected scale determines the size of the support region. Mikolajczyk & Schmid [26, 41] further introduced the Harris-Affine point detector. The detected features are invariant to affine image transformation and constructed by elliptical shape adaptation. More details can be found in [26]. Fig. 1 shows the affine covariant regions detected in the images, which are used for the weakly supervised generic object recognition.

Another powerful feature detector is Lowe Detector (DoG points) [35], which is invariant to rotation and scale changes. It is implemented efficiently by constructing a Gaussian pyramid and searching for local peaks in the scale space of difference-of-Gaussian (DoG), as shown in Fig. 2. Candidate feature points are localized to sub-pixel accuracy and selected based on measures of their stability. The selected scale determines the size of the support region. More details can be found in [35]. Fig. 3 shows the extreme of DoG regions detected. Compared with affine covariant region detector, Lowe’s detector detects more regions in the images, especially in the images with complex texture.
2.2 Feature Descriptors

The detected local features are further described by a descriptor, which are robust to the local deformation, rotation/affine transformation and illumination.

Many feature descriptors have been developed [24, 64, 62, 8, 19]. Recently David Lowe presented a very robust descriptor - Scale Invariant Feature Transforms (SIFT) in [35], as shown in Fig. 4. Once the location and scale have been assigned to candidate features, the dominant orientation is computed by determining the peaks in the orientation histogram of its local neighborhood weighted by the gradient magnitude. The descriptor is then formed by computing local orientation histograms (with 8 bin resolution) for each element of a $4 \times 4$ grid overlayed over a $16 \times 16$ neighborhood of the point. This yields a 128 dimensional feature vector which is normalized to unit length in order to reduce the sensitivity to image contrast and brightness changes. This descriptor provides robustness against localization errors and small geometric distortions, and has good repeatability across variations of scale and pose. Mikolajczyk & Schmid [40] evaluated those descriptors based on ROC curves of detection rate under rotation, scale change, affine transformation and illumination changes. The results show that SIFT descriptors obtain better results than others except for light changes.

2.3 Part-based Object Recognition

Object can be represented as collection of features, or “parts”, with the distinctive appearance and “shape”, the mutual spatial positions of those features. Lowe [35] developed a robust matching procedure to discard
large number of false matches. The works in [41, 39, 26, 42, 40, 14] used voting for object matching. In order to capture the uncertainty of descriptor and spatial relationship, probabilistic approach is widely used where the objects are modelled as random constellations of parts. The works in [12, 13, 22] modelled the feature appearance and shape together in a Bayesian framework. Considering the local feature appearance only, the works in [1, 18, 52, 59, 51, 10, 61] recognized the objects with different machine learning models from local features.

2.4 Proposed Approach

In order to scale current object recognition methods to be able to deal with large number of objects and/or object categories, it is essential that the representations of objects can be learned directly from images, without prior segmentation. It is called weakly supervised learning, where the object images are given as well as the background clutter. The presence of background clutter greatly increases the complexity of task and prohibits many object recognition algorithms to be used, which require the clean and pre-segmented training images. The motivation of our approach is shown in Fig. 5. Through the feature selection, the clutter background features are discarded and the discriminative features are selected, which build the parts of the objects and yield the local evidence for recognition. General Boosting strategy is used to integrate the local evidence and fuse them into final strong classifier.

In the proposed framework, a discriminative strangeness measure is used to do feature selection and measure the local evidence. The outline of the approach is shown by following:

1. Get the local appearance features for the input images.
2. Select the discriminant features for each class using strangeness measure.

3. Clustering the selected features to build the parts.

4. Object is represented by parts with the feature instances.

5. Boosting model learning with non-parametric weak classifier - the strangeness measure of each instance of each part - through validation.

6. Hierarchical classifier learning for weak supervised learning:
   - Objects vs. background classifier.
   - Object vs. all other objects classifier.

7. Testing the query image the classifier learned.

In the following sections, Section 3 will present the definition, properties of strangeness. Section 4 will show how to use strangeness for feature selection and the performance. Section 5 will present our generic learning model with Boosting and strangeness, and the experiment evaluation with different data bases.

3. Strangeness Measure

Section 2 discussed the main outline of the proposed approach, the detections and representations of local patches from images, and how to integrate the information from the local patches. In this section, a new informative measurement - Strangeness is exploited to measure the uncertainty of the individual features or sample with respect to the class labels. We demonstrate how the strangeness can be used to construct the decision boundary and derive its generalization capability. It can further be used to select discriminative local patches, integrate local evidence without modelling “complex” model space and parameter estimation.
3.1 Strangeness measure definition and the related work

There have been numerous attempts to assign probabilities to machine learning classifiers in order to interpret their decision with confidence and reliability over all possible classes. In fact, we can trivially convert every machine learning classifier’s output to a probability distribution by assigning the predicted class the probability between 0 and 1 to all possible classes. However, such estimation may in general not be good or overfitting due to inherent applied algorithm’s biases. Bayesian approach is often to use to estimate the posterior probability of the predicted class, which is argued to be always the best method and without “overfitting” problem. The only important issues in Bayesian method are how to choose a good prior distribution and how to efficiently calculate the posterior. However, the optimality of the Bayesian method is based on the assumption that the data we observe are generated according to one of the distribution models in the chosen class of models. While this assumption is attractive for theory, it almost never holds in practice. In practice, one usually uses relatively simple models, either because there is not enough data to estimate the “true” model and the computational complexity of estimation is prohibitive, or because our prior knowledge of the system is only partial. Even when very complex models are used, it is rarely the case that one can assume that the data are generated by a model in the class.

Instead of assuming a family of models, Vovk et al. [38, 46, 50] introduce an individual strangeness measure and construct the confidence machine using the algorithmic theory of randomness and Transduction methodology with a weak assumption of the model space, that the data is generated i.i.d. with some distribution $D$ but make absolutely no assumption about $D$ other than it is a fixed distribution. Several strangeness definitions were proposed [20, 38, 16], which need complex learning strategy and high computational cost. There are several simpler definitions which do not require complex learning procedure. If the example of class $j$ is sampled from some Gaussian model, the distance from example $x^j_i$ to the mean $\bar{x}_j$ is defined as the strangeness:

$$\alpha_i = \|x^j_i - \bar{x}_j\|,$$

where $\bar{x}_j = \frac{1}{N_j} \sum_k x^j_k$.

Without any absolutely assumption about distribution $D$ of $z$, $k$-nearest neighbor classifier is widely used in [46, 50, 30] to define the strangeness measure if the examples are measurable in some metric space. Assume we have $C$ classes, for class $c = 1, \cdots, C$, let us denote the sorted sequence (in ascending order) of the distances of example $x^c_j$ from the other examples with the same classification $c$ as $d^c_j$ and $d^{c,j}_l$ will stand for the $l$th shortest distance in this sequence. Let $d^{-c}_j$ denote the sorted sequence of distances containing examples with classification different from $c$. For each example, the individual strangeness measure is assigned as:

$$\alpha_j = \frac{\sum_{l=1}^k d^{c,j}_l}{\sum_{l=1}^k d^{-c,j}_l}. \quad (1)$$

The measurement for strangeness is the ratio of the sum of the $k$ nearest distances from the same class to the sum of the $k$ nearest distances from all other classes. This definition of strangeness is very natural and straightforward. An example is considered strange if it is in the middle of examples labelled in a different way and is far from the examples labelled in the same way. The strangeness of an example increases when
the distance from the example of the same class becomes bigger or when the distance from the other classes becomes smaller. The strangeness defined in Equation 1 is related to $k$-nearest neighbor classifier (KNN). However, for multi-class classification, the definition in Equation 1 does not consider the frequency of each class in the neighborhood of the example, as does in KNN classifier. As the result, we modify the definition in Equation 1 and re-define the $k$-NN strangeness as:

$$
\alpha_j = \frac{\sum_{l=1}^k d_{jl}^c}{\min_{n,n\neq c} \sum_{l=1}^k d_{jl}^n}.
$$

(2)

In the following subsection, we will discuss its properties and show how it is related optimal decision boundary and the posterior $P(c_i|x_i)$.

### 3.2 $k$-Nearest Neighbor Strangeness

In this section we will study the properties of $k$-nearest neighbor strangeness (as defined in Equation 2), how it can be used to build the decision boundary between classes and how it is related to the discrimination ability of each example. The Cover-Hart theorem [9] proves that asymptotically the generalization error of one nearest neighbor classifier can exceed by at most twice the generalization error of the Bayes optimal classification rule. They also showed that the $k$-NN error approaches the Bayes error (with factor 1) if $k = O(\log n)$ [9]. The generalization power of $k$-NN classifier enables the $k$-NN strangeness has the similar properties.

On average the examples with $\alpha = \text{const}$ build the piecewise linear boundary between class $c_i$ and all other classes. The smaller the strangeness of the example is, the larger the distance of the example to the decision boundary and the higher discriminative ability of the example with respect to other examples and class label. Asymptotically, the examples with $\alpha = 1$ will build the optimal boundary between two classes. Those examples can be considered as samples from the optimal Bayes classification boundary which serves as the ground truth if the data distribution and prior is known. We demonstrate the comparison between those two decision boundaries in synthetic experiments next.

We consider a two-class classification first. Let examples $(z_1, \cdots, z_n) = ((x_1, y_1), \cdots, (x_n, y_n)$) be drawn independently from the same distribution over $Z = X^d \times Y$ where $Y$ is the label space $(0, 1)$. For each class $c_i$, the data is generated independently from a Gaussian distributions $P(x|c_i) = N(x; \mu_i, \Sigma_i^{-1})$ and priors $p_i = P(c_i)$, $i = 0$ or 1. Applying the Baysian rule, the optimal boundary between two classes is a quadratic function if the two classes have different covariance matrix:

$$
f(x) = x^T (\Sigma_0^{-1} - \Sigma_1^{-1}) x - 2 (\Sigma_0^{-1} \mu_0 - \Sigma_1^{-1} \mu_1) x + (\Sigma_0^{-1} \mu_0 - \Sigma_1^{-1} \mu_1)^T - \log \frac{|\Sigma_1|}{|\Sigma_0|} - 2 \log \frac{p_0}{p_1}. \quad (3)
$$

If $\Sigma_0 = \Sigma_1 = \Sigma$, the decision boundary becomes a linear hyperplane:

$$
f(x) = (\mu_0 - \mu_1)^T \Sigma^{-1} x - \frac{1}{2} (\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) + \log \frac{p_0}{p_1} = w^T \cdot x + b, \quad (4)
$$

where $w = (\mu_0 - \mu_1) \Sigma^{-1}$ and $b = \log \frac{p_0}{p_1} - \frac{1}{2} (\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1)$. In order to demonstrate the properties of
strangeness, we compute the boundaries in two dimensional space using the Gaussian example, \(k\)NN classifier and strangeness, respectively. Let the means of two classes be \([0, 0]^T\) and \([5, 5]^T\) and the same covariance matrix \(\Sigma = \text{diag}(\sigma, \sigma)\). Both classes have the same number \(N\) of samples, that is \(p_0 = p_1 = 0.5\). The following experimental results show the optimal boundaries and boundaries constructed by \(k\)NN and \(k\)NN-strangeness with \(\alpha = 1\), with respect to different \(\sigma\), \(k\) and \(N\).

If two classes are well-separable, the boundaries constructed by \(k\)NN and \(k\)NN-strangeness are very close to each other, and both of them are close to the optimal boundary, even when both \(N\) and \(k\) are small. Given \(\sigma = 1\), Fig. 6 shows the obtained boundaries. If \(k\) is small, those boundaries have little improvement even when \(N\) increases (comparing Fig. 6(c) with Fig. 6(a) and (b)). If both \(N\) and \(k\) increase, both boundaries converge to the optimal boundary, as shown in Fig. 6(d). This is consistent with generalization ability of \(k\)-nearest neighbor classifier – \(k\)-NN error approaches the Bayes error (with factor 1) if \(k = O(\log n)\).

![Figure 6: The boundaries with different \(N\) and \(k\)](image)

From the examples above, it seems that the definition of strangeness has no advantage over \(k\)-nearest neighbor classifier. In real application, however, the classes are rarely well-separable; and the data are often in very high dimension space. Therefore we now examine the comparison with non-separable data set in a high dimensional space. Suppose \(\sigma = 3\), and a small number of training examples is given, \(N = 100\), both boundaries are not very close to the optimal boundary. However, the boundary constructed by strangeness is much more smooth and closer to the optimal boundary. Fig. 7 shows the boundaries with different \(k\). When \(k\) is small, the strangeness smoothes many isolated regions created by \(k\)NN classifier. With respect to the op-
If we consider the problem in a regularization framework, strangeness introduces a smooth penalty term, which is defined through the examples with the parameter $k$. Let $k = \sqrt{N}$. Fig. 8 shows the boundaries with different $N$. The boundaries constructed highly depend on the training examples while they both converge to the optimal boundary as $N \to \infty$. For each $N$, the boundary created by strangeness is much more smooth.

Now let’s consider the test error - the generalization ability of both classifiers. For each $N$, different training and testing sets are sampled in 100 trials. Fig. 9 shows the optimal Bayesian error, test errors of $k$NN and strangeness classifiers, and the corresponding error standard deviation over the trials. The error

![Figure 9: The test errors and their standard deviations with different $N$.](image)
of strangeness and its standard deviation are always lower than those of the corresponding \( k \)NN classifier, which is consistent with the conclusion from the comparison of the boundaries. The smooth term of the classification function reduces the test error and hence improves the generalization capability of the algorithm. Comparing with \( k \)NN classifier, the strangeness not only predicts the label of the test examples, but also gives the “reliability” or “confidence” of the prediction – the value of the strangeness measure: the higher the measure, the higher the uncertainty of the prediction.

We have introduced the strangeness measure and discussed how it can be used as a classifier to build the decision boundary, measure the uncertainty of the prediction and its generalization ability. In order to capture the deep insight of strangeness, we plan to investigate it in more details:

- Investigate its relationship to margin and posterior.
- The behavior of strangeness in high dimensional space and non-Gaussian distribution.
- Obtain the finite sample generalization bound for strangeness as a classifier using large margin principals and Lipschitz function theory [56].

4. Feature Selection based on Strangeness

Section 3 presents the definition of the strangeness, its properties and its generalization capability. The strangeness measure not only predicts the label, but also gives the relevance between samples or features and classes. In this section, we will present how such measurement can be used to evaluate the relevance of features for feature selection. Using the same strangeness definition, different feature selection algorithms are proposed to do variable selection and feature instance selection, which are demonstrated efficiently with different data set.

4.1 Related work on Feature Selection

In many supervised learning tasks, the input data is often represented by a very large number of features and/or high dimensional features. Even state-of-art learning algorithms cannot overcome the presence of large number of weakly relevant or irrelevant features. This is attributed to the “curse of dimensionality”, or the fact that irrelevant features decrease the signal-to-noise ratio, and confuse the learning algorithm. In addition, many learning algorithms become computationally intractable when the dimension is high. On the other hand, once a very good set of features is obtained, even the very basic and simple classifiers can achieve high performance. As a results, feature selection is crucial for efficient learning. Furthermore, there are many other potential benefits of feature selection: facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing the complexity of learned models, and defying the curse of dimensionality to improve prediction performance.

Various aspects of feature selection have been studied. One of the key aspects in feature selection is to measure the goodness of a feature subset to determine the optimal one. Different feature selection algorithm can be broadly divided into two categories: filters and wrappers. The filter approaches evaluate the relevance
of each feature (subset) using the data set alone, regardless of the subsequent learning phase. Because performing exhaustive search is usually intractable, a variety of search heuristics are applied. RELIEF [49] and information theoretic methods [57, 45, 18, 31] are the representatives of this class. RELIEF [49] assigns feature weights based on the consistency of the feature value in the $k$ nearest neighbors of every data points. The philosophy behind the information theoretic methods is that the mutual information between a relevant feature and the class labels should be high [7]. Nonparametric methods can be used to compute the mutual information involving the continuous features [31]. A feature can be regarded as irrelevant if it is conditionally independent of the class labels given other features. The concept of Markov blanket is used to formalize this notion of irrelevancy in [28].

On the other hand, the wrapper approach [27, 59, 52, 22] invokes the learning algorithm to evaluate the quality of each feature (subset). A learning phase, such as Boosting [59, 52], Bayesian approach [13], decision tree [45] is run on a feature subset and the feature relevancy is assessed by the estimation of the classification accuracy. Wrappers are usually more computationally demanding, but can be superior in accuracy when compared with filters. Both approaches involve combinatorial searches through the space of possible feature subsets with different types of heuristics.

Given the training features $F = (F_1, \cdots, F_N)$, where $F_i$ is a point in $\mathbb{R}^d$, there are two different feature selection directions: one is to select the optimal subspace along the column direction of the feature matrix $F$ - variable selection; the other one is to select the optimal sub-instance along the row direction of $F$ - feature instance selection. The first one is widely researched in machine learning field, where one assumes each instance of $F$ has some contribution for classification and tries to find the optimal subspace and compact representation [7, 27, 31, 49, 57]. The second one is widely investigated in computer vision community where the objects are represented by the local patches (features) (corresponding to feature instances in $F$) and the feature selection algorithms try to find the most discriminative and relevant patches to represent the objects [59, 22, 45, 18, 13]. In many cases, the number of features is different in different images. Finding the optimal subspace is also widely used in computer vision field, such Principal Component Analysis (PCA), Fisher Linear Discrimination (FLD) analysis, Independent Component Analysis (ICA) for holistic object recognition. Few works have involved with the feature selection in both directions of the feature matrix $F$. In the following sections, we will present strangeness based feature selection algorithms in either direction and the empirical evidence on the performance of these algorithms.

4.2 Feature Selection Algorithms

4.2.1 Variable Selection

In this case, the training set is often given with labelled fixed-length feature vectors. Each feature is described as an assignment of values $f = (f_1, \cdots, f_d)^T$ with $d$ dimension to a set of features $F = (F_1, \cdots, F_d)$ and one of $l$ possible classes $Y_1, \cdots, Y_l$. Feature selection is the task to try to find the optimal subset from $F$ which has equal or as close as possible classification performance to the one using the whole feature. It is closely related to the more general problems of dimensionality reduction and efficient data representation, while it provides a much simpler approach as it does not require the evaluation of new complex functions of
the irrelevant features.

Mutual information between features and class labels is commonly used in this area as the evaluation function. Based on Fano’s bound, the Bayes error of $M$-class classification problem is lower bounded by the mutual information plus some constant:

$$
p_e \geq \frac{H(X|Y) - H(p_e)}{\log(M - 1)} \geq \frac{H(Y) - I(X;Y) - 1}{\log(M)}.
$$

Thus the lower bound on error probability is minimized when the mutual information between $X$ and $Y$ is maximized. Alternatively, finding such features subset achieves the lowest possible bound to the error of a classifier.

Parzen window approach is often used to estimate the mutual information [31]. However, it is hard and unreliable to evaluate for high dimensional data with limited number of samples. To overcome this problem, the marginal probability and marginal diversity at each feature dimension is estimated in [57]. The feature selection is based on the maximal marginal diversity criterion. This criterion is optimal for one dimensional features, and optimal for multiple dimensions if the mutual information between features is not affected by the knowledge of the class label. This assumption may not be hold for many feature representations. Vasconcelos further decomposed $F$ into several mutually exclusive feature subset and used same criterion to do feature selection [58].

Based on the performance of each feature with respect to the classification, the features can be divided into three categories: strong relevance, weak and relevance and irrelevance. Using this category information, one explicit feature redundancy analysis was proposed in [63] based on the feature’s Markov blanket [28]. Koller et.al [28] obtained an optimal subset by a backward elimination procedure, so called Markov blanket filtering. Let $G$ be the current set of features, a feature is redundant and hence should be removed from $G$ if and only if it is weakly revenant and has a Markov blanket $M_i$ within $G$ [63]. However measuring the conditional dependency for high dimensional data points is much hard in real practice.

A good generalization can be guaranteed if many sample points have small strangeness (see Section 3). In stead of estimating mutual information, we introduce an evaluation function which assigns a score to sets of features according to the strangeness they induce. Similar to RELIEF algorithm [49], we first introduce a weight vector $\omega \in \mathbb{R}^d$ over the feature set $F$. Then the strangeness can be re-formulated as the function of the weight vector $\omega$:

$$
\alpha^\omega_j = \frac{\sum_{l=1}^{k} d(\omega)_{jl}^c}{\sum_{l=1}^{k} d(\omega)_{jl}^c},
$$

or

$$
\alpha^\omega_j = \frac{\sum_{l=1}^{k} d(\omega)_{jl}^c}{\min_{n,n \neq c} \sum_{l=1}^{k} d(\omega)_{jl}^n},
$$

where $d(\omega)_{jl} = ||F_j - F_i||_\omega = \sqrt{\sum_k (F_{jk} - F_{ik})^2\omega_k}$. Defining the evaluation function as $\Phi(\omega) = \sum_{F_i \in F} \alpha^\omega_j$, the feature variable selection algorithm is shown in Algorithm 4.1. In the algorithm, $\varphi_1(\omega) = \sum_{l=1}^{k} d(\omega)_{jl}^c$, $\varphi_2(\omega) = \sum_{l=1}^{k} d(\omega)_{jl}^c$ or $\varphi_2(\omega) = \min_{n,n \neq c} \sum_{l=1}^{k} d(\omega)_{jl}^n$, $\xi_1(\omega)\omega_i = \frac{\partial \varphi_1(\omega)}{\partial \omega_i}$, $\zeta_2(\omega)\omega_i = \frac{\partial \varphi_2(\omega)}{\partial \omega_i}$, and $\rho$ is
the learning rate. It is a stochastic gradient descend procedure and very similar to the standard Relief [49]. We plan to compare the performance of our approach with Relief algorithm for feature selection in multi-class classification problems.

**Algorithm 4.1 Iterative Strangeness Feature Selection**

1. Initialize \( \omega = (1, \cdots, 1) \).
2. For \( t = 1, \cdots, T \)
   - Pick randomly an instance \( F_j \) from training data \( F \).
   - Calculate the strangeness of \( F_j \) with respect to \( \omega \).
   - For \( i = 1, \cdots, d \), calculate
     \[
     \Delta_i = \frac{\zeta_1(\omega)\varphi_2(\omega) - \zeta_2(\omega)\varphi_1(\omega)}{\varphi_2^2(\omega)}\omega_i
     \]
   - Update \( \omega = \omega - \rho \Delta \)
3. The selected feature set is \( \{ i | \omega_i > \tau \} \), where \( \tau \) is a threshold.

**4.2.2 Feature Instance Selection**

In order to deal with large variation of object appearance, due to occlusions, pose varieties, deformation, and size, many appearance-based approaches to object recognition characterize the objects by the local image features (patches) [59, 22, 45, 18, 13]. Each image is represented by \( M_i \) features \( \{ g_j \} \) in \( d \) dimensional space. The image patches are extracted by pixel-wise filtering [59, 45, 18] or by feature detectors [22, 13]. To improve the performance of the final classifier, several methods do feature instance selection on those local features. Viola and Jones [59] use an AdaBoost trained classifier to select the discriminative rectangular local features. Mahamud and Herbert [36] find discriminative object parts and develop an optimal distance measure for nearest neighbor search. Dorko and Schmid [10] develop one classifier for each object part, which is obtained by clustering the descriptor of local features. The object parts are selected according to the performance of the classifiers on the validation data set using the likelihood ratio and mutual information. Similar architecture has been applied for people detection [43]. However, in those works, the training images are pre-segmented. Many irreverent background pixels are eliminated by hand. The works in [59, 43] even need normalization.

Recently several object recognition approaches are developed on weakly supervised training data set, where the training images are given with positive objects as well as the cluttered background. In such a scenario, only the label of the whole image is given while the location of the object in the image is unknown. The goal is to obtain object representation in terms of parts without further segmentation. For weakly supervised object recognition, each image has different number local features detected as well as the large percentage of irrelevant features due to the cluttered background. At the same time, the correspondence of local features is unknown between the images of the same classes. Because of the repeatability of the feature
detectors, the same local feature cannot guarantee to be detected in all images of the same classes. As a result, most filter feature selection approaches can not work appropriately in such case. Wrapper approaches with certain classifiers are often used, such as Support Vector Machines and Gaussian Mixture Model in [10], Bayesian framework in [13, 12, 22] and linear Perceptron classifier in [55]. Those approaches either need pre-segmented training data or complex learning method to construct the model. Because of computational complexity of learning, the Bayesian framework approach in [13, 12, 22] only works with the small number of parts ($P \leq 10$), which can distinguish the object from the background but has not enough discriminative power to separate one object from other objects.

Clustering is another popular approach to deal with irrelevant features from cluttered background. All features in training data set are clustered in certain number of clusters and the each training image is represented by the histogram of the cluster centers. With such representation, many standard feature selection algorithm can be directly applied. Bishop [55] uses automatic relevance determination to discard the clusters belonging to the background. However, the number of clusters and clustering algorithm have great influence on the performance and generalization ability of the final classifier. Since the features from the background are assumed to be distributed uniformly in the descriptor space, large number of irrelevant features may have large number of clusters and overwhelm the relevant features in the clustering algorithm. In the end, totally different features coming from different objects may share the same cluster. In order to deal with the cluttered background, we propose a simple and efficient algorithm to discard the irrelevant features and select the discriminative features for later learning stage, as shown in Algorithm 4.2. The algorithm is based on the strangeness measure to evaluate the relevance between each local feature and the class label of whole image.

**Algorithm 4.2 Strangeness Feature Instance Selection**

1. Given local features $\{g_i\}$ in $\mathbb{R}^d$ and class label.
2. Compute the strangeness of each feature $g_i$ based on Equation 2.
3. Initialize the threshold of strangeness $\gamma$.
4. for $t = 1, 2, ..., T$
   - Select the features $\{g_k\}$ with the strangeness $\alpha_k \geq \gamma$.
   - Discard $\{g_k\}$ and update the strangeness of remaining features.
   - If the strangeness of all features is less than $\gamma$, terminate.
5. end for

Strangeness Instance Feature Selection algorithm 4.2 is an iterative backward elimination method. The algorithm repeatedly iterates over the feature set and updates the set of chosen features. There is one threshold in the algorithm $\gamma$, which determines the features to be eliminated in each iteration and controls the largest strangeness, that is, the minimal margin, of the chosen features in the end. In each iteration, the strangeness of each feature is recomputed if necessary. The algorithm can be applied very efficiently if suitable data structure
are used, because only small portion of strangeness need updating in each iteration. Compared with other feature selection algorithms, Algorithm 4.2 not only has the advantage of filter approaches – evaluating the relevance of feature and simple, but also have the properties of wrapper approaches – related to the predictor generalization performance.

4.3 Experiments and Evaluation

4.3.1 Variable Selection

In this section, we plan to demonstrate the behavior and performance of Iterative Strangeness Feature Selection (Algorithm 4.1) on different data set.

4.3.2 Feature Instance Selection

In this section, we demonstrate the behavior and performance of Strangeness Instance Feature Selection (Algorithm 4.2) on a small synthetic two-class classification problem. Then we test it on a task of feature selection for discriminating between objects and cluttered background and different locations.

Consider two classes with different kinds of features sampled from different distributions. As shown in Fig. 10(a), the first class has two kinds of features sampled from two distributions: Gaussian distribution $D_1$ with mean $[0, 0]^T$ and standard deviation $\sigma = 2$, and uniform distribution $D_0$ over region $(3.5, 8.5) \times (-8.5, -3.5)$; the second class also has two kinds of features sampled from two distributions: Gaussian distribution $D_2$ with mean $[3, 5]^T$ and standard deviation $\sigma = 2$, and uniform distribution $D_0$ over region $(3.5, 8.5) \times (-8.5, -3.5)$. Two different classes have the features sampled from the same distribution $D_0$, which is very common in weakly supervised object recognition task. If no feature selection is applied, it is very hard to find a learning algorithm to get the optimal decision boundary and achieve high classification performance. For each distribution in each class, 300 points are randomly sampled as the training data set. Fig. 10(b) shows the selected features after Algorithm 4.2 is applied. As we can see from the figures, most informative feature points are kept and most features with low discriminative ability are discarded. Only very small number features is chosen from $D_0$. Good decision and high performance can be now achieved if the learning algorithm is applied on the selected features.

![Figure 10: The features in training data set and the results after feature selection.](image-url)
Now we apply Algorithm 4.2 on efficient location recognition for robot navigation task. The video sequence is partitioned into 18 different locations, each of which is represented by different number of views. Each individual view is represented by scale-invariant (SIFT) keypoints described in [35]. More detail information can be found in [29, 32]. We will show the performance of Algorithm 4.2 on this data set. We also show that the feature selection is effective and efficient.

![Original features detected and the selected feature set.](image)

Algorithm 4.2 is further applied on weakly supervised object category recognition using Caltech data base. The more detail information about the data base can be found in [12]. Fig. 11 shows the original features detected and the chosen features by Algorithm 4.2. Obviously, most selected features are on the objects while most background features are discarded. The feature selection is very effective. More results on more classes will be presented.

4.4 Proposed work

We have proposed the feature variable selection and instance selection algorithms and shown the preliminary results. These algorithms remove the irrelevant features while keep the redundant ones. We plan to investigate the algorithms in more details:

- Apply variable feature selection algorithm on synthetic data and NIPS 2003 data set, and study the effect on overall recognition rate.
- Study in more detail the difference between proposed variable selection and Relief algorithm.
- Carry out the experiments on category recognition data set and evaluate the performance.
• Prove the convergence and the algorithms and study the connection between proposed feature instance algorithm and Markov Blanket.

5. Part-based Generic Object Recognition using Strangeness and Boosting

After the feature selection described in Section 4, most local features in training images now have strong relevance with respect to the classification and the complexity of the task are highly reduced. In this section, we address the problem of modelling and recognizing the objects from the selected local features with Boosting learning. The key insight here is to model the local classification evidence from each feature and integrate them.

Since the feature selection algorithms proposed in Section 4 only discard the irrelevant features, the redundant features survive in the data base. In order to achieve high generalization ability of final classifier, we first reduce the information redundancy among local features by clustering them into parts, and then model the local classification evidence by a model-free, non-parametric approach. It is the strangeness of each feature instance in each part introduced in Section 3. Finally we integrate the local evidence by Boosting learning to select the most discriminant and reliable parts. We plan to demonstrate the learning approach with generic object recognition with several different data sets, each of which has different learning complexity.

5.1 Related Work

Taking a closer look at the extensive body of literature on generic object recognition, many approaches have been proposed to integrate the local evidence and build the final strong classification rules. In this section, we will discuss some of the most relevant and most recent results related to our approach and point out the differences to our method.

Boosting was successfully used by Viola and Jones [59] as the learning ingredient for a fast face and pedestrian detector. The weak classifier was the thresholded average brightness of collections of up to four rectangular regions. The training images were pre-segmented and the local features were obtained by pixel-by-pixel scanning with different scale and orientation. In our approach, we extract the features by a feature detector and experiment with much larger sets of features to be able to perform recognition of a wider class of objects on more complicated images. Schneiderman and Kanade [51] also use Boosting to improve an already complex classifier and recognize objects from different viewpoints. Contrary to them, we are using Boosting to combine rather simple, non-parametric classifiers. As proved in [17], the generalization capability of the final strong classifier is highly related to the number of weak rules used and the complexity of the weak learner characterized by its VC-dimension. Simpler weak learner can guarantee better generalization power while complex weak learner will lead to over-fitting. Since each local feature is represented in high dimensional space, simple stump function cannot be applied directly. It is also difficult to optimize the KL divergence as shown in [34] due to the small number training images.

Also a wide variety of other learning techniques has been used to solve the task of generic object recognition. Agarwal and Roth [1] use Winnow as the underlying learning algorithm for the recognition of cars from
side views. In Winnow, one weak classifier needs to be calculated for each possible threshold a priori. War- 
vaven et al. [61] use Support Vector Machines combined with local features for object recognition. Fergus et al. [13] use a generative probabilistic model for objects built as constellations of parts. Using an EM-type learning algorithm, a good recognition performance is achieved. The parts and their constellations can be learned from cluttered images without supervision. We use a model-free approach and propose Boosting as a very different learning algorithm. Compared with EM, we have no limitation of the number of parts in the learning model. Ferrari et al. [14] present an approach where object recognition works even if aggravating factors like background clutter, scale variations or occlusion are very strong. Starting with a small initial set of corresponding features good results are obtained through iterative search. While this work presents a powerful concept of an iterative “active exploration” approach, it is based on a model for a specific object which is learned from non-cluttered representations of the object. Another object recognition approach was introduced by Dorko and Schmid [10]. It is based on the construction and selection of scale-invariant object parts. One classifier is constructed for each object part. The final strong classifier is integrated through the performance of those classifiers on the validation data set using the likelihood ratio and mutual information.

5.2 Boosting Learning

Our learning model is based on the AdaBoost algorithm [17]. The training labelled images are provided as 
$(I_1, y_1), \ldots, (I_n, y_n)$ for the learning algorithm, where $y_i = 1$ if $I_i$ contains a relevant object and $y_i = 0$ if $I_i$ contains no relevant object. The AdaBoost algorithm (see Algorithm 5.1) was adapted by adding the possibility of putting different weights on positive and negative training images, and delivers a classifier $h(I) : I \rightarrow y$ to predict whether a given image contains the object or not.

Algorithm 5.1 Standard AdaBoost Algorithm

- **Given training images** $(I_1, y_1), \ldots, (I_n, y_n)$, where $y_i = 0, 1$ for negative and positive training images respectively.
- **Initialize weight** $w_{1,i} = \frac{1}{m}, \frac{1}{l}$ for $y_i = 0, 1$ respectively, where $m$ and $l$ are the number of negatives and positives respectively.
- **For** $t = 1, \ldots, T$
  1. **Normalizing the weights**, $w_{t,i} \leftarrow \frac{\sum_j w_{t,j}}{w_{t,i}}$, so that $w_{t,i}$ is a probability distribution.
  2. **Training each weak classifier** $h_j(I)$ in the whole weak classifier space and evaluate the error with respect to the weights $w_{t,i}$, $e_j = \sum_i w_{t,i} |h_j(I_i) - y_i|$.
  3. **Choose the weak classifier** $h_t$ with the lowest error $\epsilon_t$. If $\epsilon_t \geq 0.5$, set $T = t - 1$ and terminate the loop.
  4. **Set** $\gamma_t = \epsilon_t/(1 - \epsilon_t)$.
  5. **Update the weight** $w_{t+1,i} = w_{t,i} \gamma_t^{1-e_i}$, where $e_i = 0$ if image $I_i$ is classified correctly, otherwise $e_i = 1$. 19
6. Set $\beta_t = -\log \gamma_t$.

- The final classifier is:
  
  $$h(I) = \begin{cases} 
  1, & \sum_{t=1}^{T} \beta_t h_t(I) \geq \frac{1}{2} \sum_{t=1}^{T} \beta_t \\
  0, & \text{otherwise} 
  \end{cases}$$

In AdaBoost algorithm, the weights $w_i$ are put on each training image and the weak learner $h_t$ is constructed which has some discriminative power with respect to the weight $w_i$ distribution, that is,

$$\sum_{k=1, h_t(I_k) = y_k}^{n} w_k > \sum_{k=1, h_t(I_k) \neq y_k}^{n} w_k.$$  \hfill (7)

The best weak learner $h_t$ is not expected to classify the training data well. It should be however better than random chance. In order to boost the weak learner, a sequential learning problem is solved. At each round of learning, the examples are re-weighted according the classification error of the previous weak classifier. The weight $w_k$ is decreased if the prediction for $I_k$ was correct ($h_t(I_k) = y_k$) and increased if the prediction was incorrect. Only the difficult examples, which were incorrectly classified by the previous weak classifier, are emphasized in the current round of learning. The final strong classifier takes the linear combination of weak classifiers followed by a threshold (see more detail in Algorithm 5.1).

Freund and Schapire proved that the training error of the strong classifier approaches zero exponentially in number of rounds [17]. The higher discriminative power the weak learner has, the faster the AdaBoost algorithm converges. In addition, the adaptation of the weights $w_k$ in each round performs some sort of adaptive de-correlation of the weak hypotheses. More importantly a number of results were proved about generalization performance of AdaBoost [48]. With simple and good weak hypotheses, AdaBoost achieves large margins rapidly and low generalization error. For multi-class problem, it can be either reduced to two-class problems using error-correcting codes or treated more directly using multi-class base classifiers. Freund and Schapire [17] proposed two algorithms AdaBoost.M1 and AdaBoost.M2 which are the straightforward generalization of AdaBoost using multi-class base classifiers. A joint multi-class Boosting algorithm was presented in [4] to find the sharing feature among the classes. There are some other multi-class Boosting approaches [11].

Obviously AdaBoost is a very general learning technique for obtaining classification functions. To adapt it for a specific application and obtain good generalization performance, suitable and relatively simple weak learner have to be constructed. For the purpose of multi-class object recognition, we need use the high dimensional local feature descriptors to construct the multi-class weak classifier. Simple “stump” functions or decision trees cannot be applied directly. As shown in Section 3, the strangeness is the non-parametric measurement capturing the discriminative ability of the features and is applicable for both two-class and multi-class problems. In next section, we will show how the strangeness is used to construct the weak learner in standard Boosting procedure.
5.3 Non-parametric weak learner

After feature selection, each training \( I_k \) image is represented by the selected feature set \( \{ g_k^j \} \). Based on the definition of strangeness in Equation 2, the strangeness of each feature is computed. Considering strangeness as the base classifier, we can apply the AdaBoost algorithm on the selected feature set directly. However, several features may be extracted from the almost same location of the same object thus they have the redundancy information. If each feature is considered as a weak classifier as in [47], the final strong classifier will be over-fitting and have the low generalization capability. For example, eye is a very important feature to distinguish face from other objects. If the final strong classifier has several “eye” weak learners associated with large coefficient \( \beta_t \), it has high probability of misclassifying the test face if the “eye” feature is not detected in the image.

To reduce the redundant information among features, we apply \( K \)-means clustering algorithm on the selected feature set of each class to group them into \( P \) parts. Each part is now considered as a weak classification rule which is used as the base classifier in AdaBoost. Fig. 12 shows the parts of the face and motorcycle categories after clustering. There are two kinds of benefits for clustering. At first, it reduces the redundant information among features thus there is almost no redundant information among weak hypotheses, which improves the generalization performance of the final strong classifier. Second, it constrains the maximal number of the weak rules in the final linear combination, thus the total computational complexity of the learning algorithm. Based on recent research on Boosting learning [37], AdaBoost will be over-fitting if the number of weak learners is too large. Constraining the part number \( P \) in certain range will reduce the chance of over-fitting. Since most features are now relevant and most background features are discarded, the clustering algorithm and the number of clusters do not have much influence on the performance of the learning algorithm. As shown in Section 5.5, the performance of the final classifier has close performance when \( P \) is in some range.

Now in training image set, each object is represented by \( P \) parts, each of which has \( N_i \) feature instance \( \{ g_j^i \}_{j=1}^{N_i} \). Without parametric modelling, we keep those clusters and feature instance as the training gallery and learn the coefficients and thresholds of the weak learner through validation data set. Given the validation image \( V_i \) and its local features descriptor \( \{ g(V_i)_j \} \) with putative object label \( c \), the matched features \( \{ \tilde{g}(V_i)_j \}_{j=1}^{P} \) are found which are the closest feature from \( \{ g(V_i)_j \} \) to the parts \( \{ g_j^i \}_{j=1}^{N_i} \) of class \( c \) in the
gallery. Then the strangeness \( \{\alpha_c^c\} \) of \( \{\tilde{g}(V_i)_j\}_{j=1}^P \) is computed with the assumption of putative class label \( c \). With \( C \) classes in the training gallery, \( C \) groups of strangeness are obtained for each validation image. If \( M \) validation images are given for each class, for each part of each class, we have \( M \) positive strangeness measures and \( M(C - 1) \) negative ones. Our weak hypothesis is to select the matched feature \( \{\tilde{g}(V_i)_j\}_{j=1}^P \) and the strangeness threshold \( T_j \) for each part of the class. The algorithm of weak learner is shown in Algorithm 5.2.

**Algorithm 5.2 Strangeness Weak Learner**

- **Input:** Training gallery \( \{G^c_j\}_{j=1}^P \), \( c = 1, \cdots, C \), where \( G^c_j \) is the feature instance set of \( j \)th part of class \( c \), and validation images \( \{V_i, i = 1, \cdots, MC\} \) and associated feature \( \{g(V_i)_k\} \).
- **Distance metric function:** Let \( d(\bullet, \bullet) \) be the distance between the descriptor of the features.
- **Strangeness computation:** For each part \( j \) of class \( c \), find the nearest feature \( \tilde{g}(V_i)_j \) between \( \{g(V_i)_k\} \) and \( G^c_j \). The strangeness of \( \tilde{g}(V_i)_j \) is then computed as defined in equation 2 under the assumption that \( V_i \) has the class label \( c \). Each part of class \( c \) now has \( MC \) strangeness \( \{\alpha^c_k\}_{k=1}^{MC} \), \( M \) of which are positive and \( M(C - 1) \) are negative.
- **Strangeness sorting:** For each part \( j \) of class \( c \), let \( \pi(1), \cdots, \pi(MC) \) be the permutation such that 
  \[
  \alpha^c_{\pi(1)} \leq \alpha^c_{\pi(2)} \leq \cdots \leq \alpha^c_{\pi(MC)}.
  \]
- **Select the threshold of weak learner:** For each part \( j \) of class \( c \), find the best position \( s \) such that the maximal classification rate is achieved:
  \[
  \text{rate}(j) = \max_s \sum_{k=1}^s w_{\pi(k)} 1(\alpha_{\pi(k)})
  \]
  where \( 1(\alpha_{\pi(k)}) \) is 1 if \( \alpha_{\pi(k)} \) is positive and 0 otherwise. Then the threshold of current weak learner is:
  \[
  \theta(j) = \frac{\alpha_{\pi(s)} + \alpha_{\pi(s+1)}}{2}.
  \]
- **Select best weak learner:** Find the best part \( m = \max_j \text{rate}(j) \). Then the best weak learner of current round is the \( m \)th part with the best threshold \( T_m = \theta(m) \). Update the weight \( w_k \) according to error \( 1 - \text{rate}(m) \).

The Strangeness Weak Learner is model-free and non-parametric and as simple as the stump function. The main computational burden is the calculation of strangeness of \( g(V_i)_j \) with putative label \( c \), since it needs the distances from \( g(V_i)_j \) to all features in the training gallery. However, such computation can be done prior to Boosting and weak learner finder. The remaining calculations in Boosting are very inexpensive. This weak learner can be easily extended if the image is represented by multiple types of features. In such case, the object has multiple representations with different number of parts. The strangeness are computed for
each type of feature descriptor with the corresponding distance metric function. The weak learner gets the
performance and threshold for each type of feature and the best weak learner is found across the parts and
feature types.

5.4 Learning Model and Recognition

Based on the weak learner in Algorithm 5.2, the standard AdaBoost algorithm (see Algorithm 5.1) is applied
and the learning model diagram is shown in Fig 13. In this learning model, each round of Boosting is close
to the hypothesis test, similar to Transductive inference [60, 50]. In each iteration, the hypothesis is made
first (putative label of the image) and the strangeness is calculated under such hypothesis. The final output
of Boosting is used to test if such hypothesis is valid or not. Drawing an analogy between weak classifiers
and features, this learning model is another aggressive feature selection mechanism for selecting a small set
of “good” features which nevertheless have significant variety. Although each class is represented by \( M \)
parts, only part of them are used after Boosting learning. Some parts may have no discrimination power with
respect to the weight \( w \) and have the zero coefficients in the final linear combination. If the coefficients \( \beta_i \)
is normalized as \( \sum_i \beta_i = 1 \), the output of AdaBoost can be interpreted as the approximation the posterior.

![Figure 13: Strangeness based Boosting Learning.](image)

Recognizing query image \( Q \) is straightforward, similar to validation stage and shown in Fig. 14. With
putative label \( c \) for \( Q \), the strangeness is computed, compared with the thresholds and the output of Boosting
is used as the estimation of the posterior \( P(c|Q) \). Compared with most parametric approaches, the computa-
tional complexity of testing stage is relative expensive since we need compute the distance between the
feature \( g(Q)_j \) and all features in the training gallery. However, out learning model is non-parametric, without
complex parameter estimation and model space searching. It is easily extended for online and incremental
learning. We plan to propose the incremental strangeness based Boosting learning.

There are two parameters in our learning model which need to be tuned: the number of nearest neighbor
\( k \) in strangeness measure and the number of parts \( P \). \( k \) can be estimated from the number of training images
for each class and the repeatability of the feature detector. We can vary \( k \) and find the best one with best
performance on the validation data set. We also can vary \( P \) and find the best one through validation. As
demonstrated in Section 5.5, the performance of the final classifier is very stable and has close performance
when $P$ is in some range.

5.5 Experiments, Evaluation and Comparisons

In this section we present numerical evaluation of our described method in the application areas of specific object recognition and object category recognition. The training image data is either weakly supervised or pre-segmented. The computational complexity of our learning model is analyzed and the performance is evaluated and compared with the results of other groups.

5.5.1 Location Recognition

We first plan to test our method on location recognition task. In this task, the video sequence is partitioned into 18 different locations, each of which is represented by different number of views. Each individual view is represented by scale-invariant (SIFT) keypoints described in [35]. More detailed information about the data and representation can be found in [29, 32]. Fig 15 shows the representative views for some location, which demonstrate the variability of our data set.

![Figure 15: Examples of representative views of 12 out of 18 locations.](image)

We plan to compare our method with the voting approach (the method in [29, 32]) to demonstrate our
feature selection is effective. The output of Boosting learning can be further as the input of the Hidden Markov Model when the spatial relationship between location during navigation are considered to resolve misclassification due to the self-similarity and dynamic changes in the environment.

5.5.2 Weakly Supervised Object Category Recognition

In this section, we plan to demonstrate our method with weakly supervised object category recognition using object category data set [13, 12]. Some sample images are shown in Fig. 17.

![Figure 16: Hierarchical Boosting Classification.](image)

In stead of just discriminating the object category from background as did in [13, 12, 10], we propose a two-stage hierarchical learning to distinguish the object from both background and other objects. At first, a two-class Boosting learner is used to classify object category from background. Based on the features selected in the first stage, further feature selection is processed and another one-vs-all Boosting learner is used to classify different object categories. The diagram is shown in Fig. 16. In this stage, the label of image $Q$ is predicted by $\max P(c|Q)$, where $P(c|Q)$ is the output of the final classifier with the input $Q$. It is necessary to use those two stages. The background features are uniformly distributed in feature descriptor space hence we cannot model the background with parts. Thus we cannot reliably estimate $P(\text{background}|Q)$. Given the estimated $P(c|Q)$, it is very hard and almost impossible to find a threshold $\tau$ such that $Q$ is background if $\max P(c|Q) \leq \tau$. To avoid estimating such threshold, the Boosting learning is proposed to deal with the background. The high performance can be achieved in this stage since more features will be used. Some of them have no discriminative power to classifier object categories but have the ability to distinguish the object from background.

We further plan to evaluate the online (incremental) version of our method on those database and compare the results with others.

5.5.3 MIT database

We further plan to evaluate our method on the pre-segmented object data base [52]. There is large number of classes in this database, where the test images have large portion of background clutter. The main purpose of this experiments is to evaluate how our learning algorithm deals with the background during the testing.
5.6 Discussion and proposed work

The main contribution of this section is a general non-parametric strangeness-based Boosting learning approach, which can be used for both supervised object and weakly supervised object category recognition. For the weakly supervised category recognition, same method is used to discriminate the object from the background and other objects. The performances over different data set and learning show that the algorithms are effective and efficient.

we plan to complete the implementation of the algorithm and all the experiments by the final defense.

There are several advantages over our approach: no complex parametric model assumption and parameter estimation; good scalability, suitable for online learning and the limitation of the feature detector’s repeatability is compensated by Boosting learning stage. The limitation are: the assumption that the object is represented by parts may be invalid; for large number of classes and training image, need efficient algorithm to compute k-nearest neighbors; and the performance depends on the feature detectors

6. Discussions

We have presented a new feature selection criterion and a non-parametric Boosting learning approach and applied them to specific object and object category recognition. We have shown its promise on different large image data sets.
6.1 Expected Contributions

This proposal will expect to make the following contributions:

- A new feature selection criterion, which has the properties of both filter and wrapper approach. It works even the training sample size is small.
- The Boosting learning framework with a new weak learner, which can be used for both specific object recognition and general object category recognition. It can be easily extended for incremental learning and integrate the evidence from other type feature representation.
- Simple and efficiency, theoretical analysis of our learning approach.

6.2 Remaining work

The major pieces the remaining work to do by the defense are:

- Theoretical analysis of the strangeness as a classification rule. We will use the Lipschitz function space and Rademacher complexities [6] to prove its generalization bound, which guarantee the generalization ability of our learning framework.
- Feature selection experiment and evaluation on different data bases.
- Learning and classification experiment on different data bases.
- Theoretical and computational analysis of our learning framework.

6.3 Future work

The approach proposed in this work is based on appearance of the local features only. The spatial relationship between features are not considered here, which are modelled in the work [13, 12]. We hope the geometric information will be added in the Boosting framework as another week evidence and integrate the geometric and appearance information for learning task. To move forward from here, we need to learn how to statistically represent the geometric information and build a metric distance function and another “strangeness” measure on it. One possible solution is to conceive of object as graph. The node of graph is the appearance of local feature and its relative spatial location. The “strangeness” is computed through local subgraph.

References


28


