Submodular Asymmetric Feature Selection in Cascade Object Detection

Baosheng Yu[†] and **Meng Fang**[‡] and **Dacheng Tao**[†] and **Jie Yin**[§]
[†]Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney
[‡]Department of Computing and Information Systems, The University of Melbourne
[§]CSIRO, Australia
baosheng.yu@student.uts.edu.au, meng.fang@unimelb.edu.au
dacheng.tao@uts.edu.au, jie.yin@csiro.au

Abstract

A cascade classifier has turned out to be effective in sliding-window based real-time object detection. In a cascade classifier, node learning is the key process, which includes feature selection and classifier design. Previous algorithms fail to effectively tackle the asymmetry and intersection problems existing in cascade classification, thereby limiting the performance of object detection. In this paper, we improve current feature selection algorithm by addressing both asymmetry and intersection problems. We formulate asymmetric feature selection as a submodular function maximization problem. We then propose a new algorithm SAFS with formal performance guarantee to solve this problem. We use face detection as a case study and perform experiments on two real-world face detection datasets. The experimental results demonstrate that our algorithm SAFS outperforms the state-of-art feature selection algorithms in cascade object detection, such as FFS and LACBoost.

Introduction

Object detection deals with detecting semantic objects (such as persons, buildings, or cars) in digital images and videos. It is a fundamental problem in computer vision, which is often formulated as a search and classification problem: the search strategy generates potential image regions and the classifier determines whether or not the potential image region contains an object. Most of the state-of-art object detection systems are based on a sliding-window strategy (Viola and Jones 2001b; Wu et al. 2008; Benenson et al. 2014), which involves scanning an image with a rectangular window and determining the presence of an object through a binary classifier. However, this brute-force search strategy is computationally expensive (e.g., in a 360×360 image, there are more than 10000 rectangular windows needed to be classified). As a result, there are two main methods to improve the object detection process: cascade classifier (Viola and Jones 2001b) and efficient window search (Lampert, Blaschko, and Hofmann 2008). These two methods can work collaboratively.

In sliding-window based object detection setting, there are only a few windows containing objects, which is known as the rare event detection problem (Wu, Rehg, and Mullin 2003). The cascade explicitly uses a cascade of node classifiers with increasing complexity. Each node classifier is used to correctly reject a portion of non-object windows (measured by false positive rate) and retain any possible object windows for further precessing (measured by detection rate), as shown in Figure 1. As most of the non-object windows can be quickly rejected by simple node classifiers at early stages, the cascade runs fast and has high accuracy. The state-of-art node learning framework (Wu et al. 2008) contains two parts: feature selection and node classifier design. It has been demonstrated that selecting features and forming an ensemble classifier using linear combinations of these features is effective for building node classifiers.

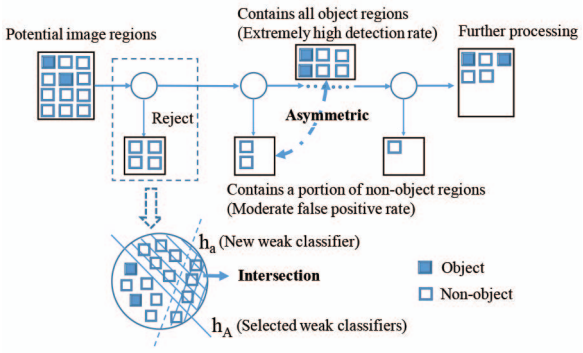

Figure 1: Cascade object detection with asymmetry and intersection problems. The asymmetry problem: maintain an extremely high detection rate and a moderate false positive rate. The intersection problem: reduce the redundant voting.

One of main considerations in feature selection is about asymmetry, which is illustrated in Figure 1. In order to guarantee the accuracy of the cascade (high detection rate and low false positive rate), each node classifier needs to be an asymmetric classifier, which has an extremely high detection rate (e.g., 0.99) and a moderate false positive rate (e.g., 0.5). Wu et al. (2008) proposed a fast feature selection algorithm, called the Forward Feature Selection (FFS) algorithm. However, FFS algorithm uses error rate as criterion for feature

selection without considering the asymmetry problem. Shen et al. (2013) also proposed a new feature selection algorithm LACBoost, which uses a weighted error rate as criterion for feature selection.

Another important consideration is about intersection, which is related to the redundancy of weak classifiers (Paisitkriangkrai, Shen, and van den Hengel 2014). Each node classifier in the cascade is built as an ensemble by selecting a set of weak classifiers. Each weak classifier h_i is associated with a single feature $i \in E = \{1, 2, \ldots, d\}$, and thus it is also called a single-feature classifier [1], which can only correctly classify a small portion of training examples. Therefore, features need to be carefully selected in order to guarantee the accuracy of an ensemble classifier. On the one hand, more selected weak classifiers are expected to reach a consensus towards classifying a given example, so that the ensemble classifier can make a correct decision. On the other hand, all the node classifiers should correctly classify as many training examples as possible to achieve high coverage. There is a trade-off between improving accuracy on each training example and improving the voting coverage on all training examples. In other words, voting on the training examples already correctly classified is redundant and useless. We give an intuitive example in Figure 1, where h_A indicates current selected weak classifiers $h_i (i \in A \subset E)$, $h_a (a \in E \setminus A)$ indicates a new weak classifier based on new chosen feature and the non-object items in the shaded region indicates negative examples correctly classified. The new weak classifier is useless to the examples in the intersection area (overlapping area of two shaded regions). Therefore, it would be desirable to select proper features that could reduce redundant votes among node classifiers. However, the intersection problem described above has not been properly addressed by previously proposed approaches.

In this paper, we propose a new feature selection algorithm called Submodular Asymmetric Feature Selection or SAFS that simultaneously considers the asymmetry and intersection problems. Submodularity is an intuitive notion of diminishing returns, which states that adding a redundant item to a set is useless. The intersection problem addressed in this paper can be well-captured by such a submodular function. Inspired by this, we first reformulate feature selection as a submodular maximization problem (Krause and Golovin 2012). We then solve it by a greedy algorithm and provide the performance guarantee. There are three main contributions of our work: 1) we explicitly address the intersection problem; 2) we introduce submoduarity to describe the intersection problem in asymmetric feature selection; 3) we proposed a new efficient algorithm to solve asymmetry and intersection problems simultaneously.

The remainder of this paper is organized as follows. We first briefly review the related works of cascade object detection and submodularity. We then describe the asymmetry and intersection problems respectively. After that, we formulate the feature selection process as a submodular function maximization problem. We then propose a new algorithm SAFS with performance guarantee to solve it. Lastly, we perform experiments on two real-world face detection datasets.

[1]Training method of the weak classifier shows in Wu et al. (2008)

Related Work

Cascade object detection framework was firstly proposed by Viola and Jones (2001b), which used AdaBoost to train a node classifier. An improved variant of AdaBoost called AsymBoost has been proposed by Viola and Jones(2001a) in order to solve the asymmetry problem. However, both these works need to retrain all weak classifiers at each iteration. As a result, they are computationally prohibitive and it takes several weeks to train the whole cascade (Viola and Jones 2004).

Recently, Wu et al. (2008) decouples the node learning process into two parts: feature selection and ensemble classifier design. There are several works focusing on the design of ensemble classifiers, which have been demonstrated to work well for object detection (Viola and Jones 2001a; Wu, Mullin, and Rehg 2005; Shen et al. 2013). As an important part of node learning, feature selection has been addressed by many studies (Wu, Rehg, and Mullin 2003; Wu et al. 2008; Shen et al. 2013). For example, Wu et al. (2008) proposed a fast feature selection algorithm called FFS, which greedily selects an optimal set of features so as to minimize error rate of the ensemble classifier. However, FFS algorithm does not take the asymmetry problem into consideration. Later, Shen et al. (2013) proposed an asymmetric feature selection algorithm called LACBoost, which is based on the biased minimax probability machine (Huang et al. 2004) and shares the same formulation with linear asymmetric classifier (Wu et al. 2008). The LACBoost is approximately formulated as a quadratic programming problem and entropic gradient is used to solve this problem. Similar to Adaboot, LACBoost algorithm has almost the same computation complexity because of solving the quadratic programming problem in each iteration. However, none of these works have well addressed the intersection problem. Here, our proposed algorithm considering both the asymmetry and intersection problems simultaneously.

Submodularity has turned out to be effective in solving a lot of NP-hard problem, such as the set cover problem (Iyer and Bilmes 2013), sensor placement (Krause, Singh, and Guestrin 2008) and diverse recommendation (Ashkan et al. 2014; Yue and Guestrin 2011). The intersection problem in feature selection is similar to a set cover problem, which can be well described by a submodular function. Submodularity has been used in many feature selection as well as subset selection problems (Krause et al. 2008; Das and Kempe 2011; Das, Dasgupta, and Kumar 2012; Liu et al. 2013). However, these submodularity based feature selection algorithms can not be used in cascade object detection setting, because none of them considers the asymmetry problem, which only appears in cascade object detection setting. Our work formulates the feature selection problem in cascade object detection as a submodular function maximization problem and propose an efficient algorithm to solve it with performance guarantee.

Problem Description

In this section, we first briefly introduce the cascade framework and the node learning process. We then describe the asymmetry and intersection problems in feature selection respectively. Lastly, we formulate the problem of asymmetric feature selection.

Node Learning

Let $X = \{(x_i, y_i)\}_{i=1}^m$ be a training dataset, where $x_i \in R^d$ is a feature vector and $y_i \in \{-1, 1\}$ is the label of x_i. Let $E = \{1, 2, \ldots, d\}$ denote a set of features, $P = \{(x_i, y_i)|(x_i, y_i) \in X, y_i = 1\}$ denote the set of all positive examples and $N = \{(x_i, y_i)|(x_i, y_i) \in X, y_i = -1\}$ denote the set of all negative examples. Let $\mathcal{H} = \{H_1, H_2, \ldots\}$ denote the cascade and $H \in \mathcal{H}$ denote a node classifier. We describe the cascade framework in Algorithm 1.

Algorithm 1 The cascade framework

Input: P, N and bootstrapping negative examples D.
Output: $\mathcal{H} = \{H_1, H_2, \ldots, H_K\}$ // the cascade.
1: **for** $r = 1, 2, \ldots, K$ **do**
2: -Training node classifier H_r by P, N. // node learning.
3: -$\mathcal{H} = \mathcal{H} \cup \{H_r\}$.
4: -Update N according to H_r and D. // bootstrapping.
5: **end for**

In Algorithm 1, node learning process is the key to construct a good cascade. The node learning process is decoupled into two parts: feature selection and ensemble classifier design. For fair comparison, we use the same voting method for the ensemble classifier (Wu et al. 2008). Let h_i denote the weak classifier corresponding to feature $i \in E$, which is actually a single-feature classifier. Let $A \subseteq E$ denote a set of selected features, then the ensemble classifier corresponding to feature set A can be constructed by the voting results of all weak classifiers $h_i (i \in A)$ and a threshold θ, i.e.,

$$H(x) = h_{A,\theta}(x) = \mathbf{sgn}\left(\sum_{i \in A} h_i(x) - \theta\right). \quad (1)$$

Asymmetry Problem

The goal of node learning, which is to achieve an extremely high detection rate and a moderate false positive rate, is asymmetric in detection rate and false positive rate. We demonstrate the relationship between error rate, false negative rate ($= 1 -$ detection rate) and false positive rate in Lemma 1.

Lemma 1. *Let FP denote the false positive rate, FN denote the false negative rate and ER denote the error rate. Then we have*

$$ER = \alpha * FP + (1 - \alpha) * FN,$$

where α is the proportion of negative examples in training set.

Proof. In Appendix. □

In Lemma 1, we know that error rate is a weighted sum of false negative rate and false positive rate. Previously, FFS algorithm uses error rate (or weighted error rate) as criterion for feature selection. However, it is not equal to the node learning goal. For the node learning goal, an extremely high detection rate is necessary while we can always increase the number of node classifiers to achieve a proper false positive rate. As a result, it is more reasonable to treat false negative rate as a constraint and then try to minimize false positive rate under this constraint.

Intersection Problem

Intersection problem appears in feature selection of node learning and we describe it as follows: in node learning framework, each feature is corresponding to a weak classifier and each weak classifier can only correctly vote to a special set of examples. That is, given weak classifiers h_i, h_j, which can correctly vote to $X_i, X_j \subseteq X$, i.e., $\forall (x, y) \in X_i, h_i(x) = y$ and $\forall (x, y) \in X_j, h_j(x) = y$. Both h_i and h_j will vote to all examples $(x, y) \in X_i \cap X_j$, thus there may exist redundant voting. For a new chosen weak classifier h_a, it will be no help to classifying examples that have been correctly classified by $h_{A,\theta}$. We describe this problem as intersection problem.

As described in asymmetry problem, we treat false negative rate as a constraint and then try to minimize false positive rate under this constraint. In other words, we try to maximize the number of negative examples correctly classified under a constraint. For the ensemble classifier, less intersection means a more evenly voting to training examples and reducing the redundant voting. As a result, the voting differences between negative examples and positive examples are maximum. We can then use a threshold θ to correctly classify all positive examples and as much negative examples as possible. We give an intuitive example in Figure 2, where Figure 2(b) is a better voting result. More negative examples are correctly classified in Figure 2(b) comparing to Figure 2(a) and it means lower false positive rate in Figure 2(b).

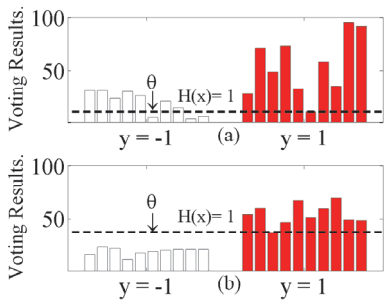

Figure 2: Example of intersection problem in feature selection.

Inspired by this, we try to select a set of features (or weak classifiers) that maximize the number of negative examples bellow the threshold θ. It is obvious that more negative examples below the threshold θ mean lower false positive rate under the same detection rate constraint. Lastly, the node

learning goal is equal to

$$\begin{aligned} \max_{A,\theta} \quad & Pr\{h_{A,\theta}(x) = -1 | y = -1\}, \\ \text{s.t.} \quad & Pr\{h_{A,\theta}(x) = 1 | y = 1\} \geq 1 - \beta, \\ & A \subseteq E, |A| \leq k. \end{aligned} \tag{2}$$

Submodular Asymmetric Feature Selection

In this section we first formulate the feature selection process as a submodular function maximization problem by considering both the asymmetry and intersection problems. We then propose a greedy algorithm, which is called Submodular Asymmetric Feature Selection or SAFS, to solve feature selection problem efficiently.

Submodular Maximization

As described before, we formulate the node learning goal as (2). However, it is a NP-hard problem, which means that the optimal solution can not be found in polynomial time. The intersection problem is actually same with a maximum coverage problem or a set cover problem, which can be well-captured by a submodular function (Iyer and Bilmes 2013). Inspired by this, we prove that (2) is equal to a submodular function maximization problem.

Definition 1. *(submodularity). Let E be a nonempty finite set and 2^E be a collection of all subsets of E. Let $f\colon 2^E \to R$ be a submodular function, i.e.,*

$$\forall X \subseteq Y \in 2^E \text{ and } a \in E \setminus Y, f(a|X) \geq f(a|Y), \tag{3}$$

where $f(a|X) = f(X \cup \{a\}) - f(X)$.

Definition 2. *(monotonity) Let E be a nonempty finite set and 2^E be a collection of all subsets of E. Let $f\colon 2^E \to R$ be a monotone function, i.e.,*

$$\forall X \subseteq Y \in 2^E, f(Y) \geq f(X). \tag{4}$$

Let $A \subseteq E$ denote a subset of features and $g_\theta(A) = Pr\{h_{A,\theta}(x) = -1 | y = 1\}$. Let $f_\theta(A)$ denote the true negative rate of a node classifier, i.e.,

$$f_\theta(A) = Pr\{h_{A,\theta}(x) = -1 | y = -1\}. \tag{5}$$

We then rewrite (2) as follows:

$$\begin{aligned} \max_{A,\theta} \quad & f_\theta(A), \\ \text{s.t.} \quad & A \subseteq E, |A| \leq k, g_\theta(A) \leq \beta. \end{aligned} \tag{6}$$

Considering that the node learning goal in (2) equals to maximizing the number of negative examples below of the threshold θ, we set θ as the maximum possible value under the constraint $g_\theta(A) \leq \beta$, i.e.,

$$\theta_A = \max_{\theta : g_\theta(A) \leq \beta} \theta. \tag{7}$$

Thus we have

$$f(A) = f_{\theta_A}(A) = Pr\{h_{A,\theta_A}(x) = -1 | y = -1\}. \tag{8}$$

We can then translate (6) as follows:

$$\max_{A : A \subseteq E} f(A), \text{ s.t. } |A| \leq k. \tag{9}$$

For the ensemble classifier used in node learning process, more weak classifiers will increase the accuracy of ensemble classifier and computation complexity at the same time. Considering the computation cost, we choose a subset of weak classifiers corresponding to feature set A ($A \subseteq E$ and $|A| \leq k$) with maximum value $f(A)$. Under the base assumption that more weak classifiers will increase the accuracy of ensemble classifier, we prove that $f(A)$ is a monotone submodular function in Theorem 1.

Theorem 1. *$f(A) = Pr\{h_{A,\theta_A}(x) = -1 | y = -1\}$ is a monotone submodular function under a base assumption that more weak classifiers will increase the accuracy of the ensemble classifier.*

Proof. Let $h_{A,\theta_A}(x), h_{B,\theta_B}(x)$ denote two ensemble classifiers corresponding to feature set $A \subset B \subseteq E$. Let $a \in E \backslash B$ denote a new feature added to the feature set A and B. We first prove that $f(A)$ is a monotone function and then prove that $f(A)$ is a submodular function.

Under the base assumption that more weak classifiers will increase the accuracy of the ensemble classifier, we have that: $\forall (x, y) \in X$ and $h_{A,\theta_A}(x) = y$, then $h_{B,\theta_B}(x) = y$ with high probability. Let $X_a = \{(x, y) | h_a(x) = -1, y = -1\}$, $X_A = \{(x, y) | h_{A,\theta_A}(x) = -1, y = -1\}$ and $X_B = \{(x, y) | h_{B,\theta_B}(x) = -1, y = -1\}$. We then have $X_A \subseteq X_B$. As a result,

$$\begin{aligned} & f(B) - f(A) \\ & = Pr\{h_{B,\theta_B}(x) = y | y = -1\} \\ & \quad - Pr\{h_{A,\theta_A}(x) = y | y = -1\} \\ & = Pr\{h_{A,\theta_A}(x) = 1, h_{B,\theta_B}(x) = -1 | y = -1\} \\ & = \frac{|X_B \setminus X_A|}{|P|} \geq 0. \end{aligned} \tag{10}$$

That is, $f(B) \geq f(A)$, i.e., $f(A)$ is a monotone function. Next, we prove $f(A)$ is a submodular function. Let $X_{a|A} = X_{A \cup \{a\}} \setminus X_A$ and $X_{a|B} = X_{B \cup \{a\}} \setminus X_B$. Then we have

$$\begin{aligned} & f(a|A) - f(a|B) \\ & = Pr\{h_{A,\theta_A}(x) = h_{A \cup \{a\}, \theta_{A \cup \{a\}}}(x) = -1 | y = -1\} \\ & \quad - Pr\{h_{B,\theta_B}(x) = h_{B \cup \{a\}, \theta_{B \cup \{a\}}}(x) = -1 | y = -1\} \\ & = \frac{|X_{a|A}| - |X_{a|B}|}{|P|} \\ & = \frac{|\,(X_B \setminus X_A) \cap X_a|}{|P|} \\ & \geq 0. \end{aligned} \tag{11}$$

That is, $f(a|A) \geq f(a|B)$, i.e., $f(A)$ is a submodular function. From (10) and (11), we have that $f(A)$ is a monotone submodular function. □

The feature selection problem in (9) is thus defined as a submodular function maximization problem, i.e.,

$$A^* \in \arg\max_{A : A \subseteq E} f(A), \text{s.t.}, |A| \leq k. \tag{12}$$

Algorithm

Inspired by Nemhauser, Wolsey, and Fisher (1978), we propose a new asymmetric feature selection algorithm, which is called Submodular Asymmetric Feature Selection or SAFS. In SAFS algorithm, we greedily choose each feature a, which maximally increases the value of $f(A \cup \{a\})$, i.e.,

$$a \in \arg\max_{a:a \in E \setminus A} f(A \cup \{a\}). \tag{13}$$

From (13) and the definition of $f(A)$, SAFS algorithm chooses the new feature according to two respects: 1) the new ensemble classifier $h_{A\cup\{a\},\theta_{A\cup\{a\}}}$ satisfies the constraint that detection rate is greater or equal to $1-\beta$; 2) the new ensemble classifier maximumly increases the number of negative examples correctly classified (i.e., maximumly reduces the false positive rate). The detail of SAFS algorithm is shown in Algorithm 2.

Algorithm 2 The SAFS algorithm

Input: X, E and $k(|A| \leq k)$.
Output: h_{A,θ_A} // the node classifier.

1: Make a table $V_{i,j} = h_i(x_j)$, for all $i \leq d$ and $j \leq m$.
2: **for** $t = 1, 2, \ldots, k$ **do**
3: $H(x) = \mathbf{sgn}\left(\sum_{i \in A\cup\{a\}} h_i(x) - \theta_{A\cup\{a\}}\right)$.
4: $f(A \cup \{a\}) = \sum_{i=1}^{m} \mathbf{1}\{H(x_i) = -1, y_i = -1\}$.
5: $a \in \arg\max_{a:a \in E\setminus A} f(A \cup \{a\})$.
6: $A = A \cup \{a\}$.
7: **end for**
8: $h_{A,\theta_A}(x) = \mathbf{sgn}\left(\sum_{i \in A} h_i(x) - \theta_A\right)$.

As (12) is still a NP-hard problem, we show a $\left(1-\frac{1}{e}\right)$-approximation bound for SAFS algorithm in Theorem 2.

Theorem 2. *Let $f : 2^E \to R_+$ be defined in (8) and A be the greedily selected set defined in (13). We then have*

$$f(A) \geq \left(1 - \frac{1}{e}\right) \max_{|A|\leq k} f(A).$$

Theorem 2 is simply derived from Nemhauser, Wolsey, and Fisher (1978). We can then guarantee that the worst case of our algorithm will not worse than $\left(1-\frac{1}{e}\right)$-approximation of the optimal solution and it is the best result for a polynomial-time algorithm (Nemhauser, Wolsey, and Fisher 1978). That is, if the optimal classifier (error rate = 0) uses k features and the feature set is A^*, then we can guarantee that $f(A) \geq \left(1-\frac{1}{e}\right) f(A^*) \approx 0.632$ (i.e., false positive rate $\leq 1 - 0.632 = 0.368$), where A is selected by our SAFS algorithm. Also, we provide an important extension of Theorem 2 according to Krause and Golovin (2012) in Theorem 3, which enhances the effectiveness of our algorithm.

Theorem 3. *Let $f : 2^E \to R_+$ be defined in (8) and A be the greedily selected set defined in (13). If the optimal set of features $|A^*| = k$, then for all positive integer l and $|A| = l$, we have*

$$f(A) \geq \left(1 - e^{-\frac{l}{k}}\right) \max_{|A|\leq k} f(A) = \left(1 - e^{-\frac{l}{k}}\right) f(A^*).$$

Proof. See proof of Theorem 1.5 in Krause and Golovin (2012). □

In Theorem 3, if the optimal set of features is $A^*(|A^*| \leq k)$, then we can always find a set $|A| = l \geq k$ by SAFS that makes $f(A)$ quickly converge to $f(A^*)$, which is illustrated in Figure 3.

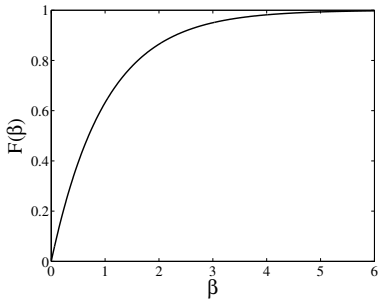

Figure 3: Let $\beta = l/k$ and $F(\beta) = f(A)/f(A^*)$, then $F(\beta)$ converges to 1 very quickly.

Experiment

In this section, we used face detection as a case of object detection and performed experiments on two real-world face detection dataset. The experimental results compare our SAFS algorithm with current state-of-art feature selection algorithms, such as FFS and LACBoost. Also, a random selection or RS algorithm is used as a baseline in our experiments.

Datasets

We used two real-world face detection datasets in our experiments.

- **CBCL Face Database #1** [2]: It consists of a training set of 6977 images (2429 face and 4548 non-face) and a test set of 24045 images (472 face and 23573 non-face) and all face images are aligned to a base resolution of 19×19 pixels.
- **Caltech 10,000 Web Faces** [3]: It contains images of people collected from the web through Google Image Search. The dataset has 10,524 human faces of various resolutions and in different settings, e.g. portrait images, groups of people, etc. All faces are cropped into a base resolution of 36×36 pixels.

For each dataset, we used totally 2000 frontal face images. The faces were selected by filtering away faces which are not

[2] MIT Center For Biological and Computation Learning, http://www.ai.mit.edu/projects/cbcl

[3] http://www.vision.caltech.edu/Image_Datasets/

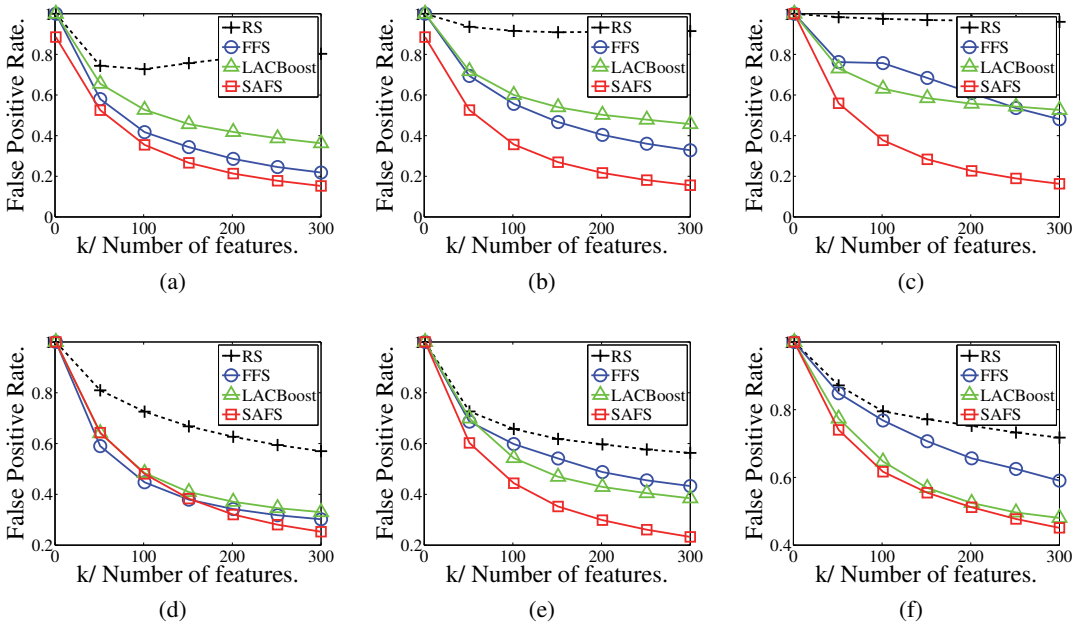

Figure 4: False Positive Rate of the ensemble classifiers using RS, FFS, LACBoost and SAFS algorithm.

high enough resolution, upright, or front facing. The non-face images were sampled and cropped from 273 high resolution scene images, which were similar to Wu et al. (2008). To train a node classifier, our training data contained 1000 frontal face images and 1000 non-face images. In each image, we used 5000 features sampled uniformly from all rectangle features similar to Wu et al. (2008), which contained five type of Harr-like features. For testing, we used 1000 frontal face images and 1000 non-face images in our experiments.

False Positive Rate Measure

In order to compare the performance of the node classifier, we use a fixed minimum detection rate $1 - \beta$ ($\beta = 0.001, 0.005, 0.01$) as the constraint. We then compare the false positive rate of all ensemble classifiers (corresponding to RS, FFS, LACBoost and SAFS) on both two datasets. We compare the number of features up to 300, which is often necessary for the final node classifier (Viola and Jones 2001b; Wu et al. 2008).

We show results on **CBCL Face Database #1** dataset in Figure 4(a), 4(b), 4(c). In regions with less features (≤ 50), only LACBoost ensemble classifier is slightly better than SAFS ensemble classifier. In regions with more features, SAFS ensemble classifier always has the best performance. We show results on **Caltech 10,000 Web Faces** dataset in Figure 4(d), 4(e), 4(f). In regions with less features (≤ 150), SAFS ensemble classifier has comparable performance with LACBoost ensemble classifier. In regions with more features, SAFS ensemble classifier has the best performance. We also demonstrate ROC curves of all ensemble classifiers in Figure 5. The ROC curves show that SAFS ensemble classifier has better performance than FFS and LACBoost ensemble classifiers on both two datasets.

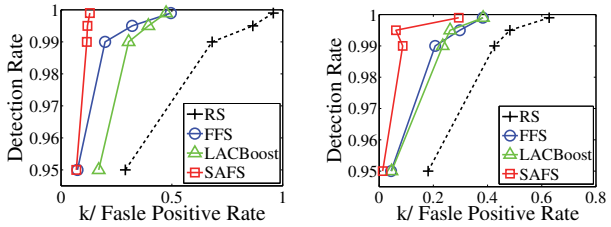

Figure 5: ROC curves comparing ensemble classifiers using RS, FFS, LACBoost and SAFS algorithm (closer to the upper left corner is better).

Conclusion

In cascade object detection, we firstly introduce a new aspect to analysis both asymmetry and intersection problems for feature selection. We formulate the feature selection as a submodular maximization problem. We propose a Submodular Asymmetric Feature Selection or SAFS algorithm, which is a greedy algorithm choosing features maximumly reduce the false positive rate under a constraint of extremely high detection rate. Considering that it is a NP-hard problem, we also provide theoretical guarantee, which is a $(1 - \frac{1}{e})$-approximation bound for SAFS. The experimental results on two real-world face detection datasets demonstrate that SAFS outperforms current state-of-art feature selection algorithms, such as FFS and LACBoost.

Acknowledgments

This research is supported by Australian Research Council Projects: DP-140102164 and FT-130101457.

Appendix

Proof of Lemma 1

Proof. Let $X = \{(x_i, y_i)|i = 1, 2, \ldots, N\}$ denote the dataset and H denote the node classifier. Let

$$N_{-1} = \sum_{i=1}^{N} \mathbf{1}\{y_i = -1\},\ N_1 = \sum_{i=1}^{N} \mathbf{1}\{y_i = 1\} \quad (14)$$

and for all $s, t \in \{-1, 1\}$,

$$N_{s,t} = \sum_{i=1}^{N} \mathbf{1}\{y_i = s, H(x_i) = t\}. \quad (15)$$

According to the definition of FN, FP and ER we have:

$$FP = \frac{N_{-1,1}}{N_{-1}},\ FN = \frac{N_{1,-1}}{N_1}, ER = \frac{N_{-1,1} + N_{1,-1}}{N}. \quad (16)$$

We then have:

$$\begin{aligned} ER &= \frac{N_{-1,1} + N_{1,-1}}{N} \\ &= \frac{N_{-1,1}}{N_{-1}}\frac{N_{-1}}{N} + \frac{N_{1-1}}{N_1}\frac{N_1}{N} \\ &= \frac{N_{-1}}{N}FP + \frac{N_1}{N}FN \\ &= Pr\{y_i = -1\} * FP + Pr\{y_i = 1\} * FN \\ &= \alpha * FP + (1 - \alpha) * FN. \end{aligned} \quad (17)$$

□

References

Ashkan, A.; Kveton, B.; Berkovsky, S.; and Wen, Z. 2014. Diversified utility maximization for recommendations. In *Poster Proceedings of the 8th ACM Conference on Recommender Systems*.

Benenson, R.; Omran, M.; Hosang, J.; and Schiele, B. 2014. Ten years of pedestrian detection, what have we learned? In *Computer Vision-ECCV Workshops*, 613–627. Springer.

Das, A., and Kempe, D. 2011. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint:1102.3975*.

Das, A.; Dasgupta, A.; and Kumar, R. 2012. Selecting diverse features via spectral regularization. In *Advances in Neural Information Processing Systems*, 1583–1591.

Huang, K.; Yang, H.; King, I.; Lyu, M. R.; and Chan, L. 2004. The minimum error minimax probability machine. *The Journal of Machine Learning Research* 5:1253–1286.

Iyer, R. K., and Bilmes, J. A. 2013. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, 2436–2444.

Krause, A., and Golovin, D. 2012. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems* 3:19.

Krause, A.; McMahon, H. B.; Guestrin, C.; and Gupta, A. 2008. Robust submodular observation selection.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research* 9:235–284.

Lampert, C. H.; Blaschko, M. B.; and Hofmann, T. 2008. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, 1–8.

Liu, Y.; Wei, K.; Kirchhoff, K.; Song, Y.; and Bilmes, J. 2013. Submodular feature selection for high-dimensional acoustic score spaces. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 7184–7188.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming* 14(1):265–294.

Paisitkriangkrai, S.; Shen, C.; and van den Hengel, A. 2014. Asymmetric pruning for learning cascade detectors. *Multimedia, IEEE Transactions on* 16(5):1254–1267.

Shen, C.; Wang, P.; Paisitkriangkrai, S.; and van den Hengel, A. 2013. Training effective node classifiers for cascade classification. *International Journal of Computer Vision* 103(3):326–347.

Viola, P., and Jones, M. 2001a. Fast and robust classification using asymmetric adaboost and a detector cascade. *Advances in Neural Information Processing System* 14.

Viola, P., and Jones, M. 2001b. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conference on*, volume 1, I–511.

Viola, P., and Jones, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57(2):137–154.

Wu, J.; Brubaker, S. C.; Mullin, M. D.; and Rehg, J. M. 2008. Fast asymmetric learning for cascade face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30(3):369–382.

Wu, J.; Mullin, M. D.; and Rehg, J. M. 2005. Linear asymmetric classifier for cascade detectors. In *Proceedings of the 22nd International Conference on Machine Learning*, 988–995. ACM.

Wu, J.; Rehg, J. M.; and Mullin, M. D. 2003. Learning a rare event detection cascade by direct feature selection.

Yue, Y., and Guestrin, C. 2011. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, 2483–2491.