Predicting and Evaluating the Power of Shared Features

Thomas S. Stepleton
Robotics Institute, Carnegie Mellon University
E-mail: tss@ri.cmu.edu

Abstract

Several recent efforts in multi-class feature-based object recognition employ shared features*, or features that simultaneously belong to multiple class models. These approaches claim a considerable time savings by reducing the total number of features used by all models, thereby lessening the concomitant computational effort of finding the features in images. In this paper we derive a Bayesian framework for predicting and evaluating the performance of shared feature-based recognition systems. We then use this framework to predict the performance of several instances of a simple multi-class object detector.*

1. Introduction

Traditional single-class feature-based object recognition systems (e.g. [6]) usually describe an object as some kind of collection of distinctive local appearance traits. Whenever one of these traits is found in an image, it serves in one form or another as a "vote" for the presence of the target class. To recognize multiple classes with such systems, it serves to independently train a new recognition system for each class and run them all in parallel.

Since the detection of each local image trait generally requires a separate analysis of all or part of the image, we can usually expect the overall computational time of a recognition algorithm to be proportional to and most directly influenced by the number of features in use. In that light, the multi-class recognition scheme outlined above seems inefficient. Surely some of the classes must have traits in common—detecting them separately for each class is wasteful. In response to this observation, several recent papers have described or used *shared feature*-based systems [7],[2],[5],[1]. These systems allow features to belong to multiple object models—in effect, a detection event for one of these features is a vote for the presence of multiple objects. The votes of many such features will ideally have in common the object actually present in the image.

Torralba et al. [7] and Krempp et al. [5] in particular have observed considerable computational savings with shared feature recognition techniques: they report that the number of features necessary for effective object recognition appears to be proportional to the logarithm of the number of classes the system must recognize. This is an empirical observation that appears to be well-borne by the data, but to our knowledge there has been no theory presented that allows us to expect this performance.

In this paper we seek a framework for predicting and evaluating the performance of a shared feature-based object recognition system based on an empirical evaluation of the discriminative characteristics of its constituent features. This framework should be useful for estimating what quality and quantity of features will be necessary for a desired level of recognition performance, as well as checking whether a recognition system is truly performing as well as it should. We additionally derive the logarithmic result discussed above and present further analysis discussing under what conditions such performance can be expected.

2. Posterior Likelihood of Object Presence

Given the detection of a certain number of features in an object's model, how certain can we be that the object is actually present? This question is the starting point for our analysis. If C denotes the presence of the target object or class and $F_1, \ldots, F_n$ denotes the detection of features 1 through n in our target object model, we seek the posterior likelihood $P(C|F_1, \ldots, F_n)$, which Bayes' Rule renders as

$$P(C|F_1, \ldots, F_n) = \frac{P(F_1, \ldots, F_n|C)P(C)}{P(F_1, \ldots, F_n)}. \quad (1)$$

Note that since F_i merely connotes the detected presence of a particular feature, equation 1 appears chiefly to concern the simple variety of object models known as "bags" of features, which consist only of lists of features present in a class without any sort of specification of their spatial arrangement. Depending on the precise meaning of "feature detection event," however, the joint occurrence of $F_1, \ldots, F_n$ may implicitly connote some kind of structural requirement. In this case, some of the independence assumptions in the following analysis may not hold.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

2.1. The conditional feature detection likelihood and the object prior

Our derivation of the numerator of (1) reveals several of the assumptions we make in our analysis. We first assume a uniform prior over all objects, giving $P(C) = 1/|\mathbf{C}|$, where $|\mathbf{C}|$ is the size of the set of all objects. We have chosen not to incorporate a background model in this first analysis (i.e. we assume that the system will always be presented with *some* object in our object set), but we expect that adding one should be relatively straightforward.

For the likelihood term, we assume that the detection of each of the model features 1 through n is independent given C, the presence of the target object; that is, $P(F_1, \ldots, F_n|C) = P(F_1|C)P(F_2|C) \cdot \ldots \cdot P(F_n|C)$. This is a common assumption for reasoning about feature-based detection. We further assume that the joint likelihood of these feature detection events can be approximated by assuming the same true positive detection probability for each feature, or $P(F_i|C) = g$ for $i = 1 \ldots n$. Altogether then, $P(F_1, \ldots, F_n|C) = g^n$, and the numerator of (1) becomes $g^n/|\mathbf{C}|$.

2.2. The marginal feature detection likelihood

The denominator of (1) is the marginal likelihood of features 1 through n ever being detected together. It is here that shared feature techniques differ from multi-class object recognition systems with separate feature sets for each object, since the number of circumstances for "legitimate" detections of a single feature increases with the degree of feature sharing between classes. To begin finding $P(F_1, \ldots, F_n)$ then, we marginalize over two broad circumstances: the likelihood of observing features 1 through n given that the target object is present, and the observation likelihood given that it is not (denoted $\bar{C}$).

$$P(F_1, \ldots, F_n) = P(F_1, \ldots, F_n|C)P(C) + P(F_1, \ldots, F_n|\bar{C})P(\bar{C}). \quad (2)$$

The first term is the same as the numerator computed above. With the uniform prior over objects, $P(\bar{C})$ in the second term is $(|\mathbf{C}| - 1)/|\mathbf{C}|$. As for the remaining non-target object feature detection likelihood $P(F_1, \ldots, F_n|\bar{C})$, we assume that the models of the $|\mathbf{C}| - 1$ remaining objects could contain any combination of features of size n or larger, *including* the features 1 through n in our original target object model. This non-exclusive policy is apt when one considers a circumstance where only a few features are detected—these few may all be shared by several models and may not be enough to select a single object.

With that in mind, we wish to marginalize over all possible non-target objects. An object may contain any subset p of features 1 through n, so we must marginalize over the power set $\mathbf{P}$ of those features.

$$P(F_1, \ldots, F_n|\bar{C}) = \sum_{p \in \mathbf{P}} P(F_1, \ldots, F_n|\bar{C}_p, \bar{C})P(\bar{C}_p|\bar{C}). \quad (3)$$

Here, each $\bar{C}_p$ represents the presence of a particular object that is not the target object. Note that an object containing feature i does not overlap with one containing i and j, since the absence of a feature in an object's model is just as important in the following analysis as its presence.

Within the sum, the conditional detection likelihood $P(F_1, \ldots, F_n|\bar{C}_p, \bar{C})$ depends on which of the features 1 through n are actually in p. Any that are missing will only be "detected" as false positives, and just as for true positives, we assume a general approximate false positive rate $\tilde{g}$. If k out of the n detected features are actually absent from p, then given the conditional independence assumptions mentioned earlier, the likelihood is $g^{n-k}\tilde{g}^k$.

To compute $P(\bar{C}_p|\bar{C})$ we introduce a new parameter s. This *shared factor* is the likelihood of any one feature being included in a particular object model, and again we assume this likelihood is the same for all features. We further assume that features are independently included in object models, which is an ideal assumption but a desirable one for a diverse and descriptive feature set. Effectively, then, each object model can be thought of as a random draw of features from the entire feature set. For this reason, $P(\bar{C}_p|\bar{C}) = P(\bar{C}_p)$. Thanks to our independence assumption, the likelihood of any model lacking k out of the n features and retaining the remaining $n - k$ is

$$P(\bar{C}_p|\bar{C}) = P(\bar{C}_p) = s^{n-k}(1 - s)^k. \quad (4)$$

Finally, since the g, $\tilde{g}$, and s parameters are shared across all models, we can compute the summed likelihoods for all models lacking k out of n features in one step, as in $\sum_{k=0}^{n} \binom{n}{k} g^{n-k}\tilde{g}^k s^{n-k}(1 - s)^k$, where we perform that computation for each k to get the overall likelihood. Applying the Binomial Theorem, we achieve our final expression for the non-target object feature detection likelihood:

$$P(F_1, \ldots, F_n|\bar{C}) = (gs - \tilde{g}s + \tilde{g})^n. \quad (5)$$

2.3. Combining the parts

Bringing the above pieces together and performing some straightforward algebraic simplification, we can express the posterior likelihood of object presence as

$$P(C|F_1, \ldots, F_n) = \frac{1}{1 + \frac{|\mathbf{C}|-1}{g^n}(gs - \tilde{g}s + \tilde{g})^n}.$$

By factoring g^n out of $(gs - \tilde{g}s + \tilde{g})^n$ and expressing the ratio of the false positive rate to the true positive rate

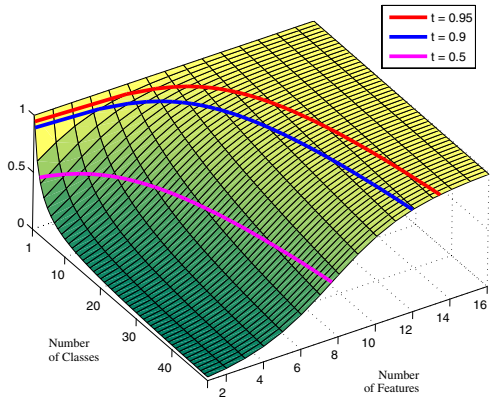


Figure 1. The posterior object presence likelihood with $r = 0.33$ and $s = 0.4$. The thick lines denote contours of equal probability t and are proportional to $\log |\mathbf{C}|$.

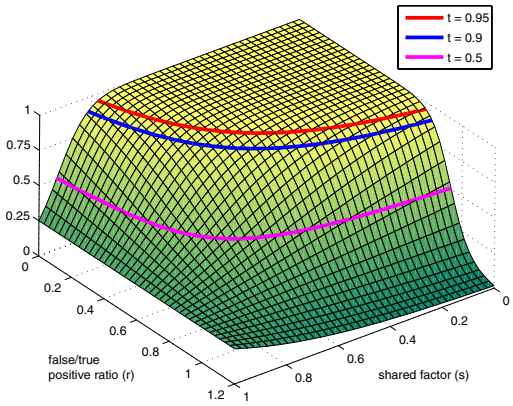


Figure 2. The posterior object presence likelihood with $|\mathbf{C}| = 4$ and $n = 16$. The thick lines denote contours of equal probability t and are described by (9).

$\tilde{g}/g$ as r, our final expression for the posterior is

$$P(C|F_1, \ldots, F_n) = \frac{1}{1 + (|\mathbf{C}| - 1)(s - rs + r)^n}. \tag{6}$$

The use of r in favor of g and $\tilde{g}$ emphasizes an important point: our posterior likelihood says nothing about the probability of the feature detection events $F_1, \ldots, F_n$ themselves. Rather, it computes the likelihood of object presence assuming the features have already been seen.

3. Examining the Presence Posterior

With our presence posterior expression in hand, we can now examine what happens if we systematically vary its parameters. Our first goal is to see if the number of features necessary for reliable object recognition really is logarithmic in the number of classes.

If we fix a posterior detection likelihood threshold t and solve (6) for n, the number of features, we get the following expression:

$$n = \frac{\log(\frac{1}{t} - 1) - \log(|\mathbf{C}| - 1)}{\log(s - rs + r)} \tag{7}$$

If we hold r and s fixed, the first logarithm in the numerator and the one in the denominator are constants; we will briefly represent the former as a and the latter as $-b$ since it is always less than or equal to 0 if $r \leq 1$. We can rewrite (7) as

$$n = \frac{a}{-b} + \frac{1}{b} \log(|\mathbf{C}| - 1), \tag{8}$$

illustrating that for a given r and s, n is indeed $O(\log |\mathbf{C}|)$. Figure 1 shows how the posterior likelihood changes with $|\mathbf{C}|$ and n if s is fixed at 0.4 and r at 0.33 (as it would be for e.g. $g = 0.6$ and $\tilde{g} = 0.2$).

We can also examine how the presence posterior responds to changes in the shared factor s and the true/false positive ratio r. Figure 2 plots the posterior for $n = 16$ features and $|\mathbf{C}| = 4$ objects. We can see that as s draws closer to 1, the posterior approaches 0.25—in other words, as the likelihood of the n detected features being in each object becomes the same, the likelihood of the target object's presence becomes chance. For smaller s values, meanwhile, as r exceeds 1 (i.e. the false positive rate exceeds the true positive rate), detection performance becomes worse than chance.

Contours of fixed posterior likelihood t can be found by solving for s or r, and are of the following form:

$$s = \frac{\left(\frac{\frac{1}{t} - 1}{|\mathbf{C}| - 1}\right)^{\frac{1}{n}} - r}{1 - r}. \tag{9}$$

Figure 2 shows three such contours, which evince a tradeoff between the degree of possible feature sharing and the accuracy required by the feature detectors. Depending on the system in use, the optimum of this tradeoff might be the point on the contour nearest to $r = 1, s = 1$. For real numbers in the first quadrant, this minimum can be found by computing r_{opt} as

$$r_{\text{opt}} = 1 - \sqrt{1 - \left(\frac{\frac{1}{t} - 1}{|\mathbf{C}| - 1}\right)^{\frac{1}{n}}}. \tag{10}$$

Conveniently, since r and s have symmetric roles in (6), $s_{\text{opt}} = r_{\text{opt}}$.

The analysis so far has assumed that s and r can remain constant while $|\mathbf{C}|$ grows. This seems unlikely in

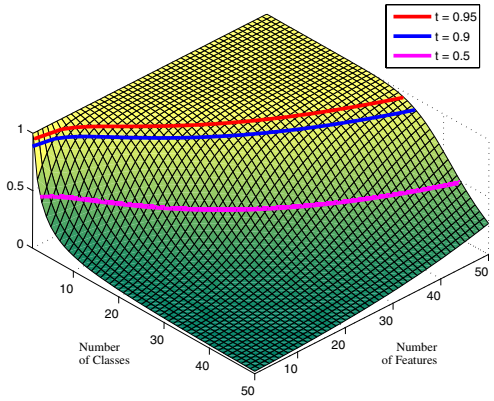


Figure 3. The posterior object presence likelihood with r determined by (11) and $s = 0.4$. The thick lines denote contours of equal probability t and are described by (12).

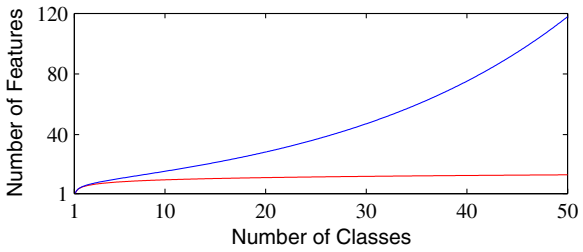


Figure 4. A comparison of equal probability contours—the lower, red line is the $t = 0.95$ contour from Figure 1, and the upper, blue line is the $t = 0.95$ contour from Figure 3.

practice. As the number of objects increases, and as a feature becomes a member of more classes, we might expect the increasingly diverse variety of object appearances to have more ways of "tricking" a feature detector into generating more false positives and false negatives. We can perform a preliminary simulation of this effect by allowing r to slowly become asymptotically close to 1 as $|\mathbf{C}|$ increases. Parameterizing the rate of an exponential feature detection performance decay with $\clubsuit$, we might say

$$r = 1 + (r_1 - 1)e^{-\clubsuit(|\mathbf{C}|-1)}, \tag{11}$$

where r_1 is the initial value r assumes at $|\mathbf{C}| = 1$. Inserting (11) into (6) and solving for n, we get

$$n = \frac{\log(\frac{1}{t} - 1) - \log(|\mathbf{C}| - 1)}{\log(e^{-\clubsuit(|\mathbf{C}|-1)}(s - r_1 s + r_1 - 1) + 1)}. \tag{12}$$

Figure 3 shows the posterior object presence likelihood with s fixed and r varying according to (11) ($q = 0.04$, $r_1 = 0.33$). Though the fixed likelihood contours start out resembling the logarithmic curves in Figure 1 (note that the scale of the n axis has changed), they soon start to grow rapidly. Apparently n is no longer proportional to $\log |\mathbf{C}|$.

Figure 4 compares the fixed likelihood contour for a fixed r with that of an r varying with (11). The initial, logarithm-like, down-concave behavior of the variable r curve is readily apparent.

4. Empirical Application

To demonstrate the predictive power of (6), we have created a simple multi-class object recognition task. While the performance of our unsophisticated object detection scheme is much poorer than that of practical systems, for reasons further discussed below, the results shown here still serve as a validation of our evaluation model. We caution the reader that the following system is an expositional tool and is neither meant to be a practical recognition system nor, as the results might suggest, an indictment of its constituent components or of shared feature techniques in general.

4.1. The recognition task and data

Our simplified object recognition task requires the detector to discern between images of ten different objects photographed in a variety of lighting conditions. Since our equations do not model background feature detection events, the objects are always placed in front of a white background, and no images besides those of the objects are seen. Figure 5 shows images of the ten objects.

For each object we collected twelve grayscale photographs. One image of each collection was added into an image set we used for training the object recognition system; three more were added to a set we used to find the s and r statistics; and the remaining eight were added to the test image set.

4.2. The recognition system and training

Our recognition system employs PCA-SIFT, an adaptation of the SIFT algorithm ([6]) that represents local scale-invariant image patches as 36-dimensional vectors [4]. The system begins by converting a test image into a "bag" of these vectors (i.e. spatial location, scale, and orientation information is discarded). A feature is detected when any of the vectors in this bag is within a certain learned L2 distance from a learned point in the 36-dimensional space.

Our object models are trained so that each model contains a distinct combination of n features (see Figure 6 for example sets of object models). Therefore, if at least

Figure 5. Training images of the ten objects used in our object recognition task. Test images closely resembled these training images with most variation stemming from lighting.

n features are detected, the system declares those objects whose models are subsets of the detected feature set to be present in the image.[1]

Because (6) predicts the likelihood of object presence given a number of detected features, we designed our recognition system mostly so that these joint detection events would be frequent enough to give us meaningful statistics. As a result, n is very small—typically around four—and recognition performance suffers. Most practical feature-based object recognition systems, even ones that only recognize single classes, rely on many more features.

To train our object detection system, we begin by creating a fixed feature "schedule" for object models—we stipulate beforehand which features belong to each model (Figure 6). In this way we ensure that there is little variance in the degree to which features are shared among objects, and also that our estimate of s will be reliable.[2] In systems with more features, the Law of Large Numbers will presumably furnish this reliability and such careful engineering will not be required.

Using the object model feature schedule, we search for individual features which will be detected in the correct classes' training images and not in any others. For our system, this amounts to picking a detection radius and then scanning every single PCA-SIFT vector in the training images to see if it will serve as the learned point for the feature. If no such vector can be found, the search is repeated with a new radius. This approach is hardly guaranteed to work, but we were able to find the scheduled features for every object model regardless. Our 1.8

FEATURE/CLASS SCHEDULES

Column 1: List of objects for modeling (see Figure 5).
Column 2: System with 3 features/model, 8 features total.
Column 3: System with 5 features/model, 7 features total.
Column 4: System with 2 features/model, 2 features total.

	Feature 12345678	Feature 1234567	Feature 12345
altoids	●●●	●●●●●	●●
grammar	●●●	●●●●●	●●
matzo1	●●●	●●●●●	●●
phone2	●●●	● ●●●●	●●
plate	●●●	●● ●●●	● ●
pocky	●●●	●●● ●●	● ●
raisins	● ●●	●●●● ●	● ●
stix	●● ●	●●● ● ●	● ●
teabox	●● ●	● ●●● ●	● ●
tmobile	●● ●	● ● ●●●	● ●
$\sim$	$s = 0.375$	$s = 0.714$	$s = 0.4$

Figure 6. Feature/class "schedules" for three shared feature-based object recognition systems. Each numbered column corresponds to a specific feature in the model. A • symbol indicates that a feature is present in the model of an object. The bottoms of the columns show the estimated shared factor s for each system.

[1] Since each image in the recognition task contains only one object, the problem of the union of detected models 1 and 2 also containing all the features of model 3 is not addressed by this experiment. Nevertheless, it is present in our analysis in the "non-exclusive policy" used to compute the marginal likelihood of joint feature detection.

[2] We say "estimate" even in this engineered case since in some circumstances the mean shared factor will be fractional.

GHz Pentium computer was able to find all the features for a particular recognition system in a few minutes.

4.3. Prediction and performance results

To predict the performance of our recognition system, we need to find its s and r statistics. To find s, we refer back to our model feature schedule and find the average fraction of classes to which each feature belongs. Finding r requires us to estimate the rate of false and true positives (g and $\tilde{g}$) for each feature, which we do by testing on an another set of images (in our experiments, an additional three images for each object). We compute both rates and an r value just for the one feature by counting the feature's appearances in images where it should and shouldn't be seen. To arrive at a final r estimate for the entire system, we average over the r values of each of the features.

Measuring the actual performance of the system is likewise straightforward: we scan every image in our 80-image test set and record every object the system detects. The empirical object presence likelihood is the ratio of true object detections to the total number of object detections. Note that as the number of features increases, object detection events of any sort become scarcer and hence this likelihood estimate becomes less reliable.

Since the r and s values are computed by averaging counts of true positives, false positives, and model membership for each feature, we can use the bootstrap on these counts to estimate a distribution over posterior presence likelihood estimates [3].[3] For each iteration of the bootstrap algorithm, we sample our set of $|\mathbf{F}|$ individual feature r values and our set of $|\mathbf{F}|$ individual feature s values with replacement (where $|\mathbf{F}|$ is the total number of features in the detector) to generate temporary new sets of r and s of size $|\mathbf{F}|$. The means of these temporary sets are inserted into (6) to generate a sample from an estimated overall distribution of likelihood estimates.

After iterating many times, the resulting distribution can be used to generate quartile box plots for our computed Bayesian likelihood. We present these plots along with our other results for a number of shared feature detection experiments in Figure 7. Of incidental interest is the theoretical best performance of each of these detectors, computed by setting r to 0 (i.e. no false positives). Often this performance is quite poor due to the limited number of features in each object model.

5. Conclusions and Future Work

We have presented a Bayesian framework for predicting and evaluating the performance of multi-class feature-based object recognition systems employing shared features. We have also shown that if the discriminative power of the shared features remains constant regardless of how many classes a system must recognize, the number of features necessary to achieve a certain threshold of performance is proportional to the logarithm of the number of classes. This is a theoretical confirmation of an empirical claim in the shared-feature literature. Nevertheless, we suggested that if the features' discriminative power does *not* remain constant, this property no longer holds. Finally, we have predicted the performance of a real multi-class shared feature-based object recognition system.

Our analysis relies on several key assumptions: that the overall performance of a recognition system's features can be adequately summarized with a single statistic r, that the likelihoods of feature detection events are independent given an object's presence, and that each feature in the recognition system is independently shared among the object models. Future work should determine the aptness of these assumptions and the effect on our expression's performance predictions if any of them are violated. Additional investigation should more clearly address the issue of how features and shared feature systems perform when a recognition system must recognize more objects.

[3] This analysis relies on the assumption that r and s are independent.

Acknowledgments

Thanks to Yan Ke, Caroline Pantofaru, Sajid Siddiqi, and the hot glue gun manufacturers of the world. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

References

[1] T. Stepleton and T.S. Lee. "Using Co-occurrence and Segmentation to Learn Feature-based Object Models from Video." In *IEEE Workshop on the Applications of Computer Vision*, January 2005.

[2] E. Murphy-Chutorian, and J. Triesch. "Shared features for scalable appearance-based object recognition." In *IEEE Workshop on the Applications of Computer Vision*, January 2005.

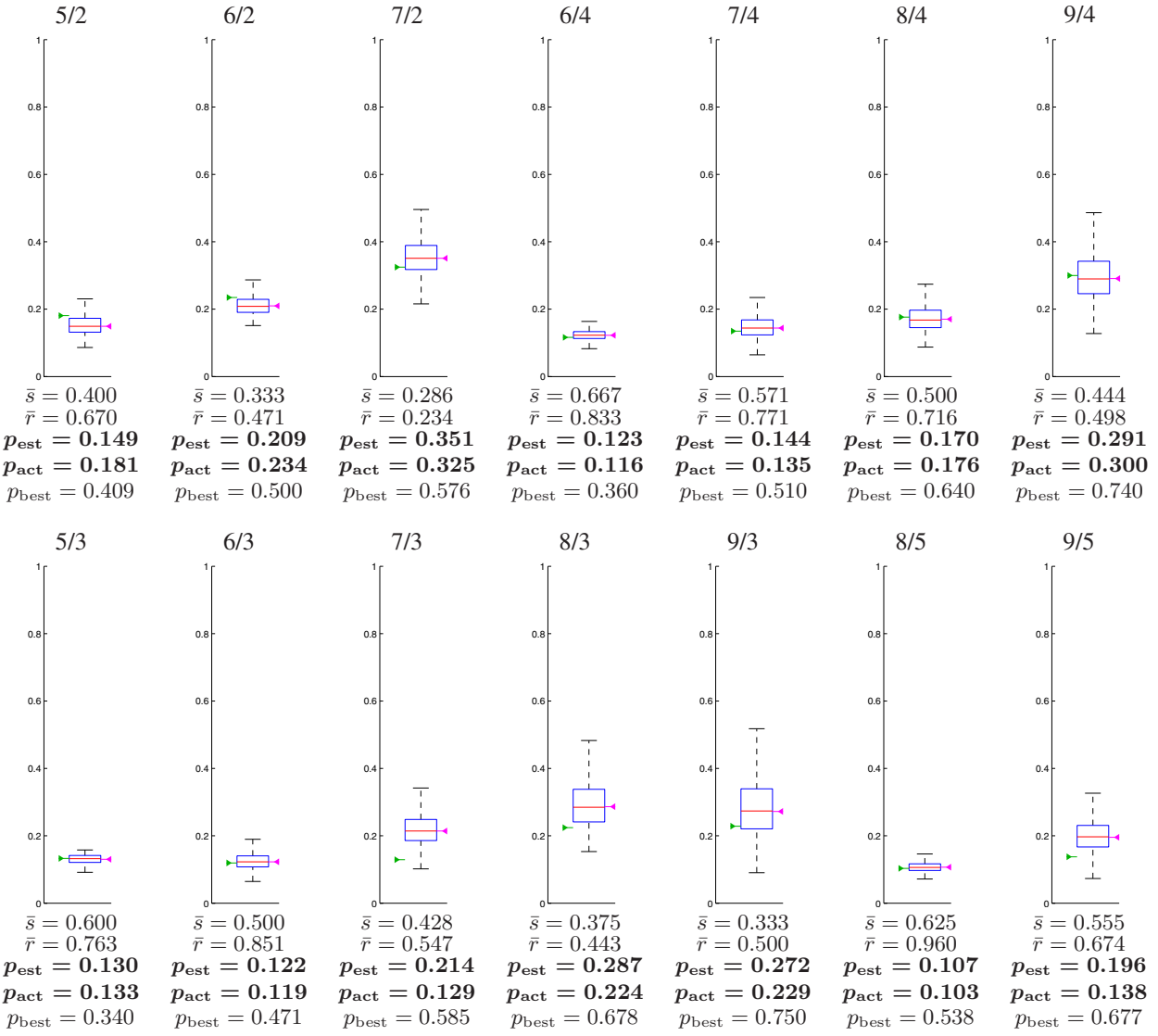
[3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*, Springer, 2001.

[4] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors." In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2004.

[5] S. Krempp, D. Geman, and Y. Amit. "Sequential learning of reusable parts for object detection." Technical report, CS Johns Hopkins, 2002.

[6] D. Lowe. "Object recognition from local scale-invariant features." In *Intl. Conf. on Computer Vision*, September 1999.

[7] A. Torralba, K. Murphy, and W. Freeman. "Sharing features: efficient boosting procedures for multiclass object detection." In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2004.



Legend

5/2: Five features in the recognition system/Two features per object model (n)
$\bar{s}$: Estimated (mean) shared factor
$\bar{r}$: Estimated (mean) false/true positive ratio
$\boldsymbol{p_{\text{est}}}$: Predicted object presence likelihood (left/magenta dart (◄) in plots)
$\boldsymbol{p_{\text{act}}$: Actual test set object presence rate (right/green dart (►) in plots)
p_{best}: Theoretical best likelihood for this experiment ($s = \bar{s}$, $r = 0$).

Figure 7. Results from several shared feature detection experiments. Each boxplot corresponds to a different experiment. The boxplots are generated by running the bootstrap on the lists of r and s values computed for each feature—at each step, the sample mean r and s values are plugged into (6). 100,000 iterations of the bootstrap renders a distribution over p_{est} values whose second and third quartiles are contained within the boxplot boxes. Red lines in the boxes are medians. Whiskers extend to the furthest distribution value within 1.5 times the interquartile range from the edge of the box, though generally there are outliers beyond them (not shown).