Inducing features from visual noise

Andrew L. Cohen
Department of Psychology, University of Massachusetts, Amherst, MA, USA

Richard M. Shiffrin
Departments of Psychology and Cognitive Science, Indiana University, Bloomington, IN, USA

Jason M. Gold
Departments of Psychology and Cognitive Science, Indiana University, Bloomington, IN, USA

David A. Ross
Department of Computer Science, University of Toronto, Toronto, ON, Canada

Michael G. Ross
Department of Psychology, University of Massachusetts, Amherst, MA, USA

We present new experimental and mathematical techniques aimed at determining the features used in visual object recognition. We conceive of these features as the parts of an object that are treated as unitary wholes when recognizing or discriminating visual objects. For example, consider a task classifying a visual target presented in pixel noise as a “P” or a “Q”. The features may correspond to particular shapes of the target letters. Two such features for “P”, for example, might be a vertical line and upper-right-facing curve. The decision may be encoded in terms of particular values of such features, and an appropriate combination of these values may determine how the expression is perceived. We utilize recent advances in statistical machine learning techniques to uncover the features used by human observers.

Keywords: feature induction, reverse correlation, classification in noise, Gaussian mixture model


Introduction

Imagine that you are asked to classify a visual target presented in pixel noise as a “P” or a “Q”. In one common class of theories, the decision is based on matching the appropriately scaled display to memory representations of each letter. Most template models fall into this class (Tarr & Bulthoff, 1998). However, our encoding, storage, and retrieval processes may be more structured than the simple template model. It may be that the decision favoring “P” over “Q” is not based on two letter–template matches, but instead on matches of the display to a larger set of units called “features” (two such features favoring a choice of “P” might be a vertical line and an upper-right-facing curve). In feature models, the matches to the various features (some favoring “P” and some favoring “Q”) are combined by an appropriate decision rule. Features models can be viewed as a generalization of template models in which the number of templates is expanded and not restricted to holistic representations of the target stimuli.

This article takes a first step toward developing new empirical and computational techniques to automatically uncover the visual features used by human observers. Although the long-range goal is the use of the techniques in complex tasks in which the potential features are not obvious by inspection (e.g., classification of faces into pleasant and unpleasant or classification of radiographs into those showing cancer and those not showing cancer), the goal of this article is far more modest. To introduce the technique and motivate further research, we apply our techniques to simple empirical and simulated tasks.

The article is organized as follows. First, we define a feature and present a number of potential properties that a good feature induction algorithm should be able to accommodate. Second, we outline the basic empirical technique used to gather data and discuss an experiment with a simple feature structure. Third, we describe Gaussian mixture modeling, a powerful feature recovery technique. Fourth, to gain an intuition on how Gaussian mixture modeling recovers features, we apply this technique to a set of simulated data and discuss how the characteristics of the features impact our ability to induce them. Fifth, we apply this technique to data simulated from a more complicated feature structure. Sixth, we...
apply the technique to human data. Finally, we conclude and discuss some of the many potential future directions for this research.

**Features: Definition**

We begin by describing features in the context of the simplified tasks of this article and then discuss generalizations. The tasks in this article involve simple perceptual judgments and discriminations. For these tasks, each possible target stimulus is a set of pixels arranged in a rectangular grid, and each pixel has an assigned grayscale value. For example, one target might be a set of black pixels in the shape of a “P”, with the remaining pixels in the grid set to neutral gray. The tasks we will consider involve presentation of a target stimulus for identification, but with independent Gaussian noise added to every pixel. Each feature in memory is defined to be a grid of pixels of the same size as the stimulus, each with an assigned grayscale value. A feature could therefore exactly match one of the target stimuli (and hence be a traditional template), could match a part of one target (e.g., black pixels matching the vertical line on the left of the target “P”, with neutral gray for the remaining pixels), or be an arbitrary set of gray values (partially matching the different targets to differing degrees). Furthermore, a feature could either equally weight each pixel location (as in a traditional template) or differentially weight pixel locations (e.g., look for a match of the vertical line of the “P” while disregarding all other pixels; also, see Ullman, Vidal-Naquet, & Sali, 2002). We assume a decision process in which the test display (target plus noise) is matched separately to each feature in memory. Each match will provide a degree of evidence favoring or disfavoring each alternative target, and a reasonable decision rule must be assumed to combine the separate matches to reach a decision.

In this model, features can overlap, but we assume that the system calculates evidence for each feature separately in that the presence of one feature does not directly impact the determination of the presence another feature. This is clearly not optimal. For example, if two features differ slightly in only 1 pixel, the evidence provided by one is highly redundant to the evidence provided by the other. Despite this issue, assuming little overlap is reasonable. If it is assumed that the observer has a large number of features in memory and must select a small subset to use in a discrimination task, it is unlikely that he or she would choose highly redundant features.

This definition of “feature” and decision-making approach is most useful for very simple tasks with well-defined targets. The detection of a “P” of unknown size, position, orientation, font, and gray value might not best be characterized by the uncountable feature templates corresponding to each possible variant. It is also possible to imagine high-level abstract features that cannot be defined by simple pixel templates (such as “happy” as a face feature). There are other potential definitions of features (e.g., Treisman, 1986; Wolfe, 1998), but the template feature definition is a useful starting point for many visual discrimination tasks. Ideas on how to extend our approach to more complex situations will be discussed in the Discussion section.

**Experimental paradigm**

We turn next to a simple classification task from which observer data will be analyzed to demonstrate a computational technique that can be used to extract and induce features. The experimental paradigm is based on the reverse-correlation or response-classification technique (e.g., Ahumada & Lovell, 1971). On each of numerous trials, an observer’s task is to identify a stimulus presented in a noise field, much like identifying an object viewed on a badly tuned television. Although it can be dangerous to use low-dimensional examples to guide intuition about high-dimensional situations, it is useful at this stage to consider a very simple situation where the images consist of 2 pixels (i.e., dimensions). The term “pixels,” as used in this article, refers to the elements produced by dividing a grayscale image into a grid in which each element has a uniform brightness. The pixels in a stimulus might be scaled so they are displayed over several pixels on the monitor used to present the stimuli.

On each trial, the observer is shown a stimulus generated by randomly selecting one of two targets, depicted in Figure 1, with independent Gaussian noise added to both locations. A series of possible stimuli are shown in Figure 2. The observer’s task is to indicate which target was presented. In this very simple categorization task, there are two response categories (A and B). The contrast of the targets is placed at a level where performance is at threshold (near 71% correct). The data are the noisy test images (e.g., Figure 2) and observer’s responses (A or B) for every trial.

On many trials, the noise will cause the observer to make judgment errors. For example, if Target A is the true target, the noise will sometimes be distributed in such a way as to make this stimulus look more like Target B. The data from such a study can be analyzed to determine the features that observers use to make classifications.

The classification image technique is used to assess the degree to which different image pixels correlate with...
overcomes this limitation.

features. The algorithm presented in the next section indicate if the subject is processing subsets of pixels with difference template per response category. It cannot certain limitations, such as its production of a single Sekuler, 2000; Watson, 1998) tasks. This technique has auditory (e.g., Ahumada & Lovell, 1971) and visual (e.g., Ahumada & Beard, 1999; Gold, Murray, Bennett, & Sekuler, 2000; Watson, 1998) tasks. This technique has been applied to tion judgments. This technique has been applied to subsets of pixels that define the features used to classify. This section provides a description of one classification image and shows the degree to minus the targets) from trials on which the subject gives a summing (averaging all the noise fields (the noisy stimuli noise).

Figure 2

Sample two-pixel stimuli (contrast-reduced targets plus noise).

observers’ responses of A or B. This is achieved by summing (averaging) all the noise fields (the noisy stimuli minus the targets) from trials on which the subject gives a particular choice (e.g., A) and subtracting the result from a similar sum for the other choice. The resulting map is called a classification image and shows the degree to which different pixels produced the differential classification judgments. This technique has been applied to auditory (e.g., Ahumada & Lovell, 1971) and visual (e.g., Ahumada & Beard, 1999; Gold, Murray, Bennett, & Sekuler, 2000; Watson, 1998) tasks. This technique has certain limitations, such as its production of a single difference template per response category. It cannot indicate if the subject is processing subsets of pixels with features. The algorithm presented in the next section overcomes this limitation.

The Gaussian mixture modeling technique

Although it provides a wealth of useful data, the classification image itself is only a map of pixel correlations and does not allow us to extract the separate subsets of pixels that define the features used to classify the stimuli. The data from such studies, however, can be analyzed with unsupervised machine learning techniques that allow induction of the perceptual features used in classification. This section provides a description of one such technique: the Gaussian mixture model (GMM; Duda, Hart, & Stork, 2001).

For the current endeavor, the goals of a GMM are to cluster the noisy images in a given response category (e.g., Figure 2) into a fixed number of groups or clusters, to learn a probability distribution that characterizes each cluster, and to learn a prior distribution over the clusters. As its name implies, the clusters of a GMM are modeled by multivariate Gaussian distributions, each parameterized by a mean vector and a covariance matrix over the pixels. The prior distribution is a measure of the probability with which a random image would belong to each of the clusters. When a GMM has been fit to a collection of images, each mean vector can be interpreted as an idealization of the images in a cluster; and the covariance matrix, as a measure of the cluster’s variability or noise. In the present situation, each of the clusters corresponds to a feature, and the mean vector of such a cluster corresponds to the grayscale pixel values of that feature.

Mixture models (of which GMMs are but one instance) differ from many other clustering approaches in that data images are assigned to a cluster probabilistically: The probability that image \( \bar{Y}_n \) belongs to cluster \( k \) may be nonzero for many different \( k \), as long as the sum of these probabilities equals one. GMMs are a natural way to discover visual similarities that allow a collection of images to be grouped, and, as we shall see, these groups correspond closely to the image features.

Given a collection of data images, the parameters of a GMM are fit to (approximately) maximize the probability of the data given by

\[
\prod_{n=1}^{N} P(\bar{Y}_n) = \prod_{n=1}^{N} \sum_{k=1}^{K} P(\bar{Y}_n | \mu_k, \sum_k) \pi_k. \tag{1}
\]

This calculation can be performed using an iterative expectation–maximization procedure (e.g., Hastie, Tibshirani, & Friedman, 2001) starting from a random initialization of the model parameters, as follows.

Given \( N \) images, \( \bar{Y}_1, \ldots, \bar{Y}_N \), of \( D \) pixels each and \( K \) clusters, let \( p_{nk} \) be the (posterior) probability that image \( \bar{Y}_n \) belongs to cluster \( k \); \( (\mu_k, \sum_k) \) be the mean and covariance of Gaussian \( k \), respectively; and \( \pi_k \) be the prior probability that a point is generated by cluster \( k \). Then,

1. For each data image \( \bar{Y}_n \) and each cluster \( k \), compute the probability of image \( \bar{Y}_n \) under cluster \( k \) as the multivariate Gaussian density given by

\[
P(\bar{Y}_n | \mu_k, \sum_k) = (2\pi)^{-D/2} |\sum_k|^{-1/2} \exp \left( -\frac{1}{2} (\bar{Y}_n - \mu_k)^T \sum_k^{-1} (\bar{Y}_n - \mu_k) \right). \tag{2}
\]

2. For each image, use Bayes’ rule and the probabilities computed in Step 1 to obtain the probability that image \( \bar{Y}_n \) belongs to cluster \( k \).

\[
p_{nk} = \frac{P(\bar{Y}_n | \mu_k, \sum_k) \pi_k}{\sum_k P(\bar{Y}_n | \mu_k, \sum_k) \pi_k}. \tag{3}
\]
3. For each cluster \( k \), update the parameters for the corresponding Gaussian by computing the mean and covariance of the data points, where the contribution of each point is weighted by the probability with which it belongs to \( k \).

\[
\mu_k = \frac{\sum_{n=1}^{N} p_{nk} \bar{Y}_n}{\sum_{n=1}^{N} p_{nk}} \tag{4}
\]

\[
\sum_k = \frac{\sum_{n=1}^{N} p_{nk} (\bar{Y}_n - \mu_k) (\bar{Y}_n - \mu_k)^T}{\sum_{n=1}^{N} p_{nk}} \tag{5}
\]

4. Update the prior for each cluster \( k \).

\[
\pi_k = \frac{1}{N} \sum_{n=1}^{N} p_{nk} \tag{6}
\]

5. Repeat, starting at Step 1, until the parameter estimates do not change significantly from one repetition to the next.

To reduce the number of estimated parameters (possibly allowing this technique to scale to higher dimensional images), the covariance of each component was assumed to be a constant diagonal matrix. When applying the GMM, local minimum solutions are not uncommon, and so the best (highest likelihood) of 10 random restarts was selected (e.g., Hastie et al., 2001).

### Inducing features from the two-pixel feature structure

In this section, we show how the GMM can be used to induce features from the two-pixel example and discuss some of the characteristics that the stimuli and features must possess for this method to work.

The GMM tries to find clusters in the data set. Each cluster is represented as a Gaussian distribution with as many dimensions as there are pixels in the displays (two in this example). These clusters can be thought of as local modes in the data. The GMM will only successfully recover the features if they correspond to the modes in the data distribution.

**Figure 3** is a representation of the stimulus space for the two-pixel example. The \( x \)- and \( y \)-axes of the figure correspond to the grayscale values for the first and second pixels, respectively. Targets A and B are shown in blue and red, respectively. As described previously, 1,000 stimuli were generated, 500 from each target.

To determine whether the GMM properly recovers features from data, we created a benchmark data set in which the correct features are known. To create data sets with known features, we simulated an observer with the feature structure as shown in **Figure 4** (in the column labeled “features”) and represented in **Figure 3**. Notice that, although the Category A and B features are better predictors of Targets A and B, respectively, these features do not perfectly match the targets. Recall that it is not necessary to assume that all image pixels are relevant to each feature; a feature could match only part of one target. The single feature for Category A, for example, ignores Pixel 1 (as represented by the “x” in **Figure 4** and by the blue horizontal line in **Figure 3**).

*Figure 3. Results from a two-pixel simulation with 1,000 stimuli generated from the two targets and classified by the observer model using the indicated features. Blue and red correspond to Categories A and B, respectively.*

A model of human decision making, illustrated in **Figure 4**, was used to generate simulated data for this task. For each trial, the noisy test stimulus is compared to each of the features. The likelihood ratio, \( L(F_1) \), that each feature, \( F_1 \), is present in the stimulus (relative to random noise) is calculated. Using a Bayesian-inspired decision rule in which the likelihood ratio of each feature is weighted based on how well the feature discriminates the two targets, these likelihood ratios are then combined within each response category (\( L(A) \) and \( L(B) \) for Response Categories A and B, respectively). More weight...
is given to more diagnostic features. The observer selects the response with the greatest combined likelihood ratio. Some features may be sensitive only to a part of a target, as is the case for the Category A feature, which ignores Pixel 1 and matches a stimulus based solely on Pixel 2. Details of this decision model for the simulated observer are given in the Appendix.

This observer model was used to classify the two-pixel stimuli into one of two classes: A or B. The classifications of all 1,000 stimuli are shown in the top panel of Figure 3 as blue and red dots. (Because the contrasts have been adjusted so that performance is near 71%, the targets appear more similar than they did in Figure 1.) It is clear that there are no obvious modes near the features. Indeed, when the GMM was applied to these data (one and two features when applied to Categories A and B, respectively), the positions of the recovered features were clearly biased away from the actual features and were distributed to capture the range of classified data (see Figure 3).

It could be argued, however, that the GMM did capture the general pattern present in the features; the recovered B features, for example, are darker on Pixel 1 than the recovered A feature and one is lighter than the other on Pixel 2. This result, however, is mostly an accident of the restricted two-pixel stimulus space. In contrast, consider the simulation illustrated in Figure 5. The data were generated as before, except that the Pixel 2 value of the second B feature was shifted toward neutral gray. Because the feature shift did not substantially change the classification boundary (top panel), the features recovered by the GMM did not substantially shift. These simulations illustrate that when all of the data are used, the GMM tends to find modes that cover the classified data, without direct regard for the features.

Feature recovery would be far more robust if there were clear modes near the features. One way to identify stimuli that are close to a feature (and far from other features) is to utilize only those trials on which the likelihood of one of the responses was far greater than that of the other response. This is the equivalent of asking observers for confidence ratings and using only high-confidence responses for analysis. Such a confidence measure was generated in the following manner. For each trial, the probability that the model would respond “A” was calculated as $P(A) = \frac{L(A)}{L(A) + L(B)}$, and likewise, for “B”, $P(B) = \frac{L(B)}{L(A) + L(B)} = 1 - P(A)$.
Assuming the features are sufficiently distinct, high-confidence stimuli tend to be produced by high activation of a single feature.

For the first two-pixel simulation, the stimuli associated with a high-confidence response (the top 15% for each response class) are shown in the bottom panel of Figure 3. When only these high-confidence trials were analyzed, clear modes were produced near the features, allowing the GMM to recover features that were quite close to the generating features.

There are a number of factors that might influence the ability of the GMM to recover a feature. It is perhaps most obvious that the features should be sufficiently distinct from each other. If features are too similar, then the modes near them will merge, rendering them unrecoverable. The B modes in the bottom panel of Figure 5, for example, have merged, and thus, the two GMM-recovered features are identical. Real observers, however, are unlikely to select extremely similar features for a given task. If features are very similar, then it will also be difficult to determine the number of features that best fit the data—two close features might produce the same classifications as one feature near their mean.

The problem of identifying the number of feature components is further exacerbated if the GMM model is generalized. For simplicity, we have utilized a GMM model having Gaussian components with a diagonal covariance matrix and equal variance on each dimension. If the variances were allowed to differ, a single Gaussian component (an ellipse with a long axis parallel to the horizontal axis) could well represent the data points in the upper panels of Figures 3 and 5. Investigation of the more general GMM model is a goal for future work.

It is interesting to consider cases where the observer selects features for a given task that are systematic distortions of the targets relative to the stimulus space. The top panel of Figure 6 shows such a case for high-confidence responses. The chosen features are sufficient to do an excellent job of classifying, but because they are effectively outside the range of stimuli, the recovered features are systematically biased inwards. The opposite situation is shown in the lower panel. Now, the features are very similar to the targets and, more importantly, to each other. Stimuli that are similar to features from the opposite class cannot produce high-confidence responses. Hence, the recovered features move outward from the actual ones. These results occur because two forces compete to produce high-confidence responses: The stimuli must be close to a feature from one class and far from the features from other classes.

Because analysis of the entire stimulus (target plus noise) tends to recover the targets and not the characteristics of the observer’s classification strategy, classification images are created from the noise fields (i.e., the stimuli with the targets subtracted). To make the relationship between the generating and recovered features clear, the results in this section are analyses of the signal plus noise data. The results do not differ from those obtained by applying GMM to the noise fields. This convenient outcome is not typical. In general, the presence of the targets in the data sets will distort the recovered features.

A simple example will make this clear. Consider a three-pixel target with grayscale values (ranging from 0 to 1) of (.7, .6, .6). Assume that there is a feature with grayscale values of (.9, .9, x), where the “x” indicates that the value for this pixel is not used when matching this feature to a stimulus. Assume that this feature is far from any other features and also much more likely than the other features to match stimuli generated by the (.7, .6, .6) target. The cluster of high-confidence data for the (.7, .6, .6) target trials would tend to cluster around (.9, .9, .6), if the full stimuli are analyzed. The values for the first two pixels come from the feature; because the feature is insensitive to the third pixel, the value of that dimension comes from the target. Removing the target and fitting GMM to the noise fields would remove its influence on the third pixel (because the noise has a zero mean and is normally distributed, unused pixels tend to be 0). This example illustrates the benefits of analyzing the noise fields, that is, the target-plus-noise stimuli with the targets subtracted.

This example also suggests that, although using the noise fields from high-confidence data produces excellent results, a subtle bias from the target still remains. Consider the first two pixels of this three-pixel example. Although the first two pixels of the generating feature are
equal (.9), when the targets are subtracted, the expected recovered values are not (i.e., (.9, .9, .6) − (.7, .6, .6) = (.2, .3, .0)). Fortunately, this bias will only be extreme in the unlikely circumstance that observers are using features that are very different from the targets (e.g., if the feature were (.9, .1, x)). A subtler model of feature induction would take this negative bias into account. Because the bias is related to the subtraction of targets from the stimuli, explicitly modeling the case in which a feature ignores some pixels could solve the problem. The GMM model does not currently do this but could be extended to handle ignored pixels in the future.

In summary, the GMM will do an excellent job of recovering features that are well separated, frequently activated by the stimuli, and relatively close to their associated targets.

**More complicated features**

To demonstrate the applicability of the technique in more interesting settings, we performed simulations to see if the GMM could recover features in a more challenging situation. Assume two targets, A and B, created from a 4×4 grid, as depicted in Figure 7. As in the two-pixel simulations, we do not assume that the observer uses these targets as feature templates. Instead, the observer utilizes the features in Figure 8a to determine whether the test stimulus (target plus noise) is Target A or B. Each of these features is compared with the test stimulus on a trial. The top three features are more similar to Target A and, to the extent that they match the test stimulus, provide evidence favoring the presence of Target A. The bottom two features are more similar to Target B and, to the extent that they match the test stimulus, provide evidence favoring the presence of Target B. The areas marked with × are regions of the stimulus that are ignored by that feature. These features were selected to have certain properties that might be true of human features in more complex task settings: The features overlap with each other both within and across response categories, do not perfectly correspond to either target, and include different numbers of relevant pixels, and more features are similar to one target than the other.

We simulated an experiment using the targets in Figure 7, the features in Figure 8a, and the decision/observer model outlined in Figure 4. Thus, all stimuli were 4×4 grids of grayscale values independently selected from a Gaussian distribution and added to low-contrast versions of either Target A or B. The contrast was set so that overall performance was approximately 71% correct. There were 100,000 trials that were simulated, half with each target.

As before, the GMM was applied separately to the classification data of the two response classes. The algorithm had to induce the features given access only to the noise fields in each response class. We show results when the GMM with three clusters is applied to the A response trials and when the GMM with two clusters is applied to the B response trials. For the reasons discussed above, only high-confidence data were used. Only the trials with confidence values in the top 15%, which corresponds to 15,002 (8,104 for A and 6,898 for B) of the 100,000 original trials, were used. Similar results were achieved in a different simulation using as few as 600 randomly selected high-confidence trials. The GMM recovered the features shown in Figure 8b. To highlight patterns in the data, the figures (and all remaining figures) have also been scaled so that the darkest pixel value across each response class’s recovered features is black and the lightest is white. Comparison with Figure 8a shows excellent feature recovery. The log likelihoods for the best-fitting features for this and all further simulations in this section are given in Table 1.

These results were obtained using the correct number of features per response category. When too few clusters

![Figure 7](https://jov.arvojournals.org/pdfsaccess.ashx?url=/data/journals/jov/932847/...)

**Figure 7.** The two targets for the simulated experiment with a more complicated feature structure.

![Figure 8](https://jov.arvojournals.org/pdfsaccess.ashx?url=/data/journals/jov/932847/...)

**Figure 8.** (a) The features used for the simulated observer. The areas marked with × are not included in the features. (b) The features recovered by the GMM when applied only to the high-confidence responses. The top and bottom rows of each of the panels correspond to the features for Targets A and B, respectively.
were used, multiple features tended to be combined in a single cluster. Referring back to the two-pixel example, if a single feature were recovered for the high-confidence data from Class B of Figure 3, this single feature would capture the most data if it were positioned between the two clusters, that is, a combination of the two features. When too many clusters were used, either features were repeated (as happened for the high-confidence data in Figure 5) or additional noisy features with no structure were recovered (i.e., the additional feature was capturing noise in the data).

As might be expected, analysis showed that, according to the observer model, a single feature tended to have a significantly higher likelihood ratio than the other features on the high-confidence trials. The feature that dominates, however, does so only in relation to the other features for the given task. It is important to note that the GMM did not have access to the observer model and assumed no particular decision model when carrying out its clustering. In this sense, the method can be seen as a natural extension of the classification image approach.

High-confidence trials may be more useful than low-confidence trials, but they certainly do not allow identification of the relevant features by inspection. Figure 9, for example, shows the first nine high-confidence noise fields for A response trials, and it would be exceedingly difficult to pick out the three relevant features.

We tested some of the ideas presented in the previous section on this more complicated feature structure. In Figure 10, each block of five squares is an attempt to recover features from a simulated experiment. Two attempts to recover features from the complete experimental stimuli, target and noise together, using the GMM (as described above) are displayed in the green area of Figure 10. In the row labeled “All data”, all 100,000 trials were fed to the algorithm. Note that the features are essentially absent from this recovery attempt. Instead, the targets are recovered. In the row labeled “High Conf.” only the high-confidence data were used in feature recovery. As previously demonstrated, using these data greatly improves feature recovery. Notice, that, because the targets and noise were used, the targets still contaminated the recovered features. Furthermore, unlike the two-pixel simulation example, the high-confidence recovery missed one feature entirely, apparently due to the influence of the targets in the stimuli.

The red area of Figure 10 shows attempts to recover features from the noise fields only (subtracting the targets from the stimuli). As before, using all trials produced hints of the features but relatively poor results. The high-confidence data, which demonstrates excellent feature recovery, are the induced features shown in Figure 8b. The targets do not appear when only the noise fields are used.

Using the noise fields from high-confidence data produces excellent results, but, as discussed previously, the subtle bias from the target still remains. For example, look at the vertical-line feature recovered for Target B (a larger version is shown in Figure 8b). Note that, in this recovered feature, one pixel has a higher contrast than the others. Compare that with the Target B in Figure 7. It is exactly this pixel that was not present in the target. A similar effect can be seen in the other features.

Almost all of the findings from the two-pixel simulations generalized to this more complicated feature-induction attempt. Using all data produces poor results; thus, use high-confidence data. Features recovered from targets and noise can be contaminated by the targets. Utilizing only the noise fields is better, but the targets can still produce a negative bias in the recovered features.

Given these results, this simulation reveals the possibility of recovering relatively complex features. This provides sufficient reason to explore the use of the GMM approach with real observers and more realistic discrimination tasks. The results also point toward fruitful

Table 1. Highest log-likelihood values of the recovered features from the more complicated feature structure.

<table>
<thead>
<tr>
<th>Targets included?</th>
<th>High confidence only?</th>
<th>Category</th>
<th>ln(L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>A</td>
<td>34,759</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>30,961</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>A</td>
<td>185,655</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>164,865</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>A</td>
<td>55,602</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>30,058</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>A</td>
<td>176,828</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>157,883</td>
</tr>
</tbody>
</table>

Figure 9. The first nine of the high-confidence, Response Class A noise fields analyzed by the GMM to induce the features in Figure 8b.
extensions of the GMM model, particularly in handling ignored pixels and allowing different variances on each template pixel.

**The four-square task**

We turn next to a simple classification task, in which the data from both simulated and real observers are analyzed so as to demonstrate the further applicability of the GMM to inducing features.

As depicted in Figure 11, the observer is shown a white square (the “target”) at one of four possible locations, randomly selected on each trial. Gaussian noise is added to each pixel in each of the four locations, with each location divided into a $4 \times 4$ grid of pixels as shown in Figure 12. The observer’s task is to indicate whether the white square target appeared above or below fixation (the center of the four possible target locations). Thus, in this task, there are two categories (Top or Bottom), each with two members (Left or Right). The contrast of the white square target is again placed at a level where performance is at threshold (e.g., 71% correct). The data are the noisy test images (e.g., Figure 12) and observer’s responses (Top or Bottom) for every trial. Four subjects contributed 4,000 trials of data each to the experiment. This experiment was previously reported in Gold, Cohen, and Shiffrin (2006), for different analytic purposes. See that article for further methodological details.

As previously, benchmark data were created with known features. To create data sets with known features, we simulated three observers (using the same observer model) with the three different feature structures as shown in Figure 13. Observer 1 has two features per response category: Top–Left and Top–Right for Top and Bottom–Left and Bottom–Right for Bottom. Observer 2 simply has one feature per category: Top–Left–and–Right for Top
and Bottom–Left–and–Right for Bottom. To ensure that features that do not directly overlap with the target structure can also be recovered, Observer 3 has two features per response category that do not directly correspond to the target structure. The observer model was again used to simulate data using these known feature structures, with the contrast of the targets set so the simulated performance was approximately 71% correct.

The GMM was first applied to the top 15% high-confidence noise fields from the simulated observers, separately for trials given “Top” and “Bottom” responses. The GMM was run with both one and two clusters on the “Top” and “Bottom” response trials separately (for a total of two and four features, respectively). The log likelihoods for these fits are given in Table 2.

The best-fitting features recovered by GMM for the two simulated observers are given in Figure 14 (with the space between the four target square locations removed). The insets give the features as returned by the GMM. The features in the larger squares have been smoothed by replacing all of the values within each square region with the mean value across pixels for Observers 1 and 2 and within the white and nonwhite regions for Observer 3. Note that the correct features were recovered for each of the three observers.

Both the two- and four-feature GMMs were then applied to the data from the four human observers in the same manner (the log likelihoods of the recovered features are given in Table 2). Observers were not asked for confidence measures in this task. Although asking for confidence judgments is preferable, as a stand-in, reaction times were used as a measure of confidence: The faster the reaction time, the higher the confidence (e.g., Petrusic & Baranski, 2003). The 15% fastest trials were used for each response class.

Because the correct number of features is not known a priori, it is important to use a well-tuned method for comparing goodness-of-fit, if one wishes to choose one set of recovered features over another. Simple methods like the Akaike Information Criterion (Akaike, 1973), a technique used for comparing models with different numbers of parameters, assign far too large a complexity penalty to models with more features (each pixel in a feature counts as a parameter). The best way to choose the “correct” number of features has long been a thorny problem in many related domains, with no truly general and accepted solutions. We could imagine using Bayesian model selection (Kass & Raftery, 1995), giving little prior weight to features that we know are unlikely and unacceptable (such as a particular pattern of seemingly random pixel noise). This approach would tend to reduce

![Figure 13](https://jov.arvojournals.org/pdfaccess.ashx?url=/data/journals/jov/932847/)
the complexity penalty associated with the many pixels for each additional feature. However, the way to choose such priors is unclear. Choosing adequate priors seems to require us to know in advance what features are acceptable, when the main point of the procedure we present is feature discovery. Therefore, in this article, we do not recommend a method for choosing the “best” number of features. We recommend trying several numbers and using the internal consistency of recovered features, if any, to select likely features that can then be tested in further experiments.

Figure 15 shows the results from the GMM analysis for both the two- and four-feature models for each participant (other feature numbers could, of course, be tried as well). The results indicate that there was variation across participants in the numbers of features that they used to perform the task. Participant 1’s results are consistent with the use of two features for each category; Participant 2’s results are consistent with the use of two features for Bottom and one (or perhaps two) features for Top; Participants 3’s results are consistent with one feature for Top and one (or possibly two) features for Bottom; and the results of Participant 4 are consistent with the use of just one feature in each category. Gold et al. (2006), using other methods, suggested that each of these observers utilized four features. Unlike in Gold et al., however, the GMM was applied to the data for each participant individually rather than to a collapsed data set from all participants.

If the distributions of processing times are identical for each feature, using reaction time is probably a very good and convenient measure of confidence. There is a potential concern that, if these distributions do differ, using the fastest reaction times might bias recovery of the most easily processed features. This concern is genuine and is the main reason we recommend collecting confidence judgments. The recovered features for Participants 3 and 4 may have been biased in this way. However, because multiple features in a response class were recovered for at least two of the four observers in this task, the current data suggest that the reaction-time distributions of the features were sufficiently mixed.

**Discussion**

This work provides an important proof of concept: Features that are not directly observable but are used by observers to classify images can be induced from experimental classification data. Using the GMM, we were able to induce features from both simulated and human data in perceptually simple but computationally challenging experiments.

This research remains in a very early stage of development, and many puzzles and issues remain. We intend to explore (a) more interesting and more complex discrimination tasks (especially with real observers); (b) the use of knowledge of the visual system to put further constraints on features, for example, a feature may be constrained to be an edge detector; (c) generalizations of the GMM that allow both response classes to be fit simultaneously (e.g., Ross & Zemel, 2006); (d) the use of human judgment and prior scientific research to improve the efficiency of the algorithm by providing “good guesses” as to the features; (e) the possibility of developing hierarchical models (e.g., Frey & Jojic, 2003) that allow features to be much more abstract and invariant under a class of transformations (such as scaling, rotation, etc.); and (f) other types of computational algorithms.
Figure 15. The top and bottom of each pair of rows respectively show the four and two features recovered by the GMM from the four human observers of the four-square task.
networks, and ICA-based algorithms).

Appendix A

The observer model

All images are identically sized two-dimensional patterns of gray-value pixels ranging from 0 to 1, where 0 is black and 1 is white. Let the targets be the images that the observer is attempting to detect or discriminate. Because there may be any number of response categories and any number of targets in each response category, this version of the model is more general than the one used in the text. Let $X_{mk}$ be the $m$th target from response category $K$. Let $X_{mk}^{i,j}$ be the $(i, j)$th gray-level pixel value from $X_{mk}$.

Test images are the images to be classified by the observer into one of the $K$ response categories. Let $\tilde{Y}_n$ be the $n$th test image. Let $\tilde{Y}_n^{i,j}$ be the $(i, j)$th gray-level pixel value from test image $\tilde{Y}_n$. The tilde indicates that the image has had noise added to it. Let $Y_n$ be the original image from which $\tilde{Y}_n$ is derived. Let

$$\tilde{Y}_n^{i,j} = Y_n^{i,j} + N(0, \sigma), \quad (A1)$$

where $\sigma < .25$ and $N$ is distributed as a Gaussian with mean 0 and SD $\sigma$. To ensure that $\tilde{Y}_n^{i,j} \in [0, 1]$, first, identically adjust the image contrast for all images to restrict the pixel values to lie in the range $[0 + 2 \cdot \sigma, 1 - 2 \cdot \sigma]$, and second, resample if $N(0, \sigma) \notin (-2\sigma, 2\sigma)$.

Given all $X_{mk}$ and $\tilde{Y}_n$, the observer’s task is to determine to which target category, $K$, $\tilde{Y}_n$ belongs. Assume that the classification is based on the detection of features that may or may not be present in $X_{mk}$ and $\tilde{Y}_n$. Let $F_i$ be feature $i$. Let $F_i^{j}$ be the $(i, j)$th gray-level pixel value of feature $i$. Each $F_i$ is the same size as $X_{mk}$ and $\tilde{Y}_n$. Each $F_i^{j}$ contributes differently to the detection of $F_i$. Let $w_{i,j}^{j}$ be the weight of pixel $(i, j)$ in feature $i$, where $0 \leq w_{i,j}^{j}$ and $w_{i,j}^{j} = 0$ means that $F_i^{j}$ does not contribute to $F_i$. For simplicity, we may want to assume that $w_{i,j}^{j}$ is 0 or 1, but it is not necessary. Note that as currently implemented, there is no advantage to having both $i$ and $j$ indices. They were both included because later we may want to include constraints such as spatial contiguity of feature pixels. For simplicity, it is assumed that there is no internal noise in the detection of features.

Let the almost likelihood

$$L_{F_i \tilde{Y}_n} = \prod_{i,j} \left[ \frac{N(F_i^{j} - \tilde{Y}_n^{i,j} | 0, \sigma)}{(1 + 2 \cdot \sigma)} \right]^{w_{i,j}^{j}}. \quad (A2)$$

be the probability that $Y_n$ contained $F_i$ divided by the probability that $F_i$ was produced from uniform random noise in a range defined by the observer’s internal estimate, $\sigma$, of $\sigma$. $N(x | 0, \sigma)$ is the value at $x$ of a normal probability density function with mean 0 and SD $\sigma$. Convert to a probability as

$$Q_{F_i \tilde{Y}_n} = \frac{L_{F_i \tilde{Y}_n}}{1 + L_{F_i \tilde{Y}_n}}. \quad (A3)$$

For the special case with only two categories ($K$ and $J$) of one exemplar each described in the text, the relative degree of match of a feature to category $K$ is given by

$$D_{F_i} = \frac{\prod_{i,j} N(F_i^{j} - X_{mk}^{i,j} | 0, \nu)^{w_{i,j}^{j}}}{\prod_{i,j} N(F_i^{j} - X_{md}^{i,j} | 0, \nu)^{w_{i,j}^{j}}}, \quad (A4)$$

where $\nu$ is the observer’s comparison noise and $J$ is the contrast category. One potential generalization of Equation A4 to multiple categories and multiple exemplars per category can be generated by summing over all of the targets for each response category and then summing over each response category in the denominator.

The probability of choosing category $K$ can be given by

$$P(K \mid \tilde{Y}_n) = \frac{\prod_{i} (\alpha \cdot D_{F_i,K})^{\beta \cdot Q_{F_i \tilde{Y}_n}}}{\sum_{J} \prod_{i} (\alpha \cdot D_{F_i,J})^{\beta \cdot Q_{F_i \tilde{Y}_n}}}, \quad (A5)$$

where $\alpha$ and $\beta$ are parameters used to scale how evidence affects the decision and $J$ ranges over all categories. The probability given in Equation A5 was used as a stand-in for a measure of confidence. In the simulations reported above, category $K$ was selected if

$$P(K \mid \tilde{Y}_n) > P(J \mid \tilde{Y}_n) \forall J \neq K. \quad (A6)$$

In the simulations given above, $\sigma = .22$ in the two-pixel simulations, .235 in the four-square simulations, or .2239 for the more complicated feature structure; $\sigma = .6 \times \sigma$; $\nu = .15$; $\alpha = 1$; and $\beta = 1$.

Acknowledgments

This research was supported by NSF Grant SES-0631602 to A. L. Cohen, NEI Grant 1 R03 EY015787-01 to J. M. Gold, NIMH Grants 1 R01 MH12717 and
The authors would also like to thank Florin Cutzu, Arnab Dhua, Tom Griffiths, Adam Sanborn, Mark Steyvers, and Chen Yu for helpful discussions, information, ideas, and insights.

Commercial relationships: none.
Corresponding author: Andrew L. Cohen.
Email: acohen@psych.umass.edu.
Address: Department of Psychology, University of Massachusetts, Amherst, MA 01003-7710.

References


