A Gaussian Mixture Model Approach to Forecast Verification

VALLIAPPA LAKSHMANAN
Cooperative Institute of Mesoscale Meteorological Studies, University of Oklahoma, and National Oceanic and Atmospheric Administration/National Severe Storms Laboratory, Norman, Oklahoma

JOHN S. KAIN
National Oceanic and Atmospheric Administration/National Severe Storms Laboratory, Norman, Oklahoma

(Manuscript received 9 September 2009, in final form 4 February 2010)

ABSTRACT

Verification methods for high-resolution forecasts have been based either on filtering or on objects created by thresholding the images. The filtering methods do not easily permit the use of deformation while identifying objects based on thresholds can be problematic. In this paper, a new approach is introduced in which the observed and forecast fields are broken down into a mixture of Gaussians, and the parameters of the Gaussian mixture model fit are examined to identify translation, rotation, and scaling errors. The advantages of this method are discussed in terms of the traditional filtering or object-based methods and the resulting scores are interpreted on a standard verification dataset.

1. Introduction

Intuitively, approximating a gridded field by a Gaussian mixture model (GMM) may be thought of as the process of finding an optimal way to place Gaussian functions at various points in the image such that the sum of these Gaussians mimics the input gridded field. As shown in Fig. 1, the larger the number of Gaussian components in the mixture model, the more closely the image re-created using just the Gaussian components resembles the original image.

Given the GMM that approximates two images (the forecast and observed), we show in section 3 that it is possible to analyze the parameters of the component Gaussians to infer translation, rotation, and scaling transformations.

a. Relationship to verification approaches

The new methods of verifying model forecasts that have been proposed can be categorized into (a) filtering-based methods that operate on the neighborhood of pixels or on the basis of decomposition and (b) displacement methods that rely either on features or on field deformation (Gilleland et al. 2009). Here, we propose a method of verification that does not quite fall into either of these categories.

Our proposed method incorporates the level of detail, like the filtering methods, in that the approximation can be made as exact as desired by increasing the number of Gaussian components allowed in the mixture. The most exact representation would be a mixture of Gaussians of zero variance and a component centered at every grid point. However, our proposed method operates neither on the neighborhood of pixels nor on the basis of wavelet-like decompositions.

We propose analyzing the entire image (like field deformation), but only to find a parametric approximation to the image. Field deformation approaches such as those of Alexander et al. (1999) and Keil and Craig (2007) employ nonparametric optical flow approaches. In our approach, the parameters of the approximation are compared between the forecast and observed fields to obtain insight into the transformations (translation, rotation, and scaling) that would make the fields most like each other.

In its use of transformations, the method of this paper resembles the feature-based approaches of Davis et al. (2006) but without the dependence on thresholds (either in intensity or in size) to categorize “objects.” Therefore, our approach is not quite “object based.” It could,
however, be considered feature based if one were to extend the definition of “feature” to include the Gaussian components that form the mixture.

It should be noted that the GMM approach does require a threshold—only pixels with intensity above that threshold will be considered in the GMM fit. For the precipitation forecast fields in Fig. 1, only pixels with rainfall amounts greater than 6.6 mm, corresponding to the top 10% of the pixel values in the image, were used to fit the GMM, whereas for the synthetic fields in Fig. 2, only pixels with nonzero values were fit by the GMM. The difference between the GMM approach of this paper and the object-based approach of Davis et al. (2006) is that in the GMM approach, this threshold does not determine what the objects are. Thus, as is shown in Fig. 1, one could have either 3 features or 50 by choosing to fit all the pixels in the image above the 6.6-mm threshold to either a GMM with 3 components or to one with 50 components.

b. Advantages of the GMM approach

There are several advantages to fitting an image with a GMM and using the fitted GMM to carry out forecast verification:

(i) There is no need to be concerned with splits or merges—if two contiguous regions are better treated as a single region, then they will be approximated by a single Gaussian. Conversely, a single, contiguous region may be broken up into multiple Gaussians if needed for an optimal fit and if there are enough GMM components.

(ii) The Gaussian is a parametric function. Thus, the GMM affords a highly compressed view of the information in the data that is especially useful for comparing two images for correspondence.

(iii) The number of Gaussians used is a good measure of the level of detail at which the image is being represented. For the verification problem, by changing the number of Gaussians allowed in the mixture model, one can control the scale at which comparisons are carried out.

(iv) Transformations of Gaussians correspond to easily identifiable changes in their parameters. Translation of objects corresponds to a change in the center point of the Gaussian. Scaling (corresponding objects being smaller or larger in one of the fields) can be inferred by changes in the variance of the Gaussian. Rotation of objects can be inferred by changes in the ratio of the variance of the Gaussian in the east–west and north–south directions. Changes in the amplitude of the Gaussian correspond to changes in intensity.
The natural incorporation of the level of detail is an important characteristic of filtering-based methods. The natural incorporation of transformation is a key advantage of object-based verification methods, especially because the detection of transformation permits verification methods to avoid the “double penalty” (Gilleland et al. 2009) problem. Thus, a GMM provides the advantages of both of these methods within a simple, mathematically elegant framework that is also quite easy to implement.

The method by which a GMM is fit to forecast and observed fields is described in section 2. We present the results of comparing the GMM on fake geometric and perturbed cases drawn from Ahijevych et al. (2009) and Kain et al. (2008) and make suggestions for further work in section 3.

2. Fitting a GMM

Fitting a GMM to an image for the purposes of forecast verification consists of the following steps:

(i) initialize the GMM (section 2c),
(ii) carry out the expectation-minimization (EM) algorithm to iteratively “tune” the GMM (section 2b),
(iii) store the parameters of each Gaussian component of the GMM (section 2d), and
(iv) compute the translation, rotation, and scaling errors from the GMM parameters corresponding to the fits of the forecast and observed images (section 2e).

The strategy followed for initializing the GMM will be much more clear if it is preceded by a mathematical description of the GMM and of the EM algorithm. Hence, we define a GMM in section 2a and explain the EM algorithm that is used to fit the image to the GMM in section 2b before delving into the initialization strategy in section 2c and listing the parameters to be stored in section 2d. Error metrics are defined in section 2e and these are used to determine the corresponding Gaussians in section 2f.

a. The GMM

The GMM is defined as a weighted sum of $K$ two-dimensional Gaussians:

$$G(x, y) = \sum_{k=1}^{K} \pi_k f_k(x, y),$$  

(1)

where the amplitudes $\pi_k$ are usually chosen so that they sum to 1. Each of the two-dimensional Gaussians, $f_k(x, y)$, is defined given the parameters $\mu_x, \mu_y, \sigma_{xy}$ as (dropping the subscript $k$ for convenience)
\[ f(x, y) = \frac{1}{2\pi \sqrt{\det(\Sigma_{xy})}} e^{-\frac{1}{2}[(x-\mu_x)(y-\mu_y)]\Sigma_{xy}^{-1}[(x-\mu_x)(y-\mu_y)]}, \]

where \( \mu_x, \mu_y \) are the center of the Gaussian and \( \Sigma_{xy} \) is the variance of the Gaussian; that is, \( \Sigma_{xy} \) is a matrix whose components are

\[
\begin{pmatrix}
\sigma_x^2 & \sigma_{xy} \\
\sigma_{xy} & \sigma_y^2
\end{pmatrix},
\]

where \( \sigma_x \) is the standard deviation in the \( x \) direction and \( \sigma_{xy} \) is the covariance of \( x \) and \( y \). Here, \( \det(\Sigma_{xy}) \) is the determinant of the \( \Sigma_{xy} \) matrix. The scaling factor of the individual Gaussians \([1/(2\pi \sqrt{\det(\Sigma)})]\) has been chosen so that the Gaussians sum to 1 over all \( x, y \). If the \( \pi_k \)'s are chosen to sum to 1, then the GMM also sums to 1 over the entire image. This allows a probabilistic formulation that will be taken advantage of shortly.

### b. The expectation-minimization (EM) method

Given a set of points \( x_i, y_i \), it is possible to fit these points to a GMM, \( G(x, y) \), by following an iterative method known as the expectation-minimization method. The proof that this hill-climbing method works is available in many texts (e.g., Hand et al. 2001, 260–263), so we will limit ourselves to describing the actual technique as it applies to the problem of fitting a GMM to the set of points.

Assume that an initial choice of parameters \( \mu_{x_i}, \mu_{y_i}, \Sigma_{xy_i} \) exists for each of the \( K \) components. Because the scaling factors have been chosen to add up to one, the probability (or likelihood) that the point \( x_i, y_i \) is covered by the GMM given the set of parameters is given by

\[
P(x_i, y_i | \theta) = \sum_{k=1}^{K} \pi_k f_k(x_i, y_i | \mu_{x_k}, \mu_{y_k}, \Sigma_{xy_k}),
\]

where \( \theta \) is used as shorthand for all the parameters of all the \( K \) components.

The first step, known as the expectation step or E step, is to compute the likelihood of this given set of parameters. The probability that the pixel \( x_i, y_i \) arose from the \( k \)th Gaussian component is given by

\[
P(k|x_i, y_i, \theta) = \frac{\pi_k f_k(x_i, y_i | \mu_{x_k}, \mu_{y_k}, \Sigma_{xy_k})}{P(x_i, y_i | \theta)}.
\]

The second step, known as the minimization step or M step, is to update the parameters of all the \( K \) components based on the above likelihood calculations. To obtain the \( \mu_x, \mu_y, \Sigma_{xy} \) of the \( k \)th component, the points \( x_i, y_i \) are weighted by \( P_k(x_i, y_i) \) before the appropriate statistics are computed. For example,

\[
\mu_x = \frac{\sum_{i=1}^{N} [P_k(x_i, y_i)x_i]}{\sum_{i=1}^{N} P_k(x_i, y_i)}.
\]

Similarly, \( \mu_y \) is computed as \( E(y) \) and \( \Sigma_{xy} \) is computed as

\[
\begin{pmatrix}
E[(x - \mu_x)^2] & E[(x - \mu_x)(y - \mu_y)] \\
E[(x - \mu_x)(y - \mu_y)] & E[(y - \mu_y)^2]
\end{pmatrix}
\]

Finally, the amplitude \( \pi_k \) is computed as

\[
\pi_k = \frac{1}{N} \sum_{i=1}^{N} P_k(x_i, y_i).
\]

With the updated parameters, the E step is carried out, a new set of likelihoods is computed and used to weight the points in the next M step, and so on until convergence is reached. The convergence is tested on the total likelihood of all the points at the end of each M step as follows.

Recall that the probability that the point \( x_i, y_i \) is covered by the GMM given the set of parameters is given by \( P(x_i, y_i | \theta) \). From this, the probability that all the given points are covered by the GMM is given by the product of \( P(x_i, y_i; \theta) \) over all the points. To avoid numerical instability errors when multiplying so many small numbers, the log of this likelihood is computed instead:

\[
l(\theta) = \sum_{i=1}^{N} \log[P(x_i, y_i)].
\]

When the improvement in \( l(\theta) \) falls below some tolerance, the iterative EM process can be stopped. We stopped the EM process when the improvement fell below 1% and found that convergence happens in 5 to 10 iterations.

The entire GMM fitting process is computationally very inexpensive. Each iteration of this process consists simply of computing weights by summing up previously computed values [Eqs. (4) and (5)] and then computing weighted averages [Eqs. (6)–(8)]. We found that computing a 50-component GMM fit onto a 500 x 600 image took just 0.05 s on a 1-GHz processor.

### c. Initialization of the GMM

Recall that the E step requires a set of components, and the weights computed at the end of the E step are required to create a set of components in the M step. Thus, the EM process has to be bootstrapped with some
initial guess at a GMM. Then, the EM process will start at that point and slowly climb toward the local maximum in likelihood space. This problem, of only promising a local maximum, is a shortcoming of the EM method, but it is not a critical problem in the case of weather images because we can initialize the GMM near a “good enough” solution.

In the case of weather images, we do know that contiguous pixels “should” belong to the same Gaussian. We can take advantage of this spatial coherence to place the initial mixture components. The pixels in the image with valid data values are grouped into regions consisting of contiguous pixels. These pixels are then arranged so that all the pixels in a region are listed together. The carefully arranged list of pixels is broken into \( K \) equal parts, where \( K \) is the desired number of Gaussian components. Each pixel gets a weight of one for “its” Gaussian component and zero for all other components; that is, if a pixel falls into the \( k \)th group, the weight is one for the \( k \)th component and zero for all other components.

Thus, the initial condition consists of a number of Gaussian fits so that separate regions will tend to be fit to a Gaussian. Relatively large regions will be fit in parts to Gaussians. From this initial point, the hill-climbing approach of the EM method finds the best possible fit. However, because the EM method is only a local optimization method, there may be a better solution elsewhere but it may not be reached.

d. Parameters of the GMM

The GMM is completely specified by the following parameters: \( \pi, \mu_x, \mu_y, \sigma_x, \sigma_y \), and \( \sigma_{xy} \) for each of the \( K \) Gaussian components of the GMM. Recall, however, that the GMM was defined so as to sum to 1, and that the EM method optimized the likelihood of the parameters given the positions of the pixels (and not the intensity). Thus, two minor changes have to be made to the GMM procedure explained above:

(i) The total intensity associated with all the pixels in the image is stored and this value, \( A \), is used to scale the GMM so that the image intensities can be re-created; that is, the GMM equation is modified to be

\[
G(x, y) = A \sum_{k=1}^{K} \pi_k f_k(x, y).
\] (10)

(ii) Because the EM method does not cater to the intensity, the more intensive locations are repeated several times. This is done by creating a cumulative frequency distribution (CDF) of the pixel values in the image and using a pixel’s location \( m \) times, where \( m \) is given by

\[
m = 1 + \gamma \cdot \text{round} \left( \frac{\text{CDF}(I_{xy})}{\text{frequency}(I_{mode})} \right) \quad \forall \quad I_{xy} < I_{mode},
\] (11)

where \( I_{mode} \) is the intensity corresponding to the most frequent quantization interval in the histogram of intensities used to compute the CDF. Pixel locations with intensities lower than \( I_{mode} \) are used only once. It is apparent that if the correction factor, \( \gamma \), is zero, then pixels are not repeated and as \( \gamma \) is increased, higher intensity pixels are repeated more often. The results in this paper, unless explicitly stated otherwise, all use \( \gamma = 1 \).

The need for, and the effects of, this intensity correction can be illustrated by using the artificial dataset shown in Fig. 2. Without intensity correction (see Fig. 2b), the GMM fit simply tries to get all the nonzero pixel locations correct, and the resulting GMM fit is simply a symmetric ellipse. With low values of \( \gamma \) (see Fig. 2c), because there are many more low-intensity pixels than high-intensity pixels, the GMM fit is dragged only slightly toward the higher-intensity values. On the other hand, when the higher-intensity pixels are heavily emphasized (see Fig. 2e), there are many more high-intensity pixels in the fit and, therefore, several components of the GMM are expended toward getting the high-intensity locations correct. In this paper, we use the moderate value of \( \gamma = 1 \) because it appeared to work best on real precipitation forecast fields.

e. Error measures

Given two Gaussian components, one from the forecast field and one from the observed field, it is possible to compute the translation, rotation, and scaling errors from the parameters of the two components (how corresponding Gaussians are identified is described in section 2f).

The translation error, \( e_t \), is the Euclidean distance between their means:

\[
e_t = \sqrt{(\mu_{sf} - \mu_{so})^2 + (\mu_{sy} - \mu_{yo})^2},
\] (12)

where the subscripts \( f \) and \( o \) correspond to the forecast and observed fields, respectively.

The rotation error, \( e_{rot} \), can be computed from the two covariance matrices since the first eigenvector of a covariance matrix represents the direction of maximum variance (this is the key idea underlying principal components analysis, for example). Once the eigenvectors of the two covariance matrices are computed, the dot product of the eigenvectors yields the cosine of the angle between them. Hence, the rotation error (in degrees) can be computed as
where \( \mathbf{v}_f \) and \( \mathbf{v}_o \) are the maximum-variance eigenvectors of the covariance matrices (\( \Sigma \)) of the forecast and observed fields. As pointed out by Davis et al. (2006), however, one should be careful when using rotation error on objects that are circular. In the case of a GMM, the confidence associated with \( e_{\text{rot}} \) is low if \( \sigma_x \) and \( \sigma_y \) are nearly equal.

The scaling error, \( e_{\text{sc}} \), can be computed as

\[
e_{\text{sc}} = \frac{A_f \pi k_f}{A_o \pi k_o}, \tag{14}\]

so that if \( e_{\text{sc}} \) is less than one, it is an underforecast and if it is greater than one, it is an overforecast.

f. Finding corresponding Gaussians

All the error measures in the previous section are defined assuming that one Gaussian component from each field (forecast and observed) is given. In fact, there will be \( K \) Gaussian components available from each field. Therefore, these error measures are computed for each pair of Gaussian components (\( K^2 \) pairs in all) and the best match for each forecast component is selected by normalizing and weighting the three individual errors to compute an overall error. We chose the scaling factors and weights arbitrarily:

\[
e = 0.3 \times \min \left( \frac{e_{\text{sc}}}{100}, 1 \right) + 0.2 \times \min (e_{\text{rot}}, 180 - e_{\text{rot}})/90 + 0.5 \times \max (e_{\text{sc}}, 1/e_{\text{sc}}) - 1 \right]. \tag{15}\]

In practice, they would be chosen based on the resolution of the images and the needs of the users of the forecast. For example, under- and overforecasts may have different costs, as could translation errors beyond a certain threshold.

The overall forecast error is defined as the mean of the individual GMM component errors. Alternately, because the Gaussians are localized, the errors could be used as indicative of the errors in different regions of the forecast field.

g. Number of components

The initialization procedure assumed that we needed a GMM consisting of \( K \) components. How do we know the number of components needed in the GMM?

The traditional way of estimating \( K \) is to start with one model and slowly increase the number of models. At each \( K \), the log-likelihood obtained from the GMM fit is used to compute an information criterion such as the Bayes information criterion (BIC; Hand et al. 2001):

\[
\text{BIC} = 2l(\theta) - 6K \log(N). \tag{16}\]

The optimal value of \( K \) is the \( K \) at which the information criterion is maximum. In effect, the fitting is stopped when the number of parameters to represent the model (\( \mu_x, \mu_y, \sigma_x, \sigma_y, \sigma_{xy}, \) and \( \pi \) for each of the \( K \) Gaussian components) starts to overwhelm the advantage gained by the increased likelihood.

We found though that the maximum number of components given by this criterion is too large for the forecast verification problem. For example, for the image shown in Fig. 1, the number of components required before the BIC stops increasing is on the order of hundreds. Thus, we subjectively chose the maximum number of components to be three for all the cases considered.

3. Results, analysis, and conclusions

We computed the GMM on three datasets from a verification methods intercomparison project (Gilleland et al. 2009; Ahijevych et al. 2009) that was established to improve the understanding of the characteristics of various model forecast verification methods. The goal of the intercomparison project was to provide answers to questions such as how different verification methods provide information on location, intensity, and structure errors, as well as on model performance at different scales. To enable reasonable comparison, the verification methods were carried out on synthetic and real fields with known errors. The methods were also applied to a common dataset used in a subjective model evaluation experiment. The results of the GMM approach on the different datasets that were created by the intercomparison project are presented below.

a. Geometric

This dataset consists of a synthetic object that is subjected to geometric transformations. We carried out our GMM fitting assuming three components so as to keep the hand analysis of GMM parameters manageable. For consistency, we used the normal intensity correction (\( \gamma = 1 \)) that we employ on real-world datasets.

Even though these choices are nonideal for this synthetic object, the GMM approach does extremely well in identifying the translation, rotation, and scaling errors. The GMM fit shown in Fig. 3 is a poor approximation to the synthetic object. This is because the synthetic object is unrealistic in two specific ways. First, the synthetic object has abrupt transitions between intensity levels whereas Gaussian approximations are better suited to
more gradual variations. Second, the intensity (gamma) correction is done based on a cumulative distribution function. This works well on real-world images but does poorly on this synthetic image where the distribution function consists of just two values. Indeed, as shown in Fig. 2, it is possible to obtain a better approximation of the synthetic object by using many more components (to better approximate the high gradients) and a higher value of $\gamma$ (to better equalize the sparse intensity histogram).

By referring to Table 1, it may be observed that translation to the right, whether by 50 points as in geom001 or by 125 points as in geom005, is easily inferred by the change in the longitude direction of the appropriate number of pixels. Translation to the north or south can similarly be inferred from changes in $\mu_x$. Differences in size can be inferred quantitatively as changes in $\sigma_x$ or in the amplitude, $A\sigma_k$, as in geom004. Both numbers ($\sqrt{2110/128}$ and $167034/49734$) indicate that the region in geom003 is about 3 times too big. The wrong orientation in geom004 can be inferred from the changes in $\mu_x$ and $\sigma_x$. The new object is 4 times too small in the north–south direction and 4 times too large in the east–west direction. The translation by 125 pixels can be inferred by the change in $\mu_x$. Quantitatively, the rotation is captured by the $\epsilon_{\text{rot}}$ of 90°. When the objects become circular (as in geom003 and geom005), the rotation metric is unreliable but this is to be expected because the “orientation” of a circular object is undefined. Thus, the GMM is able to capture the transformations on this synthetic dataset (except for circular objects).

If we were to rank the different synthetic forecasts by the admittedly subjective weighted error metric of Eq. (15), the order is geom001, geom002, geom004, geom003, and finally geom005. This is intuitively what one would expect.

b. Perturbed

The “perturbed” set of cases from the Spatial Forecast Verification Methods Intercomparison Project (Ahijevych et al. 2009) consists of observed data from the 2005 National Severe Storms/Storm Prediction Center (NSSL/SPC) Spring Experiment described in Kain et al. (2008). The observed data were subjected to various transformations as shown in Fig. 4. We carried out the fit with three Gaussian components, as in the case of the synthetic cases, primarily to keep the hand analysis of the GMM parameter changes tractable. We used only the top 10% of the pixel values in each of the images to form the GMM fit so as to avoid contamination by the extremely large number of low-intensity pixels in this real-world image. This adaptive threshold was 6.6 mm on the original image and higher, due to the movement of pixels beyond the edge of the domain, for the perturbed images.

Here too, the GMM is able to capture the translations as shown in Table 2 for cases 1–3. Within the limits of round-off error, the differences in $\mu_x$ and $\mu_y$ match up well with the known translation errors (see also the first two columns in Fig. 4). In cases 4 and 5, the translations...
are larger. While the GMM fits and $e_t$ point to the magnitude of the translation error, the numerical estimates are inexact because many of the pixels that were in the original fit are now off the edges of the image. The dependence of the GMM fit on these boundary pixels can be derived analytically and is given by the partial derivative of the GMM equations with respect to $x$ and $y$. If pixels in the eastern part of the image are not included in the GMM fit, for example, the centroid moves to the west by an amount given by the partial derivative of Eq. (6) multiplied by the number of such pixels.

Case 6 involves both translation and an overestimation of the precipitation amounts; each pixel’s value is multiplied by 1.5. This overestimate is captured in the

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description</th>
<th>$\mu_y$</th>
<th>$\mu_x$</th>
<th>$\sigma_y^2$</th>
<th>$\sigma_x^2$</th>
<th>$\sigma_{xy}$</th>
<th>$A\pi_k$</th>
<th>$e_t$</th>
<th>$e_{tot}$</th>
<th>$e_{sc}$</th>
<th>$e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>geom000</td>
<td>Original</td>
<td>249</td>
<td>203</td>
<td>1720</td>
<td>4</td>
<td>128</td>
<td>49734</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>249</td>
<td>203</td>
<td>1667</td>
<td>4</td>
<td>127</td>
<td>49734</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>203</td>
<td>1668</td>
<td>9</td>
<td>127</td>
<td>49737</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>geom001</td>
<td>50 points to the right</td>
<td>249</td>
<td>253</td>
<td>1694</td>
<td>0</td>
<td>129</td>
<td>49731</td>
<td>50</td>
<td>0</td>
<td>1</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>254</td>
<td>1682</td>
<td>4</td>
<td>121</td>
<td>49741</td>
<td>51</td>
<td>0</td>
<td>1</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>253</td>
<td>1679</td>
<td>4</td>
<td>131</td>
<td>49732</td>
<td>50</td>
<td>0</td>
<td>1</td>
<td>0.15</td>
</tr>
<tr>
<td>geom002</td>
<td>200 points to the right</td>
<td>249</td>
<td>404</td>
<td>1612</td>
<td>4</td>
<td>126</td>
<td>49739</td>
<td>201</td>
<td>0</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>403</td>
<td>1682</td>
<td>4</td>
<td>127</td>
<td>49735</td>
<td>200</td>
<td>0</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>403</td>
<td>1760</td>
<td>0</td>
<td>129</td>
<td>49731</td>
<td>200</td>
<td>0</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>geom003</td>
<td>125 points to the right, too large</td>
<td>250</td>
<td>339</td>
<td>1696</td>
<td>9</td>
<td>2110</td>
<td>167034</td>
<td>136</td>
<td>91</td>
<td>3.36</td>
<td>1.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>249</td>
<td>340</td>
<td>1696</td>
<td>13</td>
<td>2048</td>
<td>167018</td>
<td>137</td>
<td>92</td>
<td>3.36</td>
<td>1.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>341</td>
<td>1647</td>
<td>4</td>
<td>2021</td>
<td>167032</td>
<td>138</td>
<td>91</td>
<td>3.36</td>
<td>1.68</td>
</tr>
<tr>
<td>geom004</td>
<td>125 points to the right, wrong orientation</td>
<td>249</td>
<td>341</td>
<td>104</td>
<td>1</td>
<td>2046</td>
<td>49736</td>
<td>138</td>
<td>90</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>249</td>
<td>340</td>
<td>101</td>
<td>1</td>
<td>2027</td>
<td>49729</td>
<td>137</td>
<td>90</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>339</td>
<td>105</td>
<td>2</td>
<td>2120</td>
<td>49740</td>
<td>136</td>
<td>90</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>geom005</td>
<td>125 points to the right, huge</td>
<td>249</td>
<td>355</td>
<td>1678</td>
<td>17</td>
<td>8271</td>
<td>323126</td>
<td>152</td>
<td>90</td>
<td>6.5</td>
<td>3.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>356</td>
<td>1688</td>
<td>34</td>
<td>8203</td>
<td>323125</td>
<td>153</td>
<td>90</td>
<td>6.5</td>
<td>3.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>356</td>
<td>1668</td>
<td>16</td>
<td>8265</td>
<td>323121</td>
<td>153</td>
<td>90</td>
<td>6.5</td>
<td>3.25</td>
</tr>
</tbody>
</table>

FIG. 4. (top) Perturbed images from Ahijevych et al. (2009) and (bottom) a GMM with three components.
amplitude ($A\pi_k$) of the Gaussian and in the scaling errors ($\epsilon_{sc}$s). Moreover, the translation effect is mostly independent of the amplitude effect, as can be noticed by comparing the $m_x$ and $m_y$ here with those of fake003. The translation error in fake006 is not identical to that of fake003 because formerly low-intensity pixels around the boundaries of a storm system were included in the GMM fit once their intensities are multiplied by 1.5.

Finally, fake007 involves both translation and a consistent underestimate of precipitation. This is reported by the GMM as a reduction in the amplitude and in the size ($s_x$ is smaller and $s_y$ larger but the net change is toward a smaller size). Note, for comparison, that fake006 showed an amplitude increase but no increase in size. Thus, the GMM is able to parsimoniously capture all the transformations on the perturbed dataset. The underforecast is captured in $\epsilon_{sc}$ but because the $e_{sc}$ was defined as a ratio, the reported error (e.g., 0.67) does not match up with the actual transformation, which was a constant underforecast of 2 mm.

Ranking the different perturbed forecasts by the error metric in Eq. (15) yields this order: fake001 (0.02), fake002 (0.04), fake003 (0.23), fake006 (0.31), fake004 (0.33), fake007 (0.42), and finally fake005 (0.44). Ordering forecasts in this manner is subjective, as the order would change depending on the weights assigned to the translation, rotation, and scaling errors and to the maximum tolerable errors in each category.

c. 1 June 2005

The third set of cases we analyzed consists of observed data and model runs from the 2005 NSSL/SPC Spring Experiment described in Kain et al. (2008). The observed data from 1 June 2005 are compared with 24-h forecasts of 1-h rainfall accumulation carried out on 31 May 2005. The GMM fits of the data and the model forecasts (from the 2CAPS, 4NCAR, and 4NCEP models) are shown in Fig. 5. The images cover the lower 48 states of the United States. The 4NCEP model forecast was produced at the National Centers for Environmental Prediction (NCEP) using a Weather Research and Forecasting (WRF) model whose core was a Nonhydrostatic Mesoscale Model (Janjic et al. 2005) with 4.5-km grid spacing and 35 vertical levels. The 4NCAR model forecast was produced at the National Center for Atmospheric Research using the Advanced Research WRF (ARW; Skamarock et al. (2005)) core with 4-km grid spacing and 35 vertical levels. The 2CAPS was produced at the Center for Analysis and Prediction of Storms at the University of Oklahoma (also using the ARW core) with 2-km grid spacing and 51 vertical levels. All three forecast systems used initial and lateral boundary conditions from the North American Model (Rogers et al. 2009). The observations are from the stage II rainfall accumulation dataset produced by NCEP (Baldwin and Mitchell 1998).
The 1 June case consists of three quite different systems: an elongated band stretching north–south in the middle of the image, somewhat weaker precipitation in the southeast, and weak, isolated storms in the northwest. As with the “fake” cases in the previous section, we carried out the fits with three Gaussian components for tractability and limited the fit to the top 10% of pixel values in each of the images. The three-component GMM fit does not capture these three events. Instead, two of the components correspond to the northern and southern sections of the elongated band and the southeastern band. The weak, isolated cells in the northwest are ignored in the GMM fit. As pointed out by Wernli et al. (2009), it would be advantageous to carry out this analysis on smaller domains where only one type of meteorological system predominates. It should also be noted, from Fig. 1, that higher-order GMM fits do capture all these systems. We chose to use only a third-order fit so as to keep the hand analysis of the component parameters tractable. An automated analysis employing more components is shown in Fig. 6.

The GMM coefficients are shown in Table 3. The GMM coefficients of the 2CAPS forecast (which is the same as the fake000 field in Table 2) are repeated for convenience.

The easy correspondence of the GMM parameters that existed in the geometric and perturbed cases does not exist in the real model forecasts. Nevertheless, interesting conclusions can be drawn from the transformations indicated by the changes in the GMM parameters. We will consider the Gaussian components one by one.

For the first Gaussian component (corresponding to the north-central part of the image), all three forecasts are displaced to the north and west. The 2CAPS forecast is the least displaced—its $\mu_x$ and $\mu_y$ are closest to that of the observation and $e_{\text{re}}$ is lowest. The 4NCAR model run underestimates the precipitation; the 2CAPS model run overestimates it, while the 4NCEP gets the intensity of precipitation nearly correct ($A\pi_k$ of 23 002 versus 22 136 or a $e_{\text{sc}}$ of 1.04). Examining the elements of the $\Sigma_{xy}$ matrix, the 2CAPS forecast gets the shape wrong, whereas the 4NCAR and 4NCEP forecasts get the extent correct in the north–south direction (the $x$ direction in our right-handed coordinate system centered at the top left of the image) but overestimate the east–west extent.

For the second Gaussian component (corresponding to the south-central part of the image), all three forecasts are displaced to the north, with the 2CAPS forecast again exhibiting the least displacement. The forecasts are extremely vertical (ratio of $\sigma_y$ to $\sigma_x$), whereas the observation indicates that the field should be more horizontal. The wrong orientation is captured in $e_{\text{rot}}$, although this error might be exaggerated because the
three-member GMM fit does not adequately capture the curvature in the line. In terms of intensity ($A_{\pi k}$ or $\varepsilon_{k}$), the 2CAPS is the closest whereas the 4NCAR and 4NCEP forecasts are significant overestimates.

On the third Gaussian component (covering the southeastern part of the image), the NCAR and NCEP model forecasts get the intensity and orientation correct but are displaced to the east. The 4NCEP forecast also exhibits a displacement to the north. In addition, the 4NCEP’s forecast is overly large in the north-south direction, indicating the precipitation, even if correct in the aggregate, is spread over too large an area.

Overall, the rank of the models, based on the subjective weighting used in Eq. (15), is 2CAPS (0.34), 4NCAR (0.49), and 4NCEP (0.50). At the extremely coarse scale at which the forecasts have been compared, the 2CAPS forecast exhibits the smallest translation, orientation, and scaling errors.

If we increase the number of Gaussians, it is possible to perform the comparison at finer detail. Recall that we used three components in this paper only so that we could do a hand analysis of the Gaussian components. Since even a 50-component GMM fit takes just 0.05 s to carry out, an automated analysis of errors can be carried out by varying the number of components from 1 to 50. This is shown in Fig. 6. The errors plotted in that graph are the translation, rotation, and scaling errors scaled according to Eq. (15); that is, the rotation error plotted there is

---

**Fig. 6.** Translation, rotation, and scaling errors for 24-h model forecasts of precipitation accumulation on 31 May 2005 indicate that the 2CAPS model run exhibits the least error and that the NCAR run is close to it in terms of performance, regardless of the number of components used in the GMM fit. The forecast fields themselves are shown in Fig. 5.
so that the errors can be averaged across components and plotted on a consistent (zero to one) y axis. Looking at the total error graph at the bottom right of Fig. 6, the relative rankings of the models are quite constant. The 4NCEP model exhibits the greatest errors while the 2CAPS one exhibits the least. The 4NCAR model is intermediate between these two, although at some scales (notably around 15 components), it does better than the 2CAPS model. These relative rankings are driven most strongly by the translation errors. In terms of rotation and scaling errors, the three models have comparable levels of performance. It is also clear that the error measures are quite robust to changes in the number of Gaussian components.

d. Areas for further exploration

This paper presents a GMM approach to model verification but is not a full-fledged verification technique. There are some unresolved questions about the GMM approach that need to be addressed in order to create a verification technique from the ideas in this paper:

(i) Association or deformation? In this paper, we approximated the observed and the forecast field by separate GMMs and picked out the correspondence of the parameters in the two GMMs by looking for the match with the lowest overall error. An alternative approach that would side step the entire association problem would be to start the EM on the forecast field with the GMM that corresponds to the observed field and observe how the GMM components get deformed. It is not known which approach is better.

(ii) Initialization of EM—The EM approach only promises convergence to a local minimum, not a global minimum. We introduced a bias toward the “known” form of the solution by organizing pixels into contiguous regions before computing the first E step. Exploration into other algorithms for initializing the EM process may prove beneficial.

(iii) Low-intensity regions—Because our GMM formulation was based on likelihood, we emphasized higher intensities by repeating the pixels at which higher intensities were present. This would have the unfortunate side effect of deemphasizing low-intensity and small cells if there is a large, high-intensity cell somewhere else. The intensity correction factor γ might depend on the verification problem.

(iv) Error measures—Other error measures are possible beyond the three (translation, rotation, and scaling) that were defined and employed in this paper. For example, an error metric based on size could be defined as

\[ e_{\text{size}} = \frac{\sigma_y \times \sigma_y - \sigma_x \times \sigma_y}{\sigma_x \times \sigma_y}. \]  

One possible solution to the problem of low-intensity regions might be to break up large spatial areas into smaller areas and then fit GMMs to them. The approach might be to fit a GMM to the entire image, then to break the image into quartiles and fit a GMM to each quartile. This process could be repeated as often as needed to create a hierarchical set of GMMs, each of which could be analyzed to obtain the forecast efficiency at the appropriate level of detail and over the appropriate spatial area. The drawback to this would be that the GMM representations would not be tied to storm morphology.

e. Summary

In this paper, we introduced the novel approach of using a Gaussian mixture model to verify model forecasts. We showed that the GMM approach is able to identify translation, rotation, and scaling errors in forecasts. We

\[ \min\left(e_{\text{rot}}, 180 - e_{\text{rot}}\right)/90, \]  

Table 3. GMM fits on observed and model forecasts from Kain et al. (2008) and the errors associated with the model forecasts.

<table>
<thead>
<tr>
<th>Description</th>
<th>(\mu_y)</th>
<th>(\mu_x)</th>
<th>(\sigma_y^2)</th>
<th>(\sigma_x^2)</th>
<th>(\sigma_{xy})</th>
<th>(A\pi_k)</th>
<th>(e_{tr})</th>
<th>(e_{\text{rot}})</th>
<th>(e_{\text{sc}})</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td>193</td>
<td>301</td>
<td>3546</td>
<td>841</td>
<td>936</td>
<td>22</td>
<td>136</td>
<td>20</td>
<td>21</td>
<td>1.19</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>264</td>
<td>684</td>
<td>1218</td>
<td>7508</td>
<td>22</td>
<td>616</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>383</td>
<td>309</td>
<td>921</td>
<td>2032</td>
<td>22181</td>
<td>20</td>
<td>061</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2CAPS forecast</td>
<td>176</td>
<td>289</td>
<td>1305</td>
<td>743</td>
<td>1328</td>
<td>26</td>
<td>437</td>
<td>43</td>
<td>1.17</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>309</td>
<td>252</td>
<td>1272</td>
<td>482</td>
<td>665</td>
<td>26</td>
<td>437</td>
<td>98</td>
<td>6</td>
<td>1.32</td>
</tr>
<tr>
<td></td>
<td>379</td>
<td>407</td>
<td>1456</td>
<td>3919</td>
<td>20490</td>
<td>26</td>
<td>437</td>
<td>73</td>
<td>6</td>
<td>1.73</td>
</tr>
<tr>
<td>4NCAR forecast</td>
<td>159</td>
<td>260</td>
<td>3134</td>
<td>2344</td>
<td>7636</td>
<td>16</td>
<td>464</td>
<td>53</td>
<td>1.05</td>
<td>0.34</td>
</tr>
<tr>
<td></td>
<td>277</td>
<td>264</td>
<td>3369</td>
<td>1607</td>
<td>932</td>
<td>39</td>
<td>139</td>
<td>73</td>
<td>6</td>
<td>1.73</td>
</tr>
<tr>
<td></td>
<td>379</td>
<td>461</td>
<td>1729</td>
<td>2840</td>
<td>14879</td>
<td>21</td>
<td>1068</td>
<td>152</td>
<td>6</td>
<td>1.05</td>
</tr>
<tr>
<td>4NCEP forecast</td>
<td>168</td>
<td>247</td>
<td>3518</td>
<td>747</td>
<td>6888</td>
<td>23</td>
<td>002</td>
<td>60</td>
<td>118</td>
<td>1.04</td>
</tr>
<tr>
<td></td>
<td>278</td>
<td>258</td>
<td>3153</td>
<td>906</td>
<td>484</td>
<td>43</td>
<td>675</td>
<td>72</td>
<td>117</td>
<td>1.93</td>
</tr>
<tr>
<td></td>
<td>405</td>
<td>416</td>
<td>3920</td>
<td>6740</td>
<td>28879</td>
<td>20</td>
<td>010</td>
<td>109</td>
<td>11</td>
<td>1.00</td>
</tr>
</tbody>
</table>

\[ e_{\text{size}} = \frac{\sigma_y \times \sigma_y - \sigma_x \times \sigma_y}{\sigma_x \times \sigma_y}. \]
also identified areas where this approach can be improved in order to create a robust verification method.

Acknowledgments. Funding for this research was provided under NOAA–OU Cooperative Agreement NA17RJ1227. We thank the anonymous reviewers for considerably strengthening this paper: in particular, Figs. 1f, 2, and 6 came about as responses to the reviewers’ questions and suggestions.

The GMM fitting technique described in this paper has been implemented within the Warning Decision Support System Integrated Information (WDSSII; Lakshmanan et al. 2007) as part of the w2smooth process. It is available for download online (www.wdssii.org).

REFERENCES
