6.891 Fall 1999: Problem Set Grading Guide

Note that there may be more than one way of obtaining a reasonable solution. Also, the decision boundaries may not exactly match those of the solutions since different training regimes will result in different classifiers. The important thing is to see that work was put into the solution and that the logic of the solution makes sense.

For each part of each problem assign one of the following grades:

- $\surd+$: got correct answer and properly justified solution (i.e., "nailed the problem and answer completely correct").
- $\surd$: showed significant work and insight but may not have completely vsolved the part, or the answer was slightly off (i.e., "got pretty much everything right except made a small error or left off some justification").
- $\surd-$: showed significant work and insight but had a some gaps in justification or answer (i.e., "had the right idea, but got lost somewhere or had a major error").
- 0 : did not show significant work or insight, or answer was not applicable to the question.

6.891 Fall 1999: Problem Set #4 Solutions

Problem 1: Preparing for the final project.

Nothing to turn in here.

Problem 2: Radial Basis Functions

The easiest way of insure that the network gives the same results is to insure that the basis functions return the same result. Thus, we want (taking $\tilde{x} = Ax + b$ to be the linear transformation)

$$\begin{aligned}
e^{-\frac{1}{2}(x-\mu_i)^T\Sigma^{-1}(x-\mu_i)} &= e^{-\frac{1}{2}(\tilde{x}-\tilde{\mu}_i)^T(\tilde{x}-\tilde{\mu}_i)} \\
(x-\mu_i)^T\Sigma^{-1}(x-\mu_i) &= (\tilde{x}-\tilde{\mu}_i)^T(\tilde{x}-\tilde{\mu}_i) \\
(x-\mu_i)^T\Sigma^{-1}(x-\mu_i) &= (Ax+b-\tilde{\mu}_i)^T(Ax+b-\tilde{\mu}_i) \\
x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_i - \mu_i^T\Sigma^{-1}\mu_i &= \begin{array}{l} x^TA^TAx + x^TA^Tb + b^TAx - \tilde{\mu}_i^TAx - x^TA^T\tilde{\mu}_i \\ +b^Tb - b^T\tilde{\mu}_i - \tilde{\mu}_i^Tb + \tilde{\mu}_i^T\tilde{\mu}_i \end{array}
\end{aligned}$$

If we now break the equality into quadric, linear, and constant terms of x and set those individually equal, we get that

$$\begin{aligned} x^T\Sigma^{-1}x &= x^TA^TAx \\ -x^T\Sigma^{-1}\mu_i - \mu_i^T\Sigma^{-1}x &= b^TAx + x^TA^Tb - \tilde{\mu}_i^TAx - x^TA^T\tilde{\mu}_i \\ \mu_i^T\Sigma^{-1}\mu_i &= b^Tb - b^T\tilde{\mu}_i - \tilde{\mu}_i^Tb + \tilde{\mu}_i^T\tilde{\mu}_i \end{aligned}$$

From the quadric terms, we note that $A^TA = \Sigma^{-1}$. If we plug this in for the linear terms, we get

$$-x^TA^TA\mu_i - \mu_i^TA^TAx = b^TAx + x^TA^Tb - \tilde{\mu}_i^TAx - x^TA^T\tilde{\mu}_i$$

From, this we can easily hypothesize that $b = 0$ and $\tilde{\mu}_i = A\mu_i$. Plugging in to the equations above, we can see these relationships do make the basis functions equal.

To summarize the solution, we find that $\tilde{x} = Ax$ and $\tilde{\mu}_i = A\mu_i$ where $A = \Sigma^{-\frac{1}{2}}$ (the definition of the solution to $A^TA = \Sigma^{-1}$).

Problem 3: Pixel Features

If we use a polynomial kernel for our classifier, this corresponds to having features of the form $x_{i_1}^{m_1}x_{i_2}^{m_2}\cdots x_{i_k}^{m_k}$. Since the image is binary, x_l can only be 0 or 1 and thus $x_l^m = x_l$ for all integer $m \neq 0$. So, we can drop the $m_1, m_2, \ldots, m_k$ and recognize that our features are simply the product of different pixels in the image. These features, therefore, are on when all of the pixels are on and off otherwise.

Thus, we have formed features which tell the classifier whether groups of pixels are all on together. If these pixels are arranged in a line, we have a line detector. If there are arranged in a circle, we have a circle detector. If they are arranged in a checkerboard pattern, we have a dithering detector of sorts. A thresholded sum of these features corresponds to asking whether a certain number of them are all on at the same time.

Problem 4: Let's start reading some papers.

A:

We will disambiguate outputs for multi-class classification the same way we did for multi-class perceptrons by picking the maximum activation energy. Classifications for 1 vs. 2 vs. 3 (no. errors = 20):

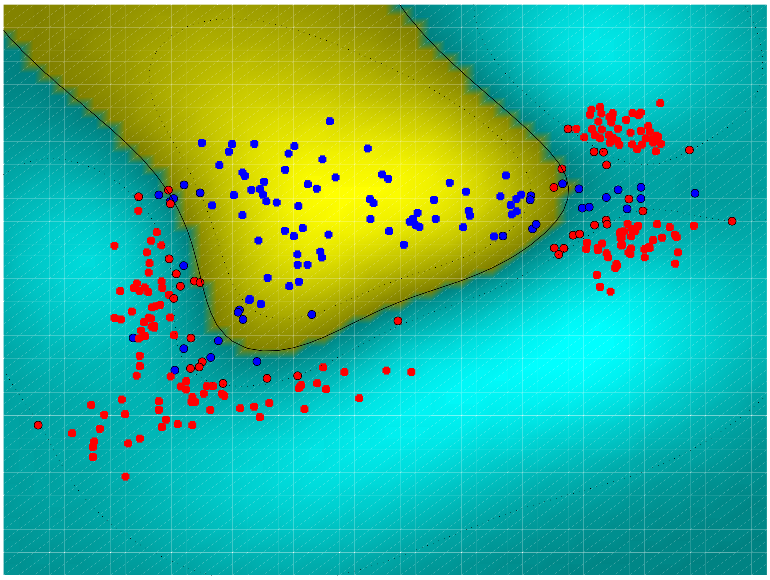

Figure 1: 1 vs. 2 and 3

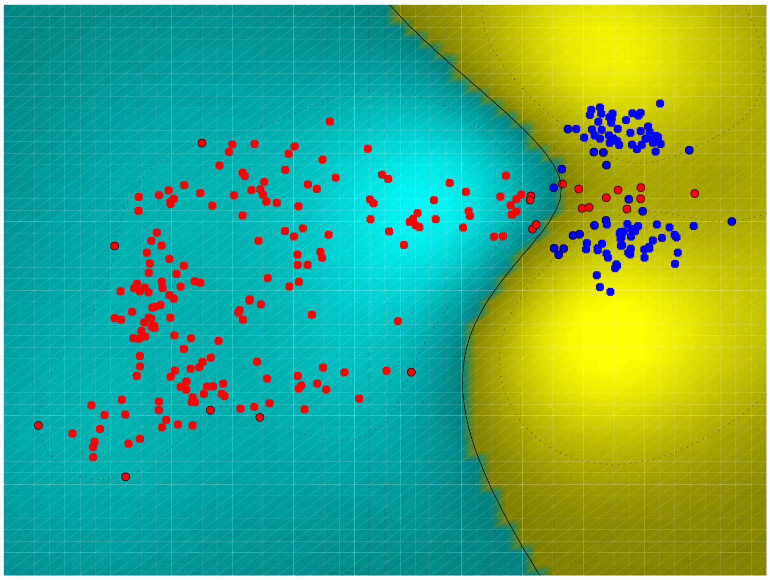

Figure 2: 2 vs. 1 and 3

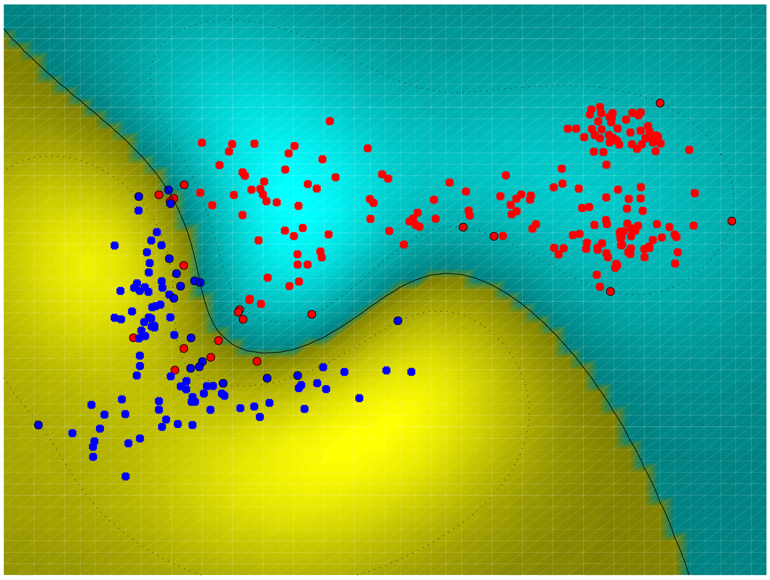

Figure 3: 3 vs. 1 and 2

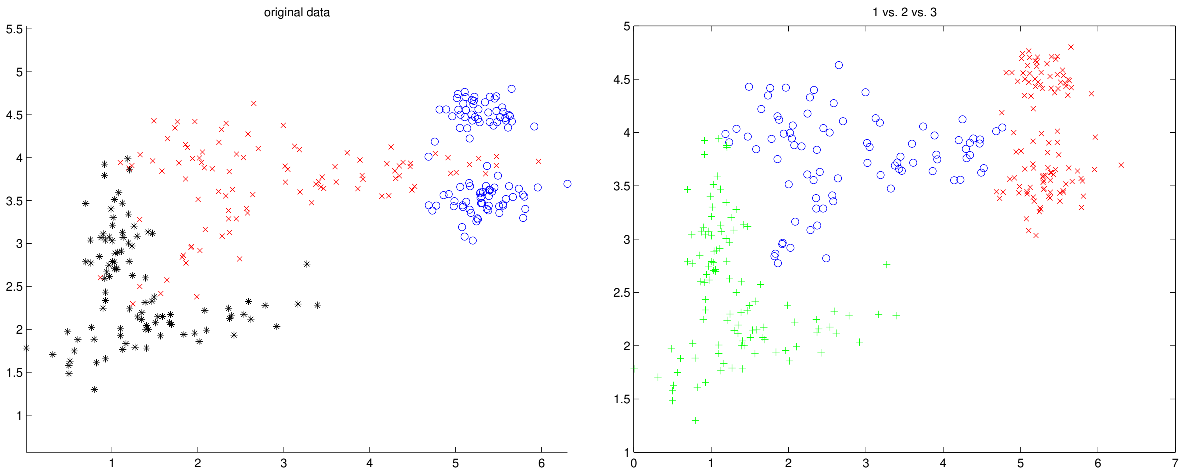

Figure 4: Original data and 1 vs. 2 vs. 3 classification.

B:

One way to train SVMs for multiclass problems directly is to reformulate the optimization problem. In particular we want to minimize:

$$\sum_{k=1}^{m} w_k^T w \tag{1}$$

subject to the constraints:

$$w_k^T x_{ki} - w_l^T x_{ki} \geq 1 \quad \forall k = 1, \ldots, m, l \neq k, i = 1, \ldots, n_k \tag{2}$$

where m is the number of classes and n_k is the number of training data in class k. In other words, the data should be separated with the maximum margin by choosing the maximum activation energy discriminant. This leads to a new Lagrangian where $(m-1)n$ $\alpha_i(k, l)$ parameters need to be simulataneously optimized.

Problem 5: Try out the Support Vector Code

There should be some indication (plots, numerical data) that the svm code was tried.

Problem 6: Hacking RBF's

In general, the more centers used, the better the reconstruction. However, for the case of the Gaussian, the ratio of the distance between the centers and the width of the Gaussian was important in the quality of the reconstructed function. If the width was too wide, then it was difficult to fit the function exactly to the data points and so the weights became very large thus causing the reconstruction to blow up in places where no data had been sampled. On the other hand, if the width was too small, the reconstructed function trailed off to zero too quickly between points thus yielding a set of bumps with no interpolation in between points.

The absolute value kernel doesn't seem to have these problems. However, the resulting functions are not smooth and tend towards infinity or negative infinity outside the range of the data.

K-means tended to do a better job than randomly selected points for picking fewer centers than data points, but these tests aren't conclusive.

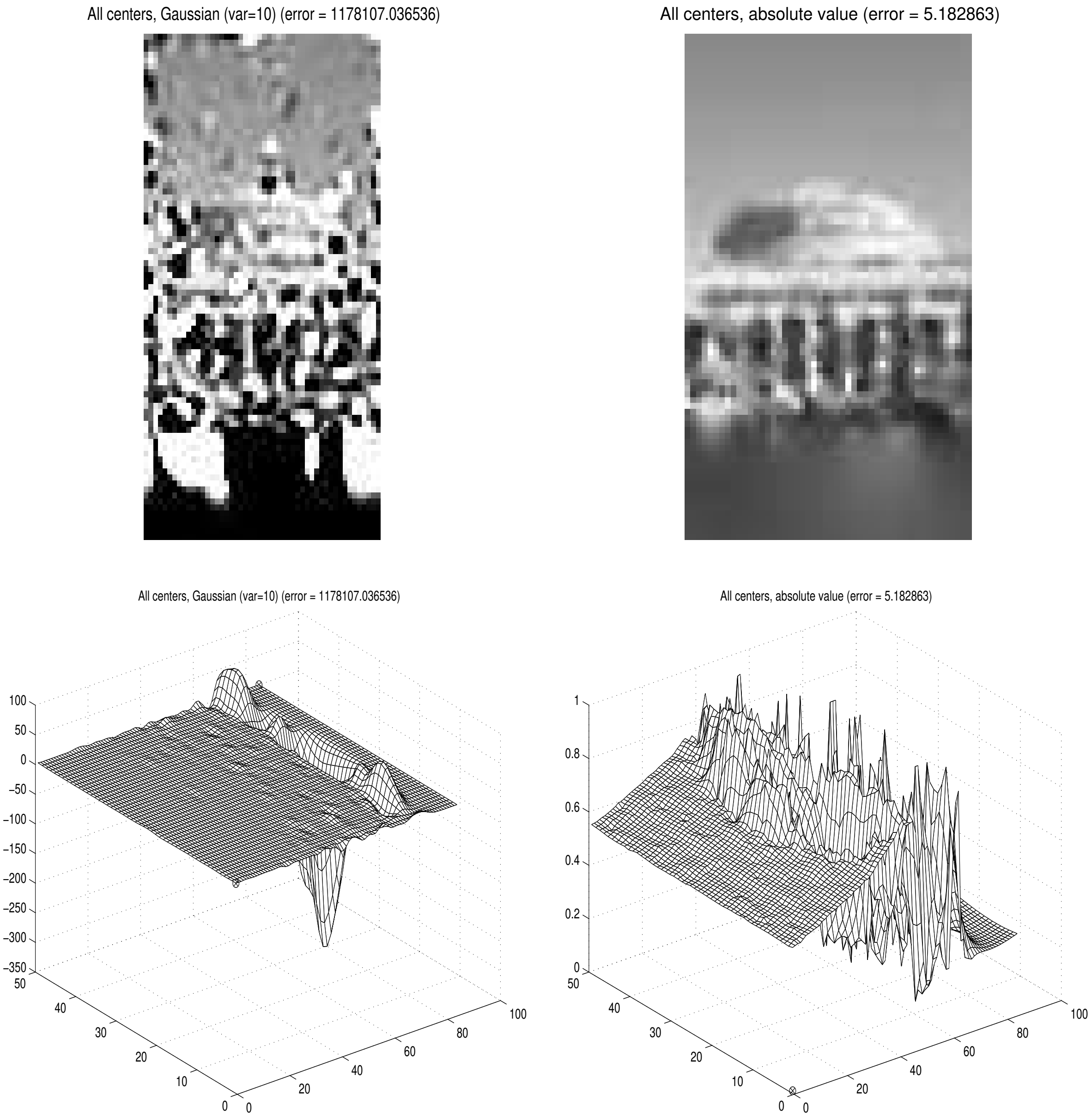

Figure 5: All centers, Gaussian Kernel ($\sigma^2 = 10$) and abs kernel

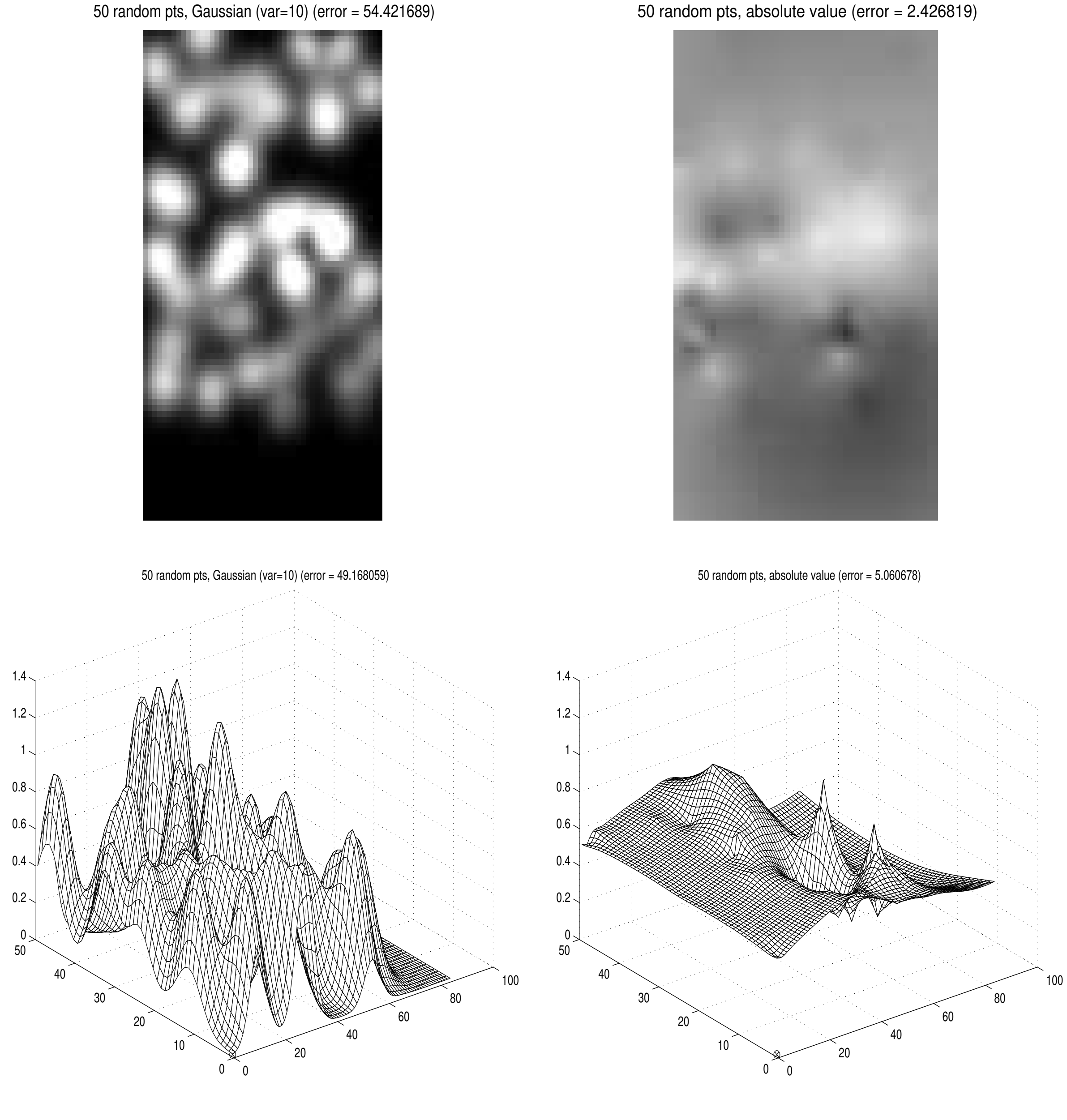

Figure 6: 50 random centers

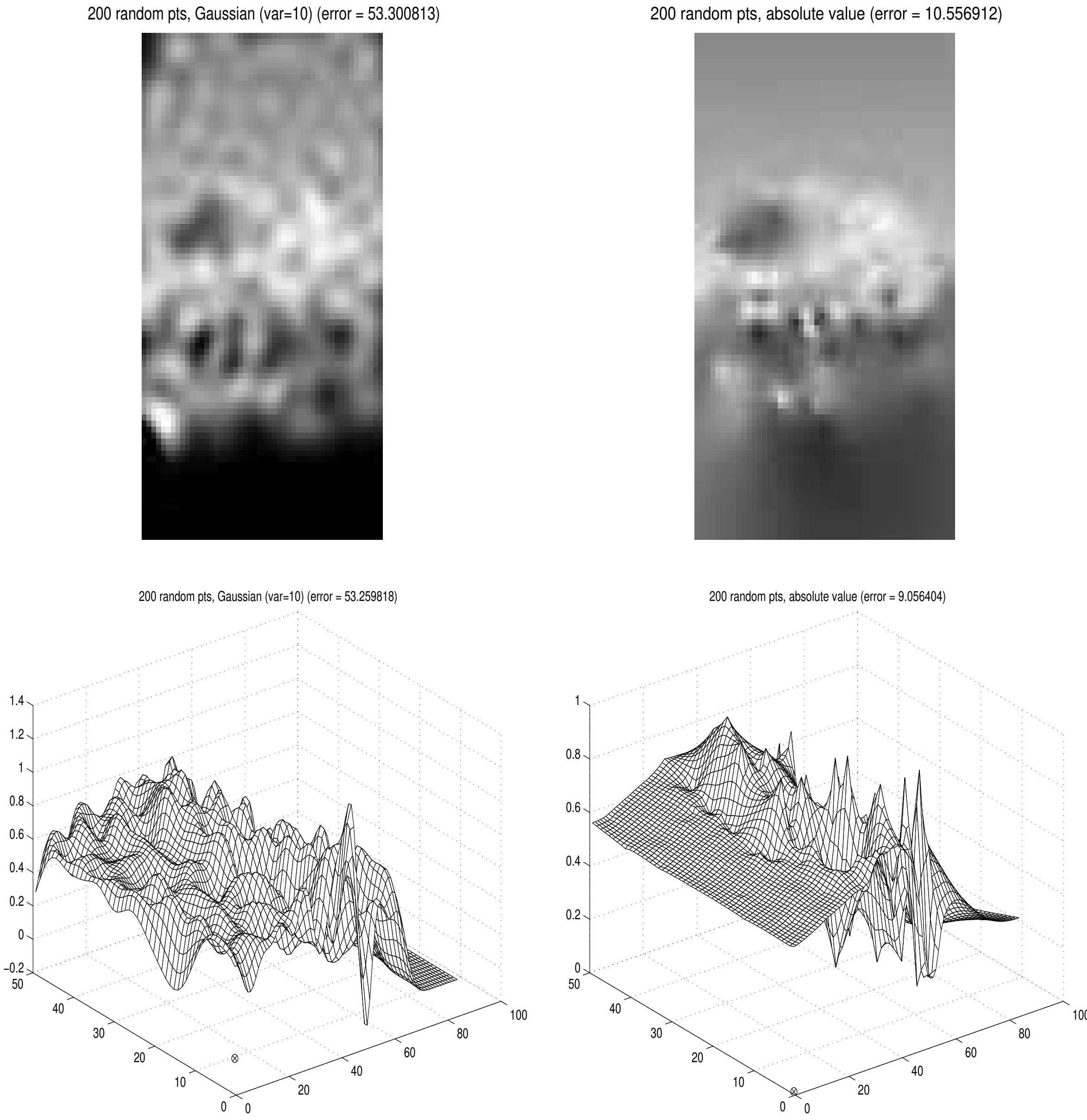

Figure 7: 200 random centers

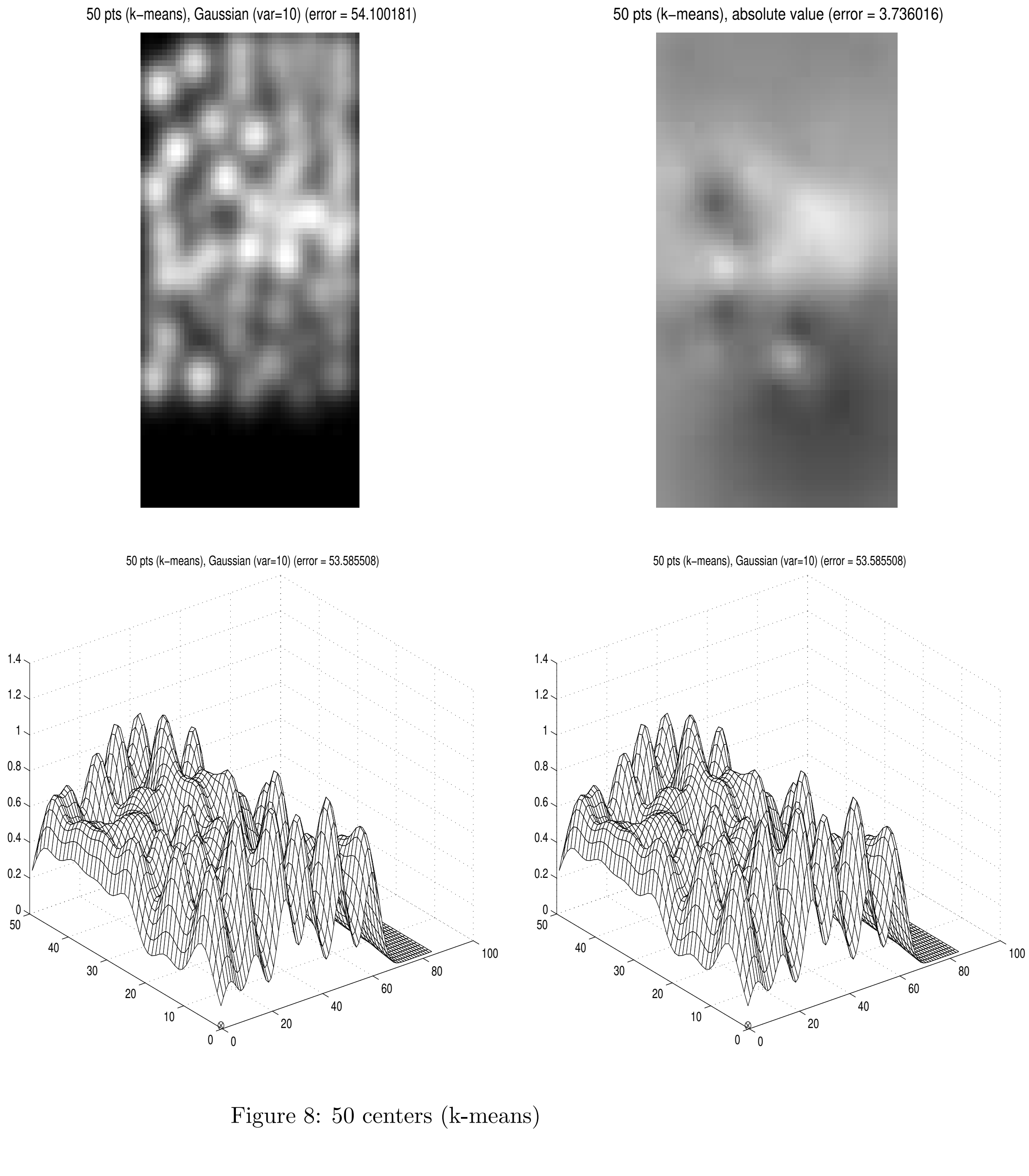

Figure 8: 50 centers (k-means)

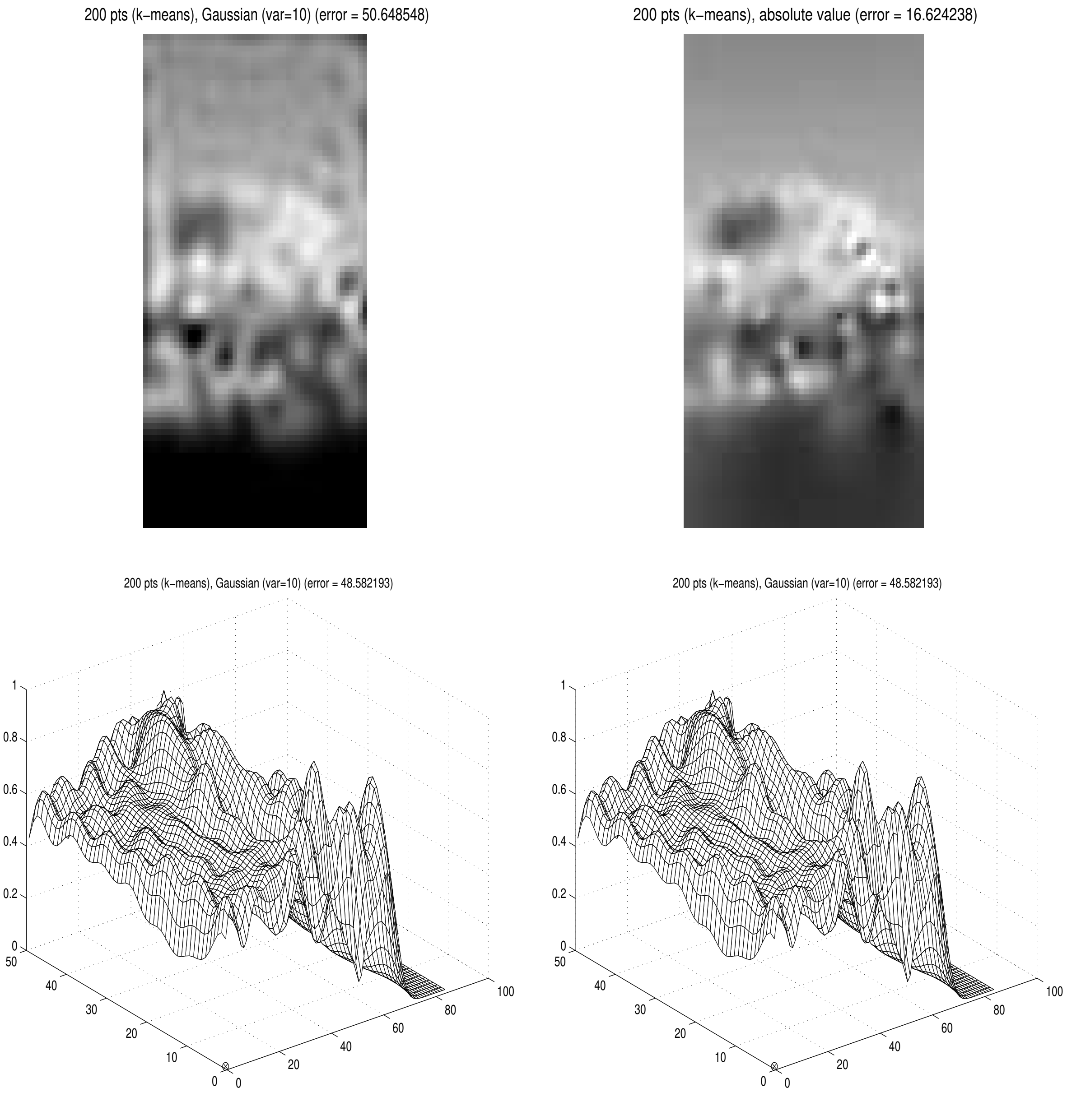

Figure 9: 200 centers (k-means)

Figure 10: the true image