LOSS FUNCTIONS FOR BAYESIAN IMAGE ANALYSIS

HÅVARD RUE*
Department of Mathematical Sciences
NTNU, Norway

MERRILEE A. HURN
School of Mathematical Sciences
University of Bath, U.K.

MARCH 1997

ABSTRACT

This paper discusses the role of loss functions in Bayesian image classification, object recognition and identification, and reviews the use of a particular loss function which produces visually attractive estimates.

1 INTRODUCTION

Bayesian image analysis often involves calculating point estimates of some appropriate quantity, for example the grey level at each pixel (in image reconstruction), the label type of each pixel (in image classification), or the number of objects together with their contours and types (in object recognition and identification). Within a decision theoretic framework, the approach to finding an estimator is first to select an appropriate *loss function* $L(\mathbf{x}, \mathbf{z})$, which gives the loss when the true value is $\mathbf{x}$ but we estimate it by $\mathbf{z}$. The optimal Bayes estimator (OBE) $\mathbf{x}^*$ is then computed by minimising the posterior expectation of this loss (the Bayes risk),

$$\mathbf{x}^* = \arg\min_{\mathbf{z}} \mathrm{E}L(\mathbf{x}, \mathbf{z}). \tag{1}$$

Commonly used $L(\mathbf{x}, \mathbf{z})$ are $L_{\text{MAP}} = 1_{[\mathbf{x} \neq \mathbf{z}]}$, $L_{\text{MPM}} = \sum_i 1_{[x_i \neq z_i]}$ in classification applications, and $L_{PE} = \sum_i (x_i - z_i)^2$ in image restoration (here, $1_{[\text{A}]}$ is the indicator taking the value one if A is true and zero otherwise). The corresponding OBE's are the posterior mode, the posterior marginal mode, and the posterior mean. For these three choices the OBE is known explicitly, and the underlying loss function often only implicitly specified. However, for more general choices of L we will need a direct minimisation of the Bayes risk (1).

Consider the following motivation: Assume for a moment that we are conducting a simulation study to investigate an image classification model using a known true scene. After blurring and adding noise to the true scene, we want to obtain a point estimate for label type of each pixel. Suppose that a

*ADDRESS FOR CORRESPONDENCE: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7034 Trondheim, Norway.
E-MAIL: havard.rue@math.ntnu.no and m.a.hurn@maths.bath.ac.uk

number of estimates are available; for example the MAP estimate approximated by a stochastic search procedure such as simulated annealing, and various other estimates found by MCMC sampling or applying other variants of coordinatewise steepest ascent algorithms as ICM (Besag, 1986). We want to see how good these estimates are by comparing them to the known true scene. To quantify this, we can use a distance measure m to rank the various estimates, and thereby choose a good estimator. But what if the true scene was unknown? A similar idea can still apply. The posterior density describes a world of possible true scenes giving the probabilities or densities for each to occur; we could instead compare the posterior expectation of m for the various estimates. It then seems reasonable to select the estimate which minimises the posterior expectation of m. We now use m implicitly as our loss function, since this is how we have chosen to quantify the distance between two scenes. It should be noted that an important assumption is being made here, we are implicitly assuming that it is possible to compute the OBE for m, a problem to which we will return in Section 3.

What are the potential benefits of widening the choice of loss function? It is likely that there will be visual benefits since none of L_{MAP}, L_{PM} and L_{MPM}, are good distance measures for images with spatial features (they all penalise errors on a pixel–by–pixel basis without considering any spatial structure in them). However, the extent to which a loss function will influence the estimate depends on the situation. Loosely speaking, the information we introduce into the inference process though a better loss function is weaker than that we introduce by designing a better prior; the role of the prior is to *introduce* information, while the role of the loss function is almost the reverse. The loss function is a way of *extracting* information from the posterior by assigning different costs to different configurations, and using this information to evaluate the posterior expectation of the loss for different configurations $\mathbf{z}$. The effect on the OBE of using different reasonable loss functions will depend strongly on the posterior. High quality data will result in similar estimates for all sensible loss functions if the posterior is unimodal with a prominent peak. For moderate quality data, a better loss function *may* yield a more sensible estimator since it minimises the loss over several reasonable configurations. For a high noise level, there is not much information to extract other than that already known though the prior. We refer the reader to Rue (1997) for a more comprehensive discussion of understanding loss functions.

In the rest of this article, we will concentrate on loss functions based on Baddeley's Delta metric (Baddeley, 1992a; Baddeley, 1992b). This metric compares images of objects or images with a small number of foreground labels. It displays reasonable visual performance and good theoretical properties. After describing the metric in Section 2, we demonstrate how to obtain the corresponding OBE in Section 3, and then show examples from image classification and object recognition in Section 4. Finally in Section 5, we will discuss some new ideas in object identification.

2 BADDELEY'S DELTA METRIC

Let Λ be a finite lattice equipped with a metric ρ, for example, $\rho(i, j)$ might be the Euclidean distance between site i and j, or the shortest path length between site i and j. Binary images can take a dichotomous value (often zero or one) at each site $i \in \Lambda$, which we interpret as background and foreground respectively. A binary image $\mathbf{x}$ can be uniquely identified by the set of foreground sites $b(\mathbf{x}) = \{i \in$

$\Lambda : x_i = 1\}$. The distance from a site i to a set $A \subseteq \Lambda$ is defined as $d(i, A) = \min_{j \in A} \rho(i, j)$. Denote by $|A|$ the number of sites in A. Let w be a strictly increasing and concave function defined on $[0, \infty]$ satisfying $w(0) = 0$. For two binary images $\mathbf{x}$ and $\mathbf{z}$, Baddeley defines their Delta distance as

$$\Delta_w^p(\mathbf{x}, \mathbf{z}) = \left(\frac{1}{|\Lambda|} \sum_{i \in \Lambda} \Big| w(d(i, b(\mathbf{x}))) - w(d(i, b(\mathbf{z}))) \Big|^p \right)^{1/p}, \quad 1 \leq p < \infty. \tag{2}$$

naming Δ_w^p the *delta metric*. The Delta metric is a L_p–average of the differences between the two images of the distances $w(d(i, b(\cdot)))$ from each site to its nearest foreground site. It is clear that (2) is a metric with respect to the distances $w(d(i, b(\cdot)))$ and also a metric with respect to the binary images $\mathbf{x}$ and $\mathbf{z}$. To simplify the following discussion we take $p = 2$. We also assume that w is eventually constant, i.e. there exists a value t_c such that $w(t) = c$ for any $t \geq t_c$. The contribution to the Delta distance from site i will then only depend on those sites in $\mathbf{x}$ and $\mathbf{z}$ within a ball of radius c centred at i. A natural choice is $w(t) = \min\{t, c\}$, where $c > 0$ is the *cut-off* value. We use the notation Δ_c^2 in this case. The cut-off value c can be interpreted as a parameter determining the importance of misplacing a foreground object in the image. Let $b(\mathbf{x})$ be a single compact object centred at some origin in the image, and let $\mathbf{x}_s$ be the image where the object is misplaced and centred at s. If s and c are less than the size of the object, then $\Delta_c^2(\mathbf{x}, \mathbf{x}_s) \approx \Delta_s^2(\mathbf{x}, \mathbf{x}_c)$, with the identity holding for a one-dimensional continuous lattice. The cost of misplacing an object by a distance s and using a cut-off value c, is roughly the same as misplacing an object by a distance c and using a cut-off value s.

To illuminate the basic properties of the Delta metric, notice that the Delta metric has two other important metrics as limits. First, if c is less than the smallest distance between two distinct sites, then $(\Delta_c^2/c)^2$ is the average sitewise error rate

$$\left(\Delta_c^2(\mathbf{x}, \mathbf{z})/c\right)^2 = \frac{1}{|\Lambda|} \sum_{i \in \Lambda} 1_{[x_i \neq z_i]}.$$

Secondly, if we let $p \to \infty$ and $w(t) = t$, we obtain the well known Hausdorff metric.

The Delta metric is not symmetric with respect to foreground and background in the image, so that the distance between $\mathbf{x}$ and $\mathbf{z}$ is in general different from the distance between the complement images $\mathbf{x}'$ and $\mathbf{z}'$ (formed by switching the foreground and background). This follows from the definition itself. The Delta metric emphasises the set of foreground sites in determining the distance between two images. This may be natural in many situations where the foreground may represent the important components in the image, for example objects, roads or edges. The background in the image is then of less interest. If the foreground and background are equally important, we can use the symmetrised version of the Delta metric $\widetilde{\Delta}_c^2(\mathbf{x}, \mathbf{z})^2 = (\Delta_c^2(\mathbf{x}, \mathbf{z}))^2 + (\Delta_c^2(\mathbf{x}', \mathbf{z}'))^2$. A metric between images with more than one foreground colour can be defined similarly.

3 Algorithm

We will now describe how to obtain the OBE using the Delta metric as a loss function, first in binary image classification (Frigessi and Rue, 1997), and then briefly how to generalise the algorithm to handle continuous models in object recognition.

Let $\mathbf{y}$ be the observed data and $\mathbf{x}$ the binary image on Λ, with posterior distribution $\pi(\mathbf{x}|\mathbf{y})$. We assume the black foreground of $\mathbf{x}$ is particularly of interest, so the choice

$$L(\mathbf{x},\mathbf{z}) = \left(\Delta_c^2(\mathbf{x},\mathbf{z})\right)^2 = \frac{1}{|\Lambda|}\sum_{i\in\Lambda}\left(\min\{c, d(i,b(\mathbf{x}))\} - \min\{c, d(i,b(\mathbf{z}))\}\right)^2 \tag{3}$$

is appropriate. To simplify the notation, we introduce $\mu_c(i;\mathbf{x}) = \min\{c, d(i,b(\mathbf{x}))\}$. How can we compute the OBE $\mathbf{x}^*$? First note that an explicit expression for $\mathbf{x}^*$ cannot be found due to non-linearity of the problem. We need to solve (1) by minimising the risk $R(\mathbf{z})$,

$$\begin{aligned} R(\mathbf{z}) &= \mathrm{E}_{\mathbf{x}|\mathbf{y}} L(\mathbf{x},\mathbf{z}) \\ &= \frac{1}{|\Lambda|}\mathrm{E}_{\mathbf{x}|\mathbf{y}}\sum_{i\in\Lambda}\left(\mu_c(i;\mathbf{x}) - \mu_c(i;\mathbf{z})\right)^2 \\ &= \frac{1}{|\Lambda|}\sum_{i\in\Lambda}\mu_c(i;\mathbf{z})\left[\mu_c(i;\mathbf{z}) - 2\mathrm{E}_{\mathbf{x}|\mathbf{y}}\mu_c(i;\mathbf{x})\right] + \text{constant.} \end{aligned}$$

The constant does not depend on $\mathbf{z}$ and is therefore of no importance in finding the OBE. Notice that $\mathbf{x}^*$ has a nice interpretation as the configuration having $\{\mu_c(i;\mathbf{x}^*)\}$ closest to the posterior expectation of $\{\mu_c(i;\mathbf{x})\}$. Also note that if c is less than the smallest distance between two pixels then $\mathbf{x}^*$ equals the MPM estimate.

The posterior expectations of $\mu_c(i;\mathbf{x})$ in (4) are not known, but we can estimate them using Markov chain Monte Carlo methods (Gilks, Richardson and Spiegelhalter, 1996), as the empirical averages of $\mu_c(i;\cdot)$ over samples from the posterior distribution $\pi(\mathbf{x}|\mathbf{y})$. Once we have estimated these posterior expectations, we then have an explicit expression for the risk $\hat{R}(\mathbf{z})$. The (estimated) OBE is then the configuration minimising $\hat{R}(\mathbf{z})$. It is fairly easy to minimise $\hat{R}(\mathbf{z})$ if we are prepared to accept local rather than global minima. Interpreting $\mathbf{z}$ as a random variable with distribution $\pi(\mathbf{z}) \propto \exp(-\hat{R}(\mathbf{z}))$, the mode of $\pi(\mathbf{z})$ can then be approximated using, for example, the Metropolis algorithm within simulated annealing. If $\mathbf{z}$ and $\mathbf{z}'$ are two configurations differing only at site i, then to compute $R(\mathbf{z}) - R(\mathbf{z}')$ we need only compare the values of $\mu_c(j;\cdot)$ for those j's where $\rho(i,j) < c$. In other words, $\pi(\mathbf{z})$ defines a Markov random field where the neighbourhood structure depends on the value of c and the metric ρ. The difference $R(\mathbf{z}) - R(\mathbf{z}')$ can be computed quite rapidly using a little book-keeping, see Frigessi and Rue (1997) for a detailed pseudo-code for this algorithm. Even modes found by applying other variants of steepest ascent and coordinatewise steepest ascent algorithms such as ICM have been shown to work well because $\hat{R}(\mathbf{z})$ is a relatively smooth function. In fact in the binary case, Frigessi and Rue (1997) give sets of linear inequalities on the $\widehat{\mathrm{E}_{\mathbf{x}|\mathbf{y}}\mu_c(i;\mathbf{x})}$'s for which it can be determined without any minimisation whether x_j^* is background, foreground or still unknown. In this case, only the set of unknown sites have to be determined using simulated annealing. These inequalities are easy to check and improve the algorithm in terms of both speed and robustness. In summary, the additional cost for estimating the OBE using (3) instead of the marginal posterior mode (MPM) estimate, is only about 20%. This includes the extra minimisation step that is needed. The additional coding needed is limited and trivial.

It is possible to generalise the algorithm just described both to other loss functions (Rue, 1995; Rue, 1997), and to other image models. We briefly describe the generalisation to object recognition (Rue and Syversveen, 1995) where $\mathbf{x}$ is the union of a unknown number of objects x_i, and where each object is represented by a low-parametric model describing its outline and position. The loss function is taken as (3) where the lattice Λ now approximates the continuous area of interest. The changes needed in the algorithm are the following; $\mu_c(i;\mathbf{x})$ is now the distance from point i in Λ to the closest object's edge (maximum c), and $\mu_c(i;\mathbf{x})$ is zero if i is inside an object. Notice that the loss for wrong number of objects in the scene is implicitly present in (3). The algorithm then follows the same outline as for image classification, involving surprisingly little extra coding effort. We refer the reader to Rue and Syversveen (1995) for details.

4 EXAMPLES

4.1 IMAGE CLASSIFICATION

We will show a typical classification experiment demonstrating how the OBE based on Baddeley's Delta metric gives a better estimate than MPM (which corresponds to the choice $c = 1$ in the regular grid case). Let $\mathbf{x}$ be a binary image taking the values $+1$ or -1 on a 32×32 lattice. We represent the foreground as black and the background as white. We use the Ising model with inverse temperature $\beta = 0.5$ as the prior model. The data $\mathbf{y}$ is the true scene corrupted by additive pixel-wise independent Gaussian noise with zero mean and variance $\sigma^2 = 1/4$. Figure 1a shows the true scene and Figure 1b shows the noisy data image The MPM estimate and the OBE based on Baddeley's Delta metric using the shortest path metric and $c = 4$ are shown in Figure 1c and Figure 1d respectively. Notice how the new estimator restores the fine structure of the letter, which is the main objective here.

4.2 OBJECT RECOGNITION

Our next example is from object recognition where the goal is to locate an unknown number of objects, estimating the outline and position of each (Rue and Syversveen, 1995). Each object is described using a low-parametric model involving a prototype and some deformations. Figure 2a and Figure 2b show the true scene (with values zero and one) and the data formed by adding independent identically distributed Gaussian noise with variance one. Figure 2c shows the MAP estimate and Figure 2d the Delta–estimate using the Delta metric as the loss function with $c = 5$. These results are typical; the Delta-estimated outline is more visually appealing and not as wiggly as the MAP estimate.

In addition to producing better estimates of the outline of each object, the Delta–estimate seems to be more stable with respect to the number of objects, seldom overshooting; the MAP-estimate can produce spurious objects (see (Rue and Syversveen, 1995) for examples). The intuitive explanation of this is that the MAP–estimate depends only on the mode of the posterior distribution and not on how much probability this mode contains. It is where (and how) the probability mass in the posterior distribution is located that should be important for the inference, not necessarily just the location of the mode in the posterior.

5 Extensions to object identification

The Delta metric implicitly gives the loss for the number of objects and their outline, but not for properties such as grey–level or object type. We will now briefly describe some recent results which make it possible to estimate properties of objects in addition to the number, their position and outline (Rue and Hurn, 1997). See also Hurn and Rue (this volume) for more details of the underlying model and applications.

To estimate the type of each of the objects in a scene, it is not possible simply to add terms into the loss function which specify the loss for object type a if we estimate it by type $\hat{a}$. This is related to the variable dimension aspects of the problem (the number of objects is unknown, and the model is defined for an unordered set of objects $\{x_1, x_2, \ldots\}$). To overcome this difficulty, we propose to associate the type–property with the discretized lattice Λ, specifying the loss for mislabeling pixels i in Λ rather than the objects themselves. For example,

$$L_{\text{type}}(\mathbf{x}, \mathbf{z}) = \frac{1}{|\Lambda|} \sum_{i \in \Lambda} 1_{\text{type}}(i; \mathbf{x}, \mathbf{z}), \tag{4}$$

where $1_{\text{type}}(i; \mathbf{x}, \mathbf{z})$ is the indicator function taking the value one if the type at i is wrong and zero otherwise. Other properties of an object, such as its grey–level, can be dealt with in a similar manner.

We now take the total loss as a weighted sum of (3) and (4),

$$L(\mathbf{x}, \mathbf{z}) = \frac{1}{|\Lambda|} \sum_{i \in \Lambda} (\mu_c(i; \mathbf{x}) - \mu_c(i; \mathbf{z}))^2 + \frac{\gamma}{|\Lambda|} \sum_{i \in \Lambda} 1^{\text{type}}(i; \mathbf{x}, \mathbf{z}).$$

To obtain the OBE we need to evaluate the risk,

$$R(\mathbf{z}) = \frac{1}{|\Lambda|} \sum_{i \in \Lambda} \mu_c(i; \mathbf{z}) \left[\mu_c(i; \mathbf{z}) - 2\mathrm{E}_{\mathbf{x}|\mathbf{y}} \mu_c(i; \mathbf{x})\right] + \frac{\gamma}{|\Lambda|} \sum_{i \in \Lambda} \mathrm{Prob}_{\mathbf{x}|\mathbf{y}}(1^{\text{type}}(i; \mathbf{x}, \mathbf{z}) = 1).$$

The required posterior expectations and probabilities can be estimated from samples from the posterior, as the first step in the algorithm. The OBE can then be approximately obtained using simulated annealing or some other optimisation algorithm. Again, the extra amount of coding needed is surprisingly small, and the additional CPU–time is quite reasonable.

6 Discussion

We have reviewed and discussed some loss functions in Bayesian imaging which give visually appealing results. In particular, we have focused on those derived from Baddeley's Delta metric, which seems to provide a good balance between complexity and benefits. We have also discussed some recent results for object identification, where we need to estimate simultaneously the number of objects, their positions, outlines and object types. See also Rue (1995) and Rue (1997) for further examples.

In conclusion, it is possible to obtain the OBE at least approximately for a variety of spatial loss functions. The additional effort needed both in terms of extra CPU and coding is surprisingly small for those derived from Baddeley's Delta metric. The improvement in performance over the usual estimators is most prominent at a medium noise level. For low–level pixel–based models in image classification and restoration, the improvements are on a local scale. For high-level models in object recognition, the differences are also on an object scale.

ACKNOWLEDGEMENTS

Parts of the research were funded by the Research Council of Norway and the British Council under their scholarship scheme for collaborative research. The second author also acknowledges the financial support of the Nuffield Foundation.

REFERENCES

Baddeley, A. J. (1992a). An error metric for binary images, *in* W. Förstner and S. Ruwiedel (eds), *Robust Computer Vision: Quality of Vision Algorithms*, Wichmann Verlag, pp. 59–78.

Baddeley, A. J. (1992b). Errors in binary images and a L^p version of the Hausdorff metric, *Nieuw Archief voor Wiskunde* 10: 157–183.

Besag, J. (1986). On the statistical analysis of dirty pictures (with discussion), *Journal of the Royal Statistical Society, Series B* 48: 259–302.

Frigessi, A. and Rue, H. (1997). Bayesian image classification with Baddeley's delta loss, *Journal of Computer Graphics and Statistics*. (to appear).

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*, Chapman & Hall.

Rue, H. (1995). New loss functions in Bayesian imaging, *Journal of the American Statistical Association* 90: 900–908.

Rue, H. (1997). A loss function model for the restoration of grey level images, *Scandinavian Journal of Statistics* 24(1): 103–114.

Rue, H. and Hurn, M. A. (1997). Bayesian object identification, *Statistics*, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway.

Rue, H. and Syversveen, A. R. (1995). Bayesian object recognition with Baddeley's delta loss, *Statistics no. 8*, Departments of Mathematics, The University of Trondheim, Trondheim, Norway.

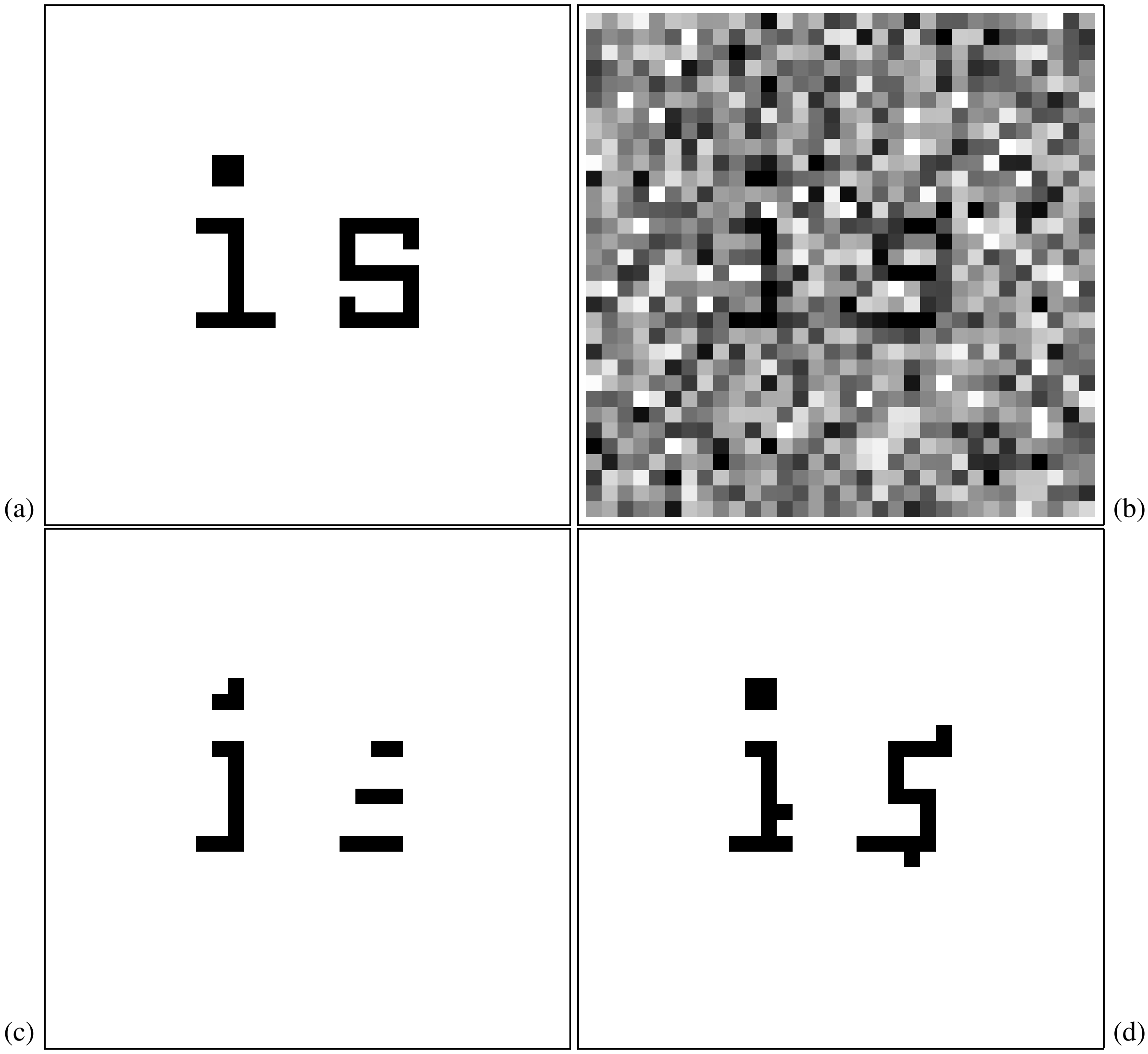

Figure 1: An image classification example. (a) the true scene; (b) the data when iid Gaussian noise with variance 0.5^2 is added; (c) the MPM classification; (d) the classification with the squared Baddeley's Delta metric as the loss function using the shortest path metric and $c = 4$.

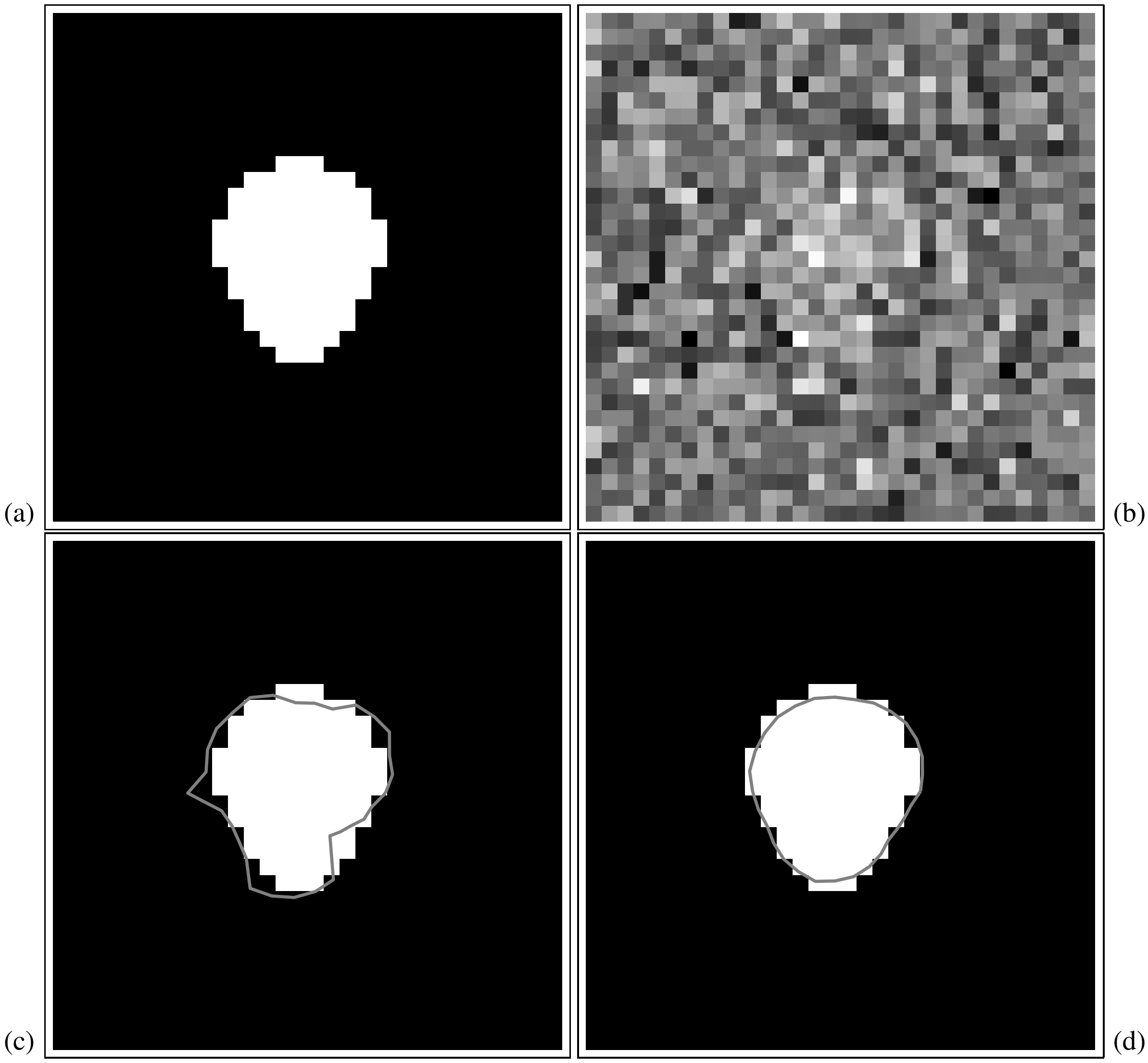

Figure 2: An object identification example; (a) the true discretized object with levels zero and one; (b) the noisy data after adding iid Gaussian noise with zero mean and variance one; (c) the MAP estimate; (d) the estimate using the Delta metric as a loss function and $c = 5$.