3.2 Probabilistic Hedging of a Skill Score

Neil Gordon*, Devin Kilminster and Andy Ziegler
Meteorological Service of New Zealand, Wellington, NZ

1. Introduction

The 2008 AMS Artificial Intelligence competition (Elmore and Richman 2009) was based on the analysis of data from a polarimetric weather radar. For each data point the observed precipitation at the ground was in one of three categories: *none*, *liquid* or *frozen*. We were supplied with a training dataset of 847 observations. Each data point had the observed category, as well as 16 potential predictor variables (attributes).

The task was to develop a classifier which was then applied to a test dataset of 363 data points, for which the observed precipitation class was withheld. The predicted classes were to be compared with the observed classes, and scored using the multi-category form of the True Skill Statistic (sometimes called the Hanssen-Kuipers Skill Score or the Peirce Skill Score).

The competition required a deterministic classification; we chose to start by estimating class probabilities and then trying different strategies for converting the probabilities into a single deterministic class, with a view to optimising the skill score. This paper describes those strategies and how they gave quite different results for optimising the True Skill Statistic (TSS), as opposed to the Heidke Skill Score.

Unfortunately, only once the competition had been judged, it was discovered that we had used the incorrect formulation of the TSS because there was an error in the definition referenced at the competition web site. The TSS we used was quite susceptible to hedging. Our highly hedged entry (referred to later as Strategy 2) would have been a clear winner of the competition, if that TSS had been used for the official judging. Classifications based on our more conventional strategies, judged using the correct TSS or Peirce Skill Score (PSS), came in the middle of the pack of the other entries.

No matter what score or utility function is used to evaluate a set of deterministic classifications, this paper demonstrates that there are strategies which are generally applicable to the conversion of class probabilities to deterministic classes which can seek to optimise that score over a complete dataset, and can also provide estimates of the distribution of the likely outcome of the score.

2. Data and Method

a. *Data*

There are 16 potential attributes, including an index number, positional information, surface wind, temperature and relative humidity, the freezing level, and radar positional, reflectivity and phase information.

We chose to make use of all the attributes, including the index number. We noted missing sequential index numbers in the training dataset, and suspected that the index number may provide useful information about the weather regime for the test dataset.

We ran our own mini competition among ourselves. We removed a random selection of 100 observations from the 847 in the training dataset, tried various classifiers on the reduced training dataset of 747 data points and then tested our results on the withdrawn test dataset of 100. We will refer to this as our internal test dataset. The approach described here is based on a combination of two of the successful methods.

b. *Class Probability Estimation*

The core algorithm we used is the random forests (Breiman, 2001) implementation included within Weka (version 3.5.8). Weka is a free, downloadable system of machine learning software available via `http://sourceforge.net/projects/weka/`, described by Witten and Frank (2005). Weka is flexible and highly configurable, and is easily able to read the datasets provided for the competition. A series of experiments using 10-fold cross-validation, trying many of the available Weka algorithms, suggested that random forests worked well. We have also had prior positive experience with the random forests algorithm, and it has been used for previous AI competitions (e.g., Williams and Abernethy 2008). For this study, it was configured for 100 trees, with each tree using three features.

The output from the random forests algorithm includes class probability estimates, as well as the most likely class (taken by default as the one with the highest probability). The class probabilities are simply a count of the number of decision trees resulting in that particular class. There is no guarantee that these are reliable probability estimates, and they are also too confident since values of 0% (0/100 trees) and 100% (100/100 trees) can appear.

We carried out a simple check on the reliability of the random forests probabilities, as applied to the internal dataset of 100 data points. With such a small sample, we used just five bins. The diagrams shown in Figs 1 to 3 suggest the probabilities are reasonably reliable, as

*[*] *Corresponding author address:* Dr Neil Gordon, MetService, P.O. Box 722, Wellington 6014, NZ; email: Neil.Gordon@msetservice.com.

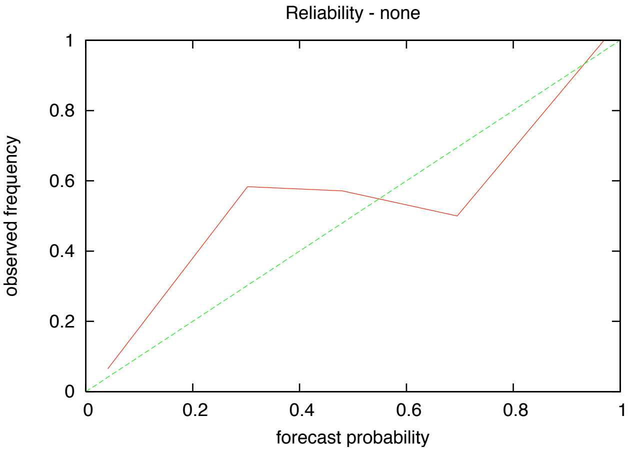

Figure 1: Reliability diagram for precipitation of *none* versus either *liquid* or *frozen*. The solid red line gives the forecast versus actual probabilities in 5 bins, with the dotted green diagonal line indicating perfect reliability.

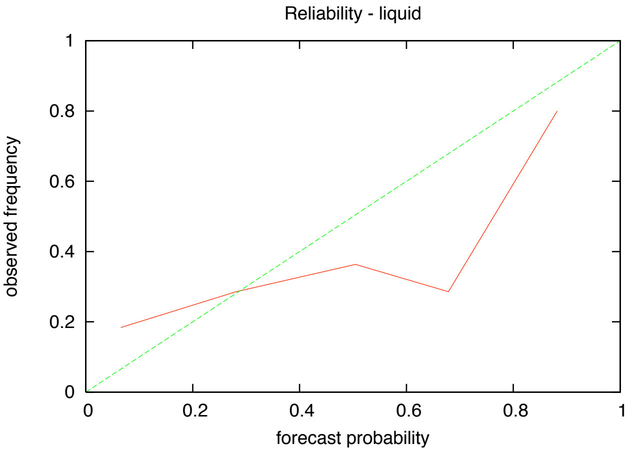

Figure 2: Reliability diagram for precipitation of *liquid* versus either *none* or *frozen*.

they follow the diagonal quite well.

It may be possible to further process the probabilities (e.g., using logistic regression) to make them more reliable. However, for the purposes of this paper and the competition we have treated the probabilities directly from the random forests algorithm as being reliable.

c. *Converting Probabilistic Forecasts to Deterministic Classes*

The competition was to be judged using the multi-category form of the True Skill Statistic. The definition of this score which we used, from `http://www.bom.gov.au/bmrc/wefor/staff/eee/verif/verif_web_page.html`, for K categories, was

$$\text{TSS} = \frac{\frac{1}{N}\sum_{i=1}^{K} n(F_i, O_i) - \frac{1}{N^2}\sum_{i=1}^{K} N(F_i)N(O_i)}{1 - \frac{1}{N^2}\sum_{i=1}^{K}(N(F_i))^2}, \quad (1)$$

where $n(F_i, O_j)$ denotes the number of forecasts in category i that had observations in category j, $N(F_i)$ denotes the total number of forecasts in category i, $N(O_j)$ denotes the total number of observations in category j, and N is the total number of forecasts.

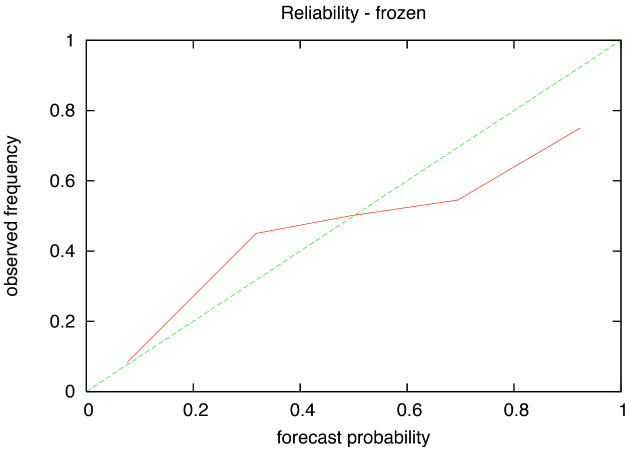

Figure 3: Reliability diagram for precipitation of *frozen* versus either *none* or *liquid*.

Unfortunately, the definition at the referenced web site was in error. The denominator used the counts of forecasts for each category, rather than the counts of observations as in the correct formulation in equation 7.21 of Wilks (2006). This made it quite susceptible to hedging. For convenience, we will continue to refer to this as the TSS in this paper.

Another score which we also computed, is the Heidke Skill Score (HSS). This is defined as

$$\text{HSS} = \frac{\frac{1}{N}\sum_{i=1}^{K} n(F_i, O_i) - \frac{1}{N^2}\sum_{i=1}^{K} N(F_i)N(O_i)}{1 - \frac{1}{N^2}\sum_{i=1}^{K} N(F_i)N(O_i)} \quad (2)$$

We trained the Weka random forests algorithm on the training dataset, and then applied it to the test dataset to produce estimated class probabilities for each observation. We then applied three Strategies:

1. Choose the class with the highest probability (also the default method for assigning the class from the Weka random forests algorithm)
2. Optimise the assignment of deterministic forecast classes to the unknown observations in order to maximise the TSS of the expected contingency table. For any such assignment, it is a straightforward matter to use the forecast probabilities to compute an expectation for each member of the contingency table, and hence calculate the objective. We approximately solve the resulting combinatorial optimisation problem by an application of simulated annealing followed by simple steepest descent.
3. As for Strategy 2, only maximise the HSS of the expected contingency table of predicted versus observed.

There are many other possible strategies, but these three were chosen because they are relatively simple and straightforward to implement, and provide at least a few representative examples. While simple, these strategies might not be the most useful. For example, we could

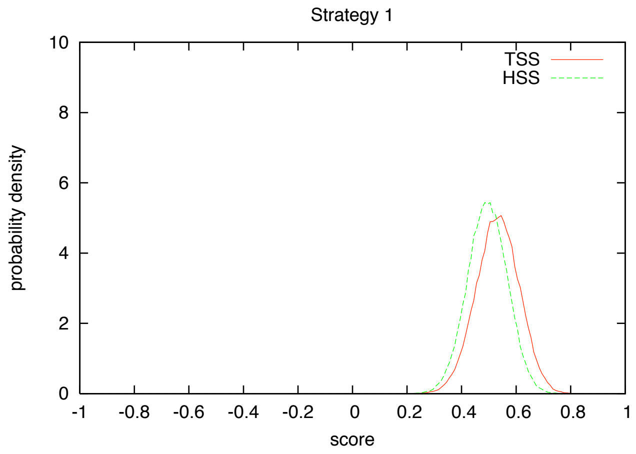

Figure 4: Histogram (based on 100,000 Monte Carlo simulations) of the TSS and HSS for Strategy 1: choosing the class with the highest probability.

have tried the more difficult strategy of assigning the deterministic forecast classes to optimise the expected TSS of the contingency table, rather than to optimise the TSS of the expected contingency table (Strategy 2).

d. *Evaluating the Strategies*

Whatever the strategy to convert probabilities into deterministic classes, given a set of resulting forecast classes, we can use a Monte Carlo technique to sample from the possible outcomes based on the predicted class probabilities for the data points in the test dataset. This allows us to compute a forecast distribution of any particular score, on the assumption of the class probabilities being reliable. We can do this to see whether the distribution conforms with our utility. For example, we might want to maximise our chances of winning a competition by maximising the probability of the score exceeding our guess at what a typical "good" method would achieve, while not also having a high chance of a very bad score.

Our Monte Carlo implementation used 100,000 samples. For each of these, we computed the resulting TSS and HSS, and formed histograms using 1% bins. This was done for both the internal test dataset and the competition test dataset.

For our own internal test dataset of 100 data points we could also compute the actual contingency table, and TSS and HSS.

3. Results

a. *Results for the Internal Test Dataset*

Figure 4 shows the forecast distribution of the TSS and HSS when applying Strategy 1 to our internal test dataset of 100 data points. Note that the HSS is expected to be slightly smaller than the TSS, with both being in the vicinity of 0.5.

		Predicted			
		none	*liquid*	*frozen*	Total
Observed	*none*	7	4	8	19
	liquid	3	10	14	27
	frozen	1	7	46	54
	Total	11	21	68	100

Table 1: Contingency table for Strategy 1: choose class with highest probability.

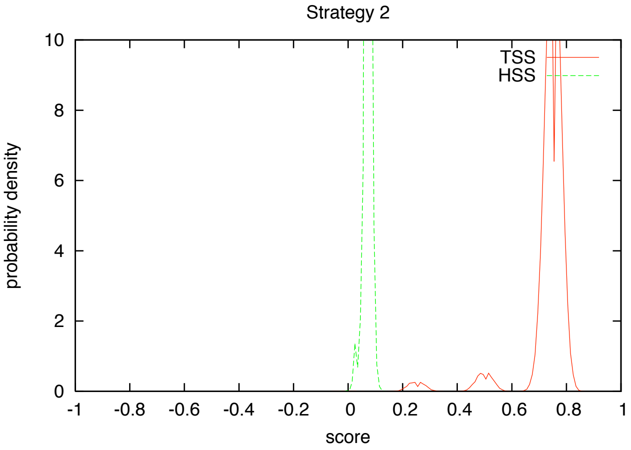

Figure 5: Histogram (based on 100,000 Monte Carlo simulations) of the TSS and HSS for Strategy 2: optimise the TSS of the expected contingency table.

For this test dataset, unlike that used for the real competition prior to judging, we know the answers. For Strategy 1, the resulting contingency table is shown in Table 1. The *froze*n category is over-predicted, probably because it is the most frequent class and so tends to be chosen too often with this strategy. It occurred just 54 times, but was predicted 68 times. The average forecast probability for *frozen* was 0.61, which implies that the random forests algorithm was expecting 61 *frozen*, and also 26 *liquid* and 13 *none*. This gives a TSS of 0.385 and an HSS of 0.333, which are within the forecast distribution from Fig. 4, albeit towards the lower end.

Strategy 2, optimising the True Skill Statistic of the expected contingency table, gives quite different results. Because of the nature of the form of the True Skill Statistic that we used, which includes in its denominator only the distribution of the forecasts, and not the distribution of the observations, we are encouraged to produce classifications which are highly hedged. In fact, applying this strategy results in a prediction of the *frozen* class for all but two data points, which are predicted to be *none*. Fig. 5 shows the results of the Monte Carlo simulations for the HSS and TSS. We seem to have a very good chance of achieving a high TSS of around 0.7. There is also a low likelihood of blowing the prediction, and getting a score in the vicinity of 0.5, or 0.3, or even negative (this chance is so small that it is imperceptible on this particular graph).

Clearly a prediction that all but two of the observations will be *frozen* is of no practical value (aside perhaps from winning competitions). In this case, the HSS seems to be a more robust indicator of value as Fig. 5 shows the expected HSS to have near or slightly above zero skill.

In actuality, for our internal test dataset, the Strategy 2

		Predicted			
		none	liquid	frozen	Total
Observed	none	2	0	17	19
	liquid	0	0	27	27
	frozen	0	0	54	54
	Total	2	0	98	100

Table 2: Contingency table for Strategy 2: optimise the TSS of the expected contingency table.

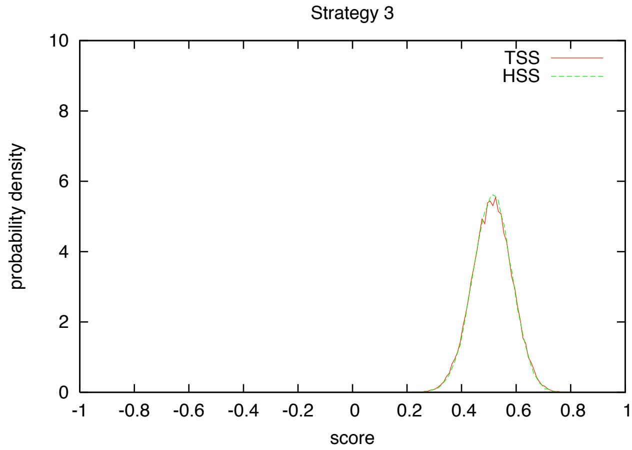

Figure 6: Histogram (based on 100,000 Monte Carlo simulations) of the TSS and HSS for Strategy 3: optimise the HSS of the expected contingency table.

prediction turned out to be correct for the two data points classified as *none*, as seen in Table 2. The resulting TSS is a very good 0.689 with an HSS (reflecting very little skill) of 0.058.

Strategy 3, which optimises the HSS of the expected contingency table, produces almost matching histograms of the expected HSS and TSS, as shown in Fig. 6, with means of around 0.5. It is worth noting that the TSS and HSS will be the same when the forecast frequencies exactly match the observed frequencies. Indeed, in this case the frequency of predicted classes is a better match to the observed frequencies, with the resulting contingency table in Table 3. The TSS is 0.379, very slightly less than the 0.385 for Strategy 1, and the HSS is 0.361, which is better than the 0.333 for Strategy 1. As with Strategy 1, both the TSS and HSS are within the distributions in Fig. 6, although towards the low end.

		Predicted			
		none	liquid	frozen	Total
Observed	none	11	3	5	19
	liquid	3	10	14	27
	frozen	3	9	42	54
	Total	17	22	61	100

Table 3: Contingency table for Strategy 3: optimise the HSS of the expected contingency table

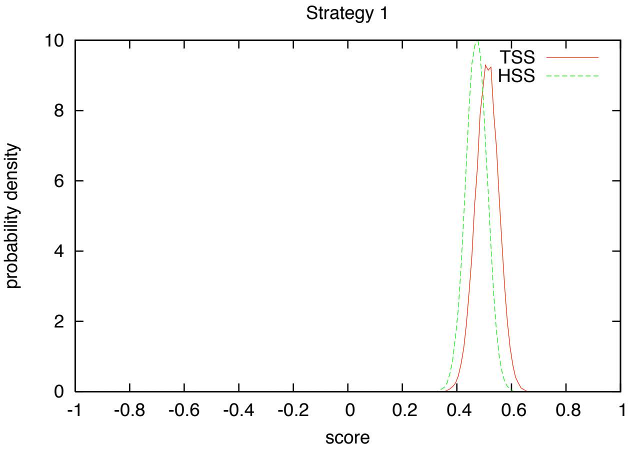

Figure 7: Histogram (based on 100,000 Monte Carlo simulations) of the TSS and HSS for Strategy 1, applied to the competition test dataset.

b. *Classifications for Competition Test Dataset*

We received the competition test dataset of 363 data points on 18 December 2008 and applied our approach. We trained the Weka random forests algorithm on the full training dataset of 847 data points, applied the algorithm to the test dataset, and then used the three strategies to convert the predicted class probabilities for the test dataset to deterministic classes.

The average predicted probabilities for the competition test dataset were 0.58 for *frozen*, 0.31 for *liquid* and 0.11 for *none*. This compares with the corresponding sample frequencies from the test dataset of 0.58, 0.28 and 0.14; the test dataset appears to have a similar distribution of precipitation types to the training dataset. For the 363 member test dataset, based on the random forests probabilities, we would expect around 211 *frozen*, 113 *liquid* and 38 *none* observed classes. We subsequently learned in January 2009 that the actual numbers were 206, 116 and 41; a good result.

Strategy 1 converts the probabilities to 242 *frozen*, 85 *liquid* and 38 *none*. So the *frozen* class is probably over-forecast again, at the expense of the *liquid*. Again, this expectation verified. The resulting histogram (based again on 100,000 Monte Carlo simulations) is shown in Fig. 7. The means of the TSS and HSS are around similar values to those for the internal test dataset in Fig. 4, but the distribution is much narrower because we have more data and therefore less uncertainty. We might expect to have a good chance of achieving a TSS of at least 0.45 from Strategy 1.

Applying Strategy 2 to the competition test dataset results in all the data points bar one being classified as *frozen.* The single *none* is the observation with the index number of 1696, for which the random forests algorithm gave a probability of 98% for the *none* class. The resulting histogram is shown in Fig. 8. The histogram for the TSS shows three peaks resulting from observed outcomes of *none* (correct), *liquid* or *frozen* for this particular data point, with some uncertainty about those points due to Monte Carlo sampling for the remainder of the data points. We expect these distributions to be under-

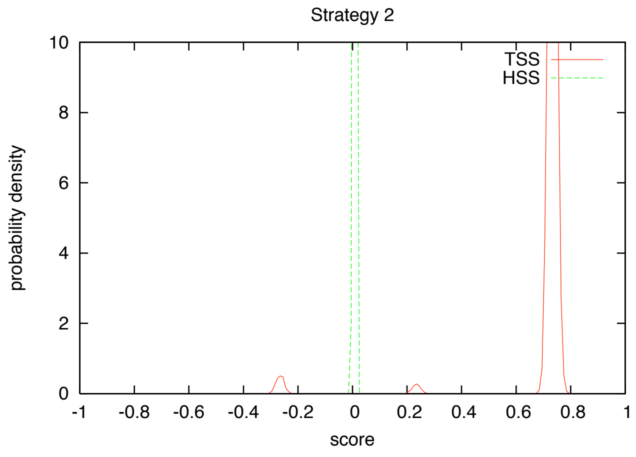

Figure 8: Histogram (based on 100,000 Monte Carlo simulations) of the TSS and HSS for Strategy 2, applied to the competition test dataset.

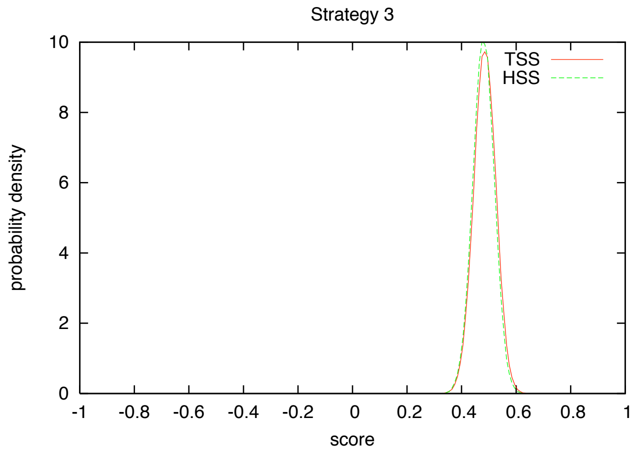

Figure 9: Histogram (based on 100,000 Monte Carlo simulations) of the TSS and HSS for Strategy 3, applied to the competition test dataset.

dispersive; they should be wider. Still, on the face of it, a competition entry based on this strategy had a very good chance of achieving a score using the TSS as defined of around or just over 0.7.

For Strategy 3, the frequency of predicted classes is 217 *frozen*, 102 *liquid* and 44 *none*, which is a better match to what we expected the observed frequencies would be in the competition test dataset. This indeed turned out to be the case, when we received the verification dataset in January 2009. The histogram of the expected TSS and HSS is shown in Fig. 9. The TSS and HSS have similar distributions, and are again much narrower than for the internal test dataset, because we have more data.

4. Choice of Official Entries to the Competition

For our main formal entry to the competition we chose to submit the results for Strategy 2. We didn't think this provided a particularly useful classification, but we believed it had a good chance of winning, given the official judging criterion of using the True Skill Statistic as defined on the referenced web page.

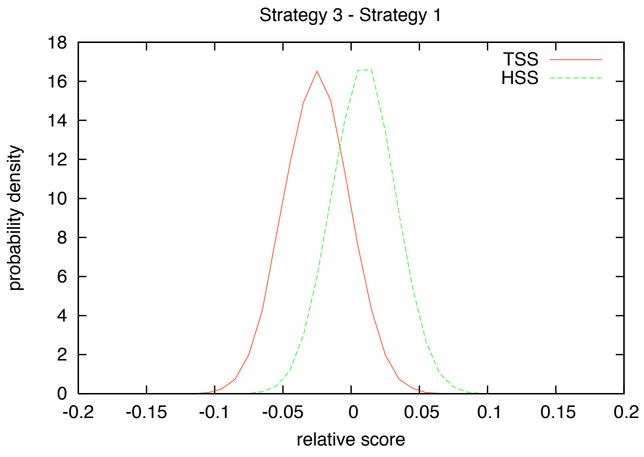

Figure 10: Histogram (based on 100,000 Monte Carlo simulations) of the pairwise difference of the TSS and HSS of the predictions using Strategy 3 minus the TSS and HSS for the predictions using Strategy 1, applied to the competition test dataset.

We also wanted to submit at least one extra entry with useful results. The question was whether that should be based on Strategy 1 or Strategy 3. We again used 100,000 Monte Carlo simulations to compare the pairwise differences between the TSS and HSS, with the resulting histograms in Fig. 10. This shows that the differences are not significantly different from zero for either of them, although the HSS had perhaps a two-thirds chance of being higher for Strategy 3 than for Strategy 1, and the TSS was quite likely to be less for Strategy 3 than for Strategy 1.

On balance, we submitted a second entry using Strategy 1, since we thought that was also likely to do reasonably well using the official judging criterion of the (incorrect) TSS. If the criterion had been the HSS, Strategy 3 is what we would have used for the second entry. In practice, we provided classifications based on all three Strategies to the organisers on 19 December 2008.

5. Competition Result and Conclusions

This competition required a deterministic classifier for weather radar observations, using a judging criterion based on the True Skill Statistic. We took the approach of training a probabilistic classifier and then developing strategies for converting the probabilities to deterministic classes to obtain good skill scores, and also using the probabilities to forecast the likely distribution of the scores on a test dataset. We found that for this particular case, and our method, the True Skill Statistic (that we used) can be hedged, and we submitted one official entry which appeared to have a good chance of a very high score using the judging criteria, but was otherwise valueless.

We were dismayed when we received the results on 20 December 2008. We were advised that the TSS for our official entry based on Strategy 2 (optimising the TSS) was 0.007. We assumed that our gamble of the single *none* had not paid off. It was only in early January 2009 that we realised that such a score was very unlikely, given the expected scores from Fig. 8. Our score should have been around +0.73, +0.25 or -0.26; not around zero.

On querying the organisers of the competition, the error in the TSS formulation we used was discovered. It also eventuated that this incorrect formulation had been used to judge the previous year's competition.

On obtaining the full verification dataset, we calculated that, using the TSS we had assumed was correct, our score was 0.730 for Strategy 2, 0.334 for Strategy 1 and 0.303 for Strategy 3. This validated our choice of entries for the rules we were playing to, and we were quite satisfied with this result. Although the scores for Strategies 1 and 3 were at the very lower limits of what might have been expected from Figs 7 and 9, the difference of 0.03 between Strategies 1 and 3 was close to the mean expectation from Fig. 10.

Using the correct formulation for the TSS or PSS, our score was 0.007 for Strategy 2 (which the organisers agreed would no longer be our official entry), 0.291 for Strategy 1 and 0.295 for Strategy 3. Strategy 3, since it was based on optimising the HSS which was similar to the correct TSS, became our new official entry; it was placed in the middle of the pack with scores for other competition entries which ranged from 0.236 to 0.355.

No matter what score or utility function is used to evaluate a set of deterministic classifications, this paper demonstrates that there are strategies which are generally applicable to the conversion of class probabilities to deterministic classes which can seek to optimise that score over a complete dataset, and can also provide estimates of the distribution of the likely outcome of the score. Of course, if future competitions were to be based on probabilistic forecasts or classifications, then such a conversion would not be necessary.

References

Elmore, K. and M. Richman, 2009: Polarimetric radar and validation data for the 2008 artificial intelligence competition. *Seventh Conference on Artificial Intelligence Applications to Environmental Science*, Amer. Meteor. Soc., Phoenix, AZ, 3.1.

Wilks, D., 2006: *Statistical Methods in the Atmospheric Sciences*. Elsevier, second edition, 627 pp.

Williams, J. and J. Abernethy, 2008: Using random forests and fuzzy logic for automated storm type identification. *Sixth Conference on Artificial Intelligence Applications to Environmental Science*, Amer. Meteor. Soc., New Orleans, LA, 2.2.

Witten, I. H. and E. Frank, 2005: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, second edition, 560 pp.