Using optimum score estimation to create a skill score ranking with samples from the Duffing map

Trevor Maynard *

February 14, 2017

A hypothetical underlying distribution is created by kernel dressing a sample of output from the Duffing distribution and a family of probability forecasts is created one of which will exactly match the underlying. The key parameter which indexes the forecasts is the kernel width. Samples from the underlying are taken and optimum score estimation is used to choose the forecast with the best score given the sample. The choice of the forecast is equivalent to estimating the kernel width which can be compared to the true kernel width of the underlying. The difference between the chosen kernel width and the true value is an objective measure of how well the skill score has performed. This allows different scores to be compared. This experiment finds that IJ Good's Ignorance score performs better than all other scores. The experiment also explains why the Root Mean Squared statistic is fundamentally flawed in this context.

*LSE, Lloyd's of London

1 Experiment design

Gneiting and Raftery [4] note that one of the uses of skill scores is 'optimum score estimation' where parameters are found by finding those which produce the best average score given observations (defined in full in equation 2 below); this is also explored in Du and Smith [2]. This section describes a situation where the true parameters are known by construction and then the optimum score estimation technique is carried out using a variety of scores. The scores which lead to parameters that are closer to the true parameters are deemed to have done better than those which find parameters further away. Such experiments lead to a clear ordering amongst scores (in each example). The Ignorance score is shown to perform well in this setting.

Optimal Score Estimators Given a forecast $f(x, \theta)$ of a variable x with parameters θ and observations $X_1, ...X_n$, and a Strictly Proper scoring rule $S(p, X)$ Gneiting and Raftery [4] define the 'Optimal Score Estimator' ($\hat{\theta}$) as follows. Let

$$\mathcal{S}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} S(f(x, \theta), X_i) \tag{1}$$

then define

$$\hat{\theta}_n = \text{argmin}_\theta(\mathcal{S}_n(\theta)) \tag{2}$$

argmin($\mathcal{S}(\theta)$) is a function that returns the minimum value of $\mathcal{S}_n(\theta)$ over all possible values of the parameter θ. Let $\tilde{\theta}$ be the true parameter underlying a data generating process, then Gneiting and Raftery [4] note that $\hat{\theta}_n \to \tilde{\theta}$ for Strictly Proper scores. (Note their expression uses argmax because they use positive orientation for their scores.)

The following procedure creates an underlying distribution, $f_u(x)$ for an observed variable x whose functional form is defined by a single parameter σ_u. Multiple probability forecasts are also produced using the same procedure. Observations are sampled from the underlying distribution and Optimal Score Estimation is used to choose the forecast with the best score. For a given set of observations this is repeated for multiple score types to test which score chooses the forecast that is 'closest' (defined on page 3 below) to the underlying distribution.

Defining the underlying distribution - Kernel Dressing The following method is known as **'Kernel Dressing'** [1]. Define a PDF f_u as follows. Given any set of N real numbers $S_u = \{r_1, ..., r_N\}$, let σ_u be a kernel width and let the **'underlying distribution'** be defined as:

$$f_u(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma_u} \phi(\frac{x - r_i}{\sigma_u}) \tag{3}$$

Where,

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \tag{4}$$

Create observations Sample N_{obs} values $x_1, ...x_{N_{obs}}$ from the underlying distribution f_u - call these **'observations'**. This process can be repeated N_{seed} times to produce multiple sets of observations to quantify the impact of sampling error.

Create a family of forecasts Let $\sigma_m \in \{\sigma_1, ...\sigma_{M_{fcsts}}\}$ be one of M_{fcsts} positive real numbers and use the same set S_u to define forecast distributions as follows:

$$p_m(\sigma_m, x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma_m} \phi(\frac{x - r_i}{\sigma_m}) \tag{5}$$

This creates a family of probability forecasts indexed by σ_m and when $\sigma_m = \sigma_u$ the forecast p_m is exactly equal to the underlying distribution f_u.

Definition of 'closeness' Given forecasts p_i ($i \in \{1, 2\}$) (with kernel widths σ_i). p_1 is **'closer'** to the underlying distribution f_u than p_2 if $|\sigma_1 - \sigma_u| < |\sigma_2 - \sigma_u|$.

Create datasets S_u to avoid serendipity Given that the underlying distributions are defined using Gaussian kernels there would be a danger that creating the data sets S_u by sampling (say) from Gaussian distributions will produce results that are not general because of the common distribution family. Even other well known distributions such as Lognormal or Gamma may be too 'well behaved'. To avoid this unwanted serendipity the data sets S_u are generated from a dynamical process (the 'Duffing map') which produces highly non-Gaussian outputs. The Duffing map, a discrete version of the Duffing equation [3], is defined as follows:

$$X_{k+1} = Y_k \tag{6}$$

$$Y_{k+1} = -bX_k + aY_k - Y_k^3 \tag{7}$$

Where a and b are parameters and X_0 and Y_0 are given initial values from which iterative values are generated.

The following algorithm is used to produce the real numbers in the set S_u:

- **Choose initial values** Let x_0 and y_0 be real numbers chosen to be on the Duffing Map attractor [1]
- For $j \in \{1, ...N_{ens}\}$
  - **Create jth perturbed initial condition** Let $x_{0,j} = x_0 + \epsilon_j$, where $\epsilon_j \sim N(0, \sigma^2_{duf})$ and let $y_{0,j} = y_0 + \nu_j$, where $\nu_j \sim N(0, \sigma^2_{duf})$
  - **Evolve forward K steps** Define $r_j = X_K$ where X_K is the Kth iterate of the Duffing Map (6) with initial conditions $X_0 = x_{0,j}$ and $Y_0 = y_{0,j}$.
- Repeat until $j = N_{ens}$
- Define $S_u = \{r_j\}_{j=1}^{N_{ens}}$.

It is now possible to define the following experiments:

Experiment C2.1 Test convergence speed of optimal score parameter.

Parameters for observations: Let $\sigma_u = 0.1$, $N_{obs} = 2^{10}$ and $N_{seed} = 10$. The set S_u is generated from the Duffing map with parameters below.

Parameters for forecast family: Let $\sigma_m \in \Sigma$ where $\Sigma = \bigcup_{i=1}^{3} \Sigma_i$ and Σ_i is defined for integer t in the table below:

Σ_1	$0.05 + 0.01t$	$0 \leq t \leq 2$
Σ_2	$0.08 + 0.0025t$	$0 \leq t \leq 23$
Σ_3	$0.14 + 0.01t$	$0 \leq t \leq 3$

Parameters of Duffing Map: Let $a = 2.75$ and $b = 0.2$, $\sigma_{duf} = 0.01$, $N_{ens} = 2^{12}$ and $x_0 = 0.283995145703728$, $y_0 = 1.092899393566238$, $K = 32$.

[1] For a definition of 'attractor' see the glossary, or Milnor [5]. In practice, potential values are found by running the Duffing map until the values have visually settled down and then choosing any values after this point.

Experiment C2.2.p Find optimal score parameters using different scores.

These experiments are designed to produce an ordering amongst scores.

Parameters for observations: Let $N_{obs} = 2^7$ and $N_{seed} = 10$. Seven different sets $S_1, ...S_7$ are produced from the Duffing map as defined in the table below. Experiment C2.2.p refers to the data set S_p.

Parameters for forecast family: Let $\sigma_m \in \Sigma$ where Σ is defined as for experiment 2.1

Parameters of Duffing Map: Let $a = 2.75$ and $b = 0.2$, $N_{ens} = 2^{12}$, $\sigma_{duf} = 0.01$, $K = 32$ and let the initial conditions be defined in the table below. These underlying data sets are illustrated in figure 1:

Experiment	Underlying dataset S	x_0	y_0
C2.2.1	S_1	-1.409707255606690	-0.952496328839017
C2.2.2	S_2	-1.237472722490239	-1.375416550272213
C2.2.3	S_3	-0.398660021372058	-0.979897892460767
C2.2.4	S_4	0.075153134286194	-0.113837933918633
C2.2.5	S_5	0.135405448765377	0.700349003561764
C2.2.6	S_6	0.283995145703728	1.092899393566238
C2.2.7	S_7	0.374505007140980	0.666289868430975

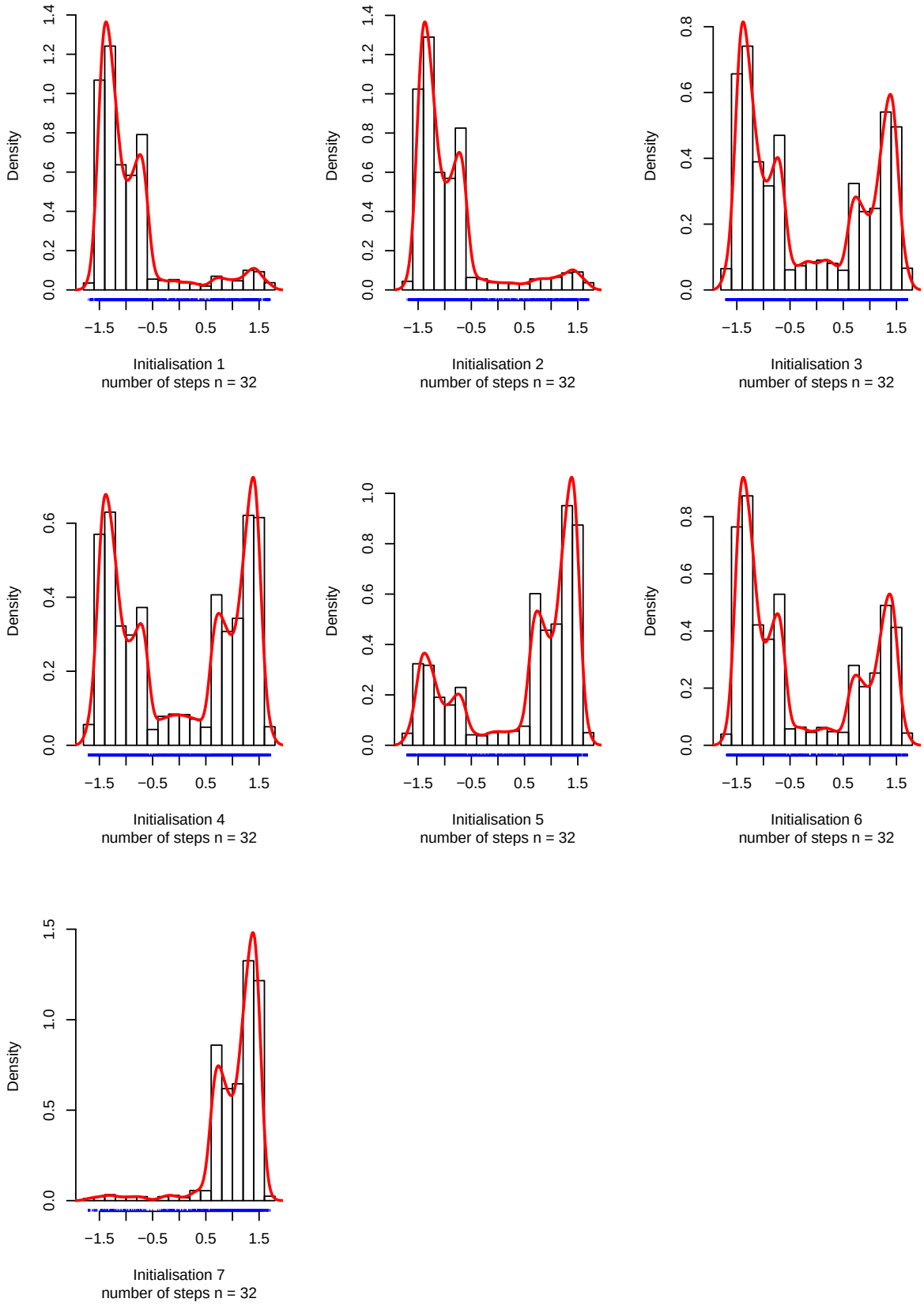

Figure 1: Illustration of the 7 underlying sets S_u drawn from a Duffing Map and shown as a blue tick marks and a histogram. The 'underlying distribution' f_u is illustrated by the red line.

2 Results

Results for experiment C2.1 Figure 2 plots the optimal score estimate $\hat{\sigma_u}$ arising from each of the 10 sets of observations, the plot character $0, ...9$ shows result for each observation set. The plot compares $\hat{\sigma_u}$ derived from the Ignorance score (x -axis) and Proper Linear score (y-axis). The cross hairs in the graphic intersect at the true underlying kernel width width (i.e. $\sigma_u = 0.1$). The shaded double-triangular area shows the region where the score type on the x-axis is closer to the true value than the score on the the y-axis. It is clear that even with $N_{obs} = 2^{10}$ observations (far higher than would be available in many practical situations) there is still some scatter around the true value.

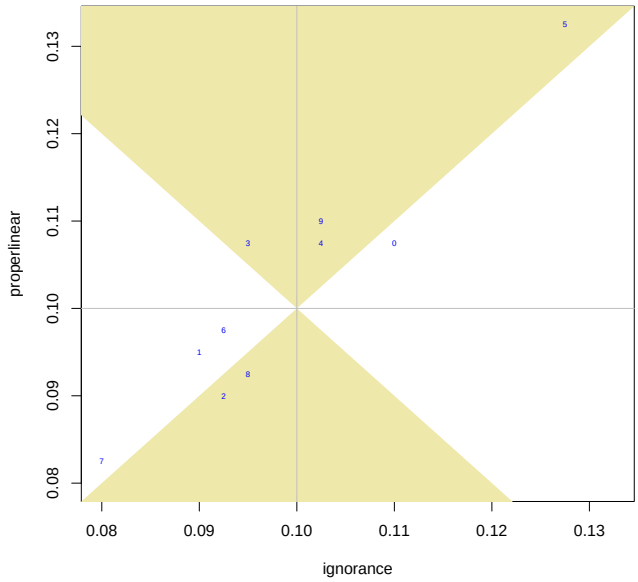

Figure 2: Experiment 2.1: Optimum Score Estimates of underlying kernel width - comparison of Ignorance and Proper Linear Scores. Grey cross hairs indicate the true underlying parameter σ_u. The shaded triangular region illustrates the zone where the parameter value derived by the Ignorance score is closer to the true value than the value derived using the Proper Linear score. The Ignorance derives a parameter that is closer to the truth 6 times out of 10.

By aggregating the observations from all 10 seeds, 10,240 equally likely observations are created. Figure 3 shows that for both score types (Ignorance and Proper Linear) the average skill score across all outcomes is lowest when σ_m is close to the true underlying parameter of 0.1. Even with this many observations the minimum is not at 0.1 which shows that convergence can be slow.

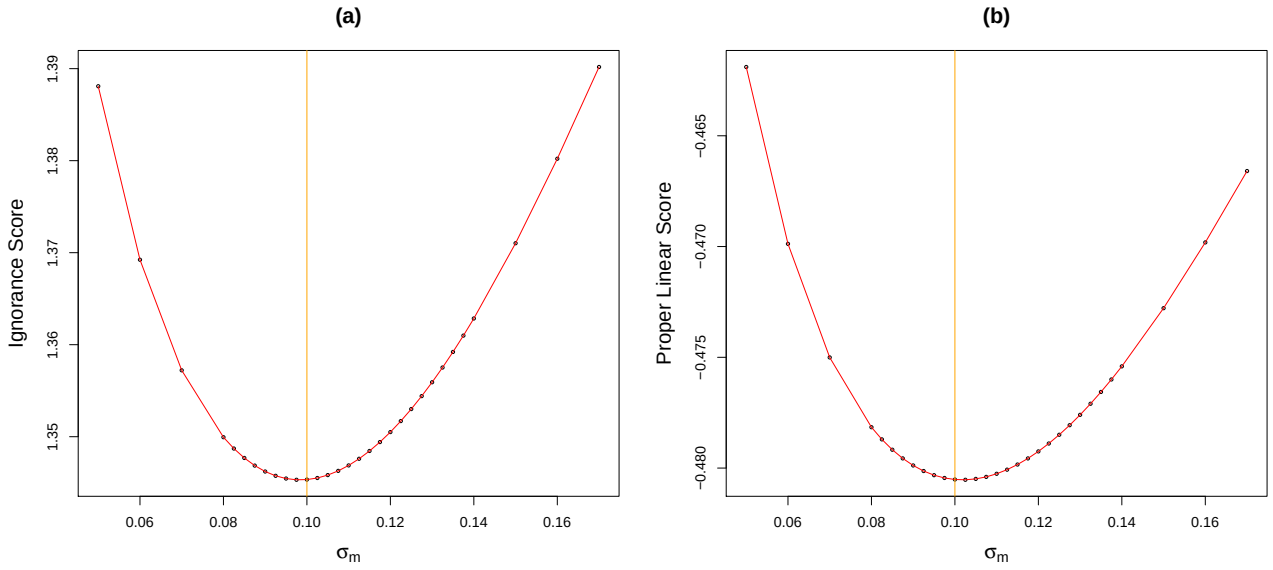

Figure 3: Average skill score values for different trial values of the forecast kernel width. Figure (a) shows the Ignorance score and figure (b) the Proper Linear. The best score (minimum) occurs close to the true underlying parameter value ($\sigma_u = 0.1$, shown with vertical orange line) in each case. Average scores for each value of σ_m are calculated over 10,240 simulated outcomes.

Results for experiment C2.2.k These experiments were carried out for all the scores described in this chapter (except the Brier score because the data is not binary). Figure 4 shows the results of all the experiments when comparing Ignorance and CRPS. The plot characters are of the form $[k,s]$ where k refers to one of the seven data sets S_k and $s \in 0, ...9$ refers to the N_{seed} observations produced by the seed indexed with s. Various conclusions follow from inspection of the graphic:

- Ignorance beats the CRPS score in this example since more of the plot characters are in the white areas reflecting the fact that Ignorance gets closer to σ_u more often than CRPS; specifically, Ignorance does better 54 times and CRPS 16 times; there are no occasions where they tie. (although one looks close, it is marginally off the diagonal);
- On a number of occasions the optimal score estimate for the CRPS score is at the extremes of the range of tested σ_m values. It is certain, however, that the optimal score estimate on those occasions is equal to or further away from the underlying parameter value σ_u. Truncating the values of $\{\sigma_m\}$ tested is therefore generous to the CRPS score by assuming it picked a parameter closer to the underlying kernel width than it would have with a wider mesh. In these cases the Ignorance score estimation method finds parameters that are closer to the truth anyway, so there is no miscounting;
- There is a wide degree of scatter arising from the different seeds;
- There is a wide degree of scatter arising from the various underlying data sets.

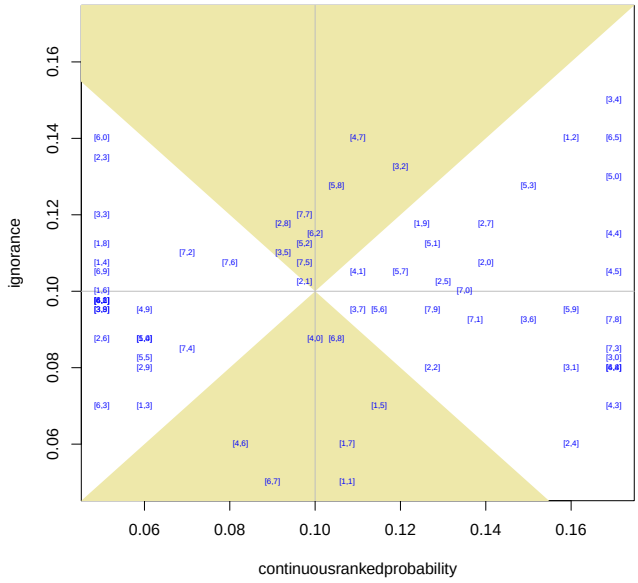

Figure 4: Comparison of Optimum Score Estimates $\hat{\sigma}_u$ for CRPS and Ignorance. The plot character is of the form [k,s] where k refers to one of the seven data sets S_k and $s \in 0, ...9$ refers to the N_{seed} observations produced by the seed indexed with s. The shaded area shows the cases where the optimal score estimator for CRPS is closer to the true underlying parameter than the value derived using the Ignorance score. Note that in 54 of 70 cases the result falls in the white area indicating that Ignorance outperforms the CRPS score.

Score comparison metric Figure 4 suggests a simple format for comparing the scores. For each pairing of scores, where $score_1$ is on the x-axis (say) and $score_2$ on the y-axis: count the number of times the result falls in the white region, call this N_2 ($score_2$ wins) and the shaded region, call this N_1 ($score_1$ wins). Also count any cases where the coordinate is on the diagonal lines, where the scores draw, call this D. Define the $score_1$ ratio, $R_1 = \frac{N_1 + \frac{D}{2}}{N_1 + D + N_2}$. R_1 is a real number between 0.000 and 1.000. A value of 1.000 denotes a case where $score_1$ produces a better optimal score estimate in every case, a value of 0.000 arises when $score_2$ does best every time. When $R_1 \approx 0.5$ the scores either regularly draw, or they each win a similar number of times. Using this method, the following table shows the results of many such tests using the same comparison approach. If $R_1 > 0.5$ then score 1 is said to be 'better' than score 2, or 'score 1 wins'.

Table 1: Skill score comparison results, Experiment C2.2.k

Scores tested	N_1	**D**	N_2	R_1
CRPS vs Ignorance	16	0	54	0.229
CRPS vs MSE	70	0	0	1.000
CRPS vs naive linear	64	1	5	0.921
CRPS vs powerrule1.5	20	4	46	0.314
CRPS vs powerrule2.0	19	4	47	0.300
CRPS vs powerrule2.5	20	4	46	0.314
CRPS vs properlinear	19	4	47	0.300
CRPS vs spherical	21	6	43	0.343
Ignorance vs MSE	70	0	0	1.000
Ignorance vs naivelinear	69	1	0	0.993
Ignorance vs powerrule1.5	37	5	28	0.564
Ignorance vs powerrule2.0	44	2	24	0.643
Ignorance vs powerrule2.5	44	2	24	0.643
Ignorance vs properlinear	44	2	24	0.643
Ignorance vs spherical	41	5	24	0.621
MSE vs naivelinear	0	4	66	0.029
MSE vs powerrule1.5	0	0	70	0.000
MSE vs powerrule2.0	0	0	70	0.000
MSE vs powerrule2.5	0	0	70	0.000
MSE vs properlinear	0	0	70	0.000
MSE vs spherical	0	0	70	0.000
naivelinear vs powerrule1.5	3	0	67	0.043
naivelinear vs powerrule2.0	3	0	67	0.043
naivelinear vs powerrule2.5	3	0	67	0.043
naivelinear vs properlinear	3	0	67	0.043
naivelinear vs spherical	4	0	66	0.057
powerrule1.5 vs powerrule2.0	28	16	26	0.514
powerrule1.5 vs powerrule2.5	32	14	24	0.557
powerrule1.5 vs properlinear	28	16	26	0.514
powerrule1.5 vs spherical	31	19	20	0.579
powerrule2.0 vs powerrule2.5	27	26	17	0.571
powerrule2.0 vs properlinear	0	70	0	0.500
powerrule2.0 vs spherical	29	20	21	0.557
powerrule2.5 vs properlinear	17	26	27	0.429
powerrule2.5 vs spherical	28	16	26	0.514
properlinear vs spherical	29	20	21	0.557

3 Conclusions

Based on this method of comparison the following conclusions can be drawn.

- The Ignorance score does best at choosing parameters that are close to the true kernel width;
- Amongst the Proper scores the CRPS does worst;
- The Improper Naive Linear score never outperforms the Ignorance score but does occasionally beat the other Proper scores;
- The Improper Mean Squared Error score does worst (and the reason for this is described in the appendix);
- The Proper Linear, Power Rule and Spherical scores all have similar performance in this test;
- As α gets smaller the power rule score performs better on this test.
- The RMSE statistic (see appendix) is deeply flawed in this context

Appendix: Behaviour of MSE for a kernel dressed ensemble

The Mean Squared Error (MSE) is defined as:

$$S(p, v) = \int_{-\infty}^{\infty} (v - z)^2 p(z) dz \tag{8}$$

This section contains a proof that the MSE always chooses the smallest kernel width available. Equation 8 is equivalent to:

$$S_{MSE}(p_m, x_i) = (x_i - \tilde{\mu})^2 + \tilde{\sigma}^2 \tag{9}$$

Where $\tilde{\mu}$ is the mean and $\tilde{\sigma}$ is the sd of the forecast p_m (using the same terminology as defined in section 5). Now $\tilde{\mu}$ is calculated as:

$$\tilde{\mu} = \int_{-\infty}^{\infty} x p_m(x) dx = \frac{1}{N\sigma_m} \sum_{i=1}^{N} \int_{-\infty}^{\infty} x \phi(\frac{x - r_i}{\sigma_m}) dx \tag{10}$$

Change variables in the integral using $s = \dfrac{x - r_i}{\sigma_m}$; then:

$$\tilde{\mu} = \frac{1}{N} \sum_{i=1}^{N} \int_{-\infty}^{\infty} (\sigma_m s + r_i) \phi(s) ds = \frac{1}{N} \sum_{i=1}^{N} r_i =: \bar{r} \tag{11}$$

The above derivation makes use of the fact that $\int s\phi(s) = 0$ since ϕ is the PDF of a unit normal distribution and also that $\int \phi(s) = 1$. So $E(p_m) =: \tilde{\mu} = \bar{r}$, i.e. the mean of the forecast is equal to the mean of the ensemble values that gave rise to it. The slight abuse of notation $E(p_m)$ is introduced to help with the next step. To calculate $\tilde{\sigma}$ the formula $\tilde{\sigma}^2 =: VAR(p_m) = E(p_m^2) - (E(p_m))^2$ is used. Now using the same change of variables the following equation arises:

$$E(p_m^2) = \frac{1}{N} \sum_{i=1}^{N} r_i^2 + \sigma_m^2 \tag{12}$$

So that,

$$\tilde{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{N} r_i^2 + \sigma_m^2 - \bar{r}^2 = VAR(r) + \sigma_m^2 \tag{13}$$

Finally substituting the values of $\tilde{\mu}$ and $\tilde{\sigma}$ into equation 14:

$$S_{MSE}(p_m, x_i) = (x_i - \bar{r})^2 + VAR(r) + \sigma_m^2 \tag{14}$$

For a given ensemble the terms $(x_i - \bar{r})$ and $VAR(r)$ are constants, hence S_{MSE} can be minimised by letting $\sigma_m \to 0$ which is exactly the behaviour observed.

RMSE The Root Mean Squared Error (RMSE) of the ensemble mean quantifies the distance between the ensemble mean and its corresponding observed value. It is defined as

$$S_{RMSE}(\bar{x}, X) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\bar{x}(i) - X(i))^2}, \tag{15}$$

where $\bar{x}(i)$ is the ensemble mean of the i^{th} forecast and $X(i)$ is the observed outcome corresponding to forecast i. While easily interpreted as a distance from the observed value, the ensemble mean is somewhat meaningless in that it does not quantify the distance of any particular forecast trajectory or distribution from what actually happened. The RMSE can be generalised as follows:

$$S^*_{RMSE}((p_1, ...p_m), (X_1, ...X_m)) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\int_{-\infty}^{\infty}(X-z)^2 p(i,z)dz.} \tag{16}$$

The original RMSE re-emerges by setting the forecast p as a delta function at the ensemble mean. It should be noted that the RMSE is not a score in the same sense as the others. These are all defined on single forecasts whereas RMSE is defined on multiple forecasts. Note the integral in the summation the Mean Squared Error defined in equation 8. The above discussion of MSE showed that the lowest score can always be attained by reducing the standard deviation to zero in the forecast (i.e. reducing the forecast to a delta function as described above)- surely an unfortunate incentive when probabilistic forecasts are intended to illustrate the uncertainty rather than hide it. The RMSE does this for the forecaster automatically which is undesirable.

References

[1] J. Brocker and L. Smith. From ensemble forecasts to predictive distribution functions. Tellus, 60A:663–678, 2008.

[2] H. Du and L. Smith. Parameter estimation through ignorance. Physical Review E, 2012.

[3] G. Duffing. Erzwungene schwingungen bei veranderlicher eigenfrequenz und ihre technische bedeuting,. G Druck und Verlag von Fridr. Vieweg and Sohn, Braunschweig., 1918.

[4] Gneiting and Raftery. Strictly proper scoring rules, prediction and estimation. Journal of the American Statistical Association, 102(477):359– 376, March 2007.

[5] J. Milnor. On the concept of attractor. Communications in Mathematical Physics, 99:177–195, 1985.