

# Evaluating Feature Relevance: Reducing Bias in Relief

José Bins

Faculdade de Informática  
Pontifícia Universidade Católica (RS)  
Porto Alegre, RS, 90619-900, Brazil  
bins@inf.pucrs.br

Bruce A. Draper

Department of Computer Science  
Colorado State University  
Fort Collins, CO 80523 U.S.A.  
draper@cs.colostate.edu

## Abstract

One of the few algorithms that can evaluate features in very large feature sets is Relief [1, 2]. This paper documents a bias in Relief against non-monotonic features, including Gaussian features, and proposes a modification to Relief that removes the bias.

## 1. Introduction

Feature evaluation is a recurrent problem in computer vision. It occurs implicitly anytime an application programmer selects features for use in a system. It occurs explicitly anytime features are weighted prior to combination. In addition, systems that learn or optimize features must evaluate them in order to measure improvement.

We are interested in feature evaluation in the context of computer vision. This implies very large feature sets, including many potentially irrelevant or redundant features. Most importantly, it implies that independence assumptions among features are almost never valid. Consequently, we need an evaluation technique that is capable of detecting and measuring relevance in the context of very large feature sets.

One of the few algorithms in the literature for evaluating features in the context of large feature sets is Relief [1, 2], as proposed by Kira & Rendell and subsequently modified by Kononenko. Relief measures the relevance of features to training signals. Relief is efficient:  $O(s^2f)$ , where  $s$  is the number of training samples and  $f$  is the number of features. Most importantly, Relief is able to detect relevance even when features interact [3].

This paper reports a bias in Relief against non-monotonic features that peak in the middle when graphed in the obvious way<sup>1</sup>. Since many common features (including features with Gaussian

distributions) are non-monotonic, this greatly reduces the utility of the algorithm. This paper proposes a modification to Relief that removes the bias, and evaluates the modified algorithm on both synthetic and real data.

## 2. Relief

Relief assigns relevance scores (weights) to features based on training samples. For every training sample it finds the nearest sample in feature space that is of the same class as the target sample, and the nearest sample in feature space from another class. These are called “near-hit” and “near-miss” samples, respectively. It then updates the score for each feature based on the difference in the feature values between the target and the near-hit/near-miss samples, under the assumption that a sample of the same class should be more similar to the target than a sample from another class. As modified by Kononenko [2], the algorithm is shown in Figure 1.

## 3. Evaluation

To evaluate Relief on very large feature sets, we need features with known relevance. To this end, we create an artificial regression problem by randomly sampling values between zero and one, and then generating synthetic feature values for these function samples. The simplest features are linear or quadratic functions of the feature value; a linear function is shown in panel (a) of Figure 2. More complex features are linear or quadratic functions of the feature value that are reflected around the middle of the feature range, as shown in panel (b), or Gaussian features, as shown in panel (c). These non-monotonic features are then corrupted with increasing levels of zero-mean Gaussian noise (ranging from 0.00 to 0.50) with standard deviation 1. For reflected features, the amount of feature noise was halved to preserve the signal-to-noise ratio. Figure 2(d-f) shows an example of a (non-reflected) quadratic feature with increasing levels of noise.

---

<sup>1</sup> With feature values along the x-axis and the training signal along the y-axis

```

Let  $S = \{X_1, X_2, \dots, X_n\}$ 
Where  $X_i = \{x_1, x_2, \dots, x_p\}$  is an
instance of  $F = \{f_1, f_2, \dots, f_p\}$ :

For  $i = 1$  to  $n$ , do // for each sample
  Let  $S^+ \subset S$  be the set of samples in the
  same class as  $X_i$ 
  Let  $S^- = S - S^+$ 
  Let  $H \subset S^+$  be the average of the  $k$ 
  nearest neighbors of  $X_i$  in  $S^+$ 
  Let  $M \subset S^-$  be the average of the  $k$ 
  nearest neighbors of  $X_i$  in  $S^-$ 
   $W_i = W_i - \text{diff}(x_i, H)^2 + \text{diff}(x_i, M)^2$ 

For  $i = 1$  to  $p$  // normalize
 $W_i = \left( \frac{\sqrt{|W_i|} - \text{MaxDist}_i}{\text{MaxDist}_i - \text{MinDist}_i} \right)^2 \frac{1}{n}$ 

```

Figure 1. The Relief Algorithm, as modified by Kononenko.

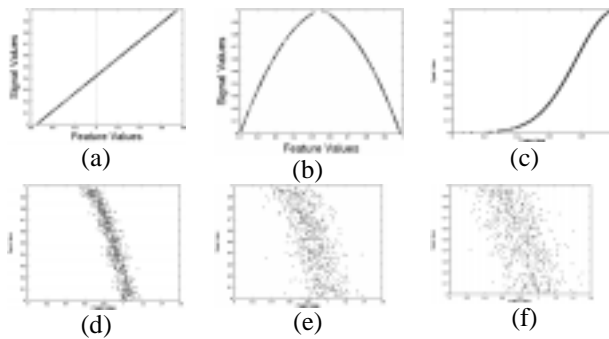


Figure 2: Panels (a-c) show noise-free linear, reflected quadratic, and (half) Gaussian features. Panels (d-f) show the effects of adding noise to a (non-reflected) quadratic feature.

Feat.	No Reflect	Left Reflect	Right Reflect
L1	0.1223	0.0016	0.0020
L2	0.1212	0.0020	0.0011
Q1	0.0977	-0.0006	0.0031
Q2	0.0975	0.0026	-0.0002
Q3	0.0931	0.0019	0.0003
Q4	0.0915	0.0000	0.0022
G1	0.0537	0.0026	-0.0001
G2	0.0543	0.0002	0.0025

Table 1. Weights computed by Relief for equivalent monotonic & non-monotonic features. Features L1 & L2 are linear, Q1-Q4 are quadratic, and G1 & G2 are Gaussian.

To demonstrate the bias in Relief, a data set with reflected and non-reflected versions of the same eight features was created<sup>2</sup>. The average Relief score (over ten trials) for the features and their reflections are

<sup>2</sup> Since a function can be reflected to either the left or right, there are two reflected features for each non-reflected feature.

given in Table 1. On average, in the presence of noise the relevance weight computed by Relief for a reflected feature is only 1.5% of the weight computed for a non-reflected version of the same feature.

This bias can be traced to two main causes: 1) the deterioration of the distance measure; and 2) the non-correlation of similar features. Relief uses distance in feature space to select hits and misses that are similar to the current training instance. If too many irrelevant features are included in a feature set, the assumption that the near-hits and near-misses are similar to the current training sample may fail. Interestingly, non-reflected features are less affected than reflected features by this problem. Figure 3 shows the effect of increasingly degrading the distance measure by adding random features.

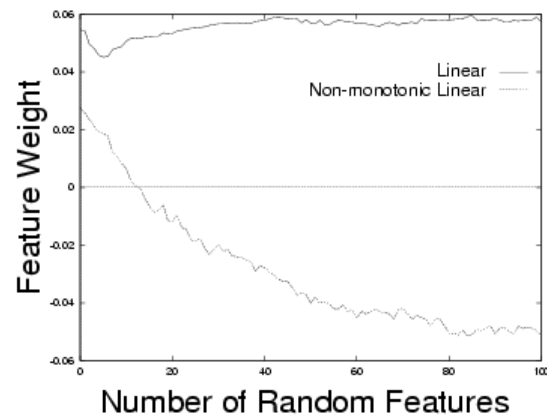
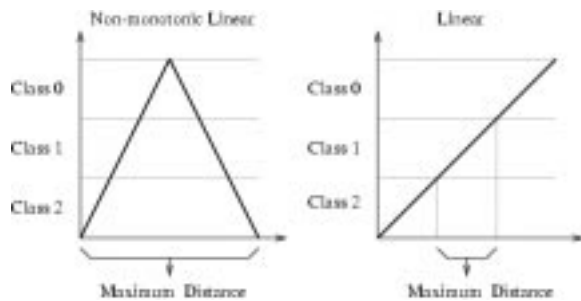


Figure 3: Weights for a linear and a reflected linear function (as computed by Relief) as the number of irrelevant features increases.

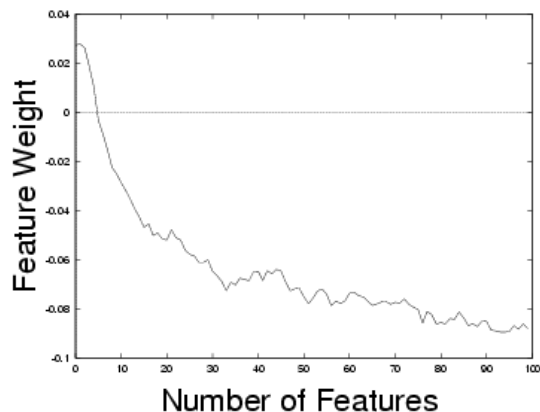
As can be seen, the relevance score for the linear function remains close to its initial value as the number of irrelevant features increases, while the score of the reflected function drops. This occurs because the shape of non-reflected features guarantees that the maximum distance between two features of the same class is bounded by the size of the class. For reflected features, on the other hand, the maximum distance between two samples of the same class is bounded only by the range of the function (see figure 4). In effect, if the sample and its near-miss are on opposite sides of the peak, then they come from different subranges of the feature even though they belong to the same class. As a result, the similarity assumption is violated and the feature looks artificially bad.

The other identifiable source of bias is the lack of correlation between similar features. Two non-reflected functions computed using the same learning signal correlate (or anti-correlate) perfectly, except

for noise. On the other hand, two reflected functions computed over the same signal may not correlate, since there are at least two possible feature values matching any given signal value. This again degrades the distance measure's ability to select "similar" samples, since two samples that have similar values for the signal and for one feature may not have similar values for another feature. In essence, the presence of many non-reflected features will degrade the quality of the distance function, even if all of the non-reflected features are meaningful. Figure 5 shows how the weight computed for a reflected linear feature drops as more features based on the same underlying function are included.



**Figure 4. Maximum distances for monotonic and non-monotonic features.**



**Figure 5: Weight (as computed by Relief) of a reflected linear function as the number of similar features is increased.**

### 3. Eliminating the Bias

#### 3.1 Algorithm Modifications

To eliminate the bias against reflected functions three modifications of Relief are proposed: 1) identify the reflected functions and compute the reflection point which divides the function into two distributions; 2) update each feature using only hits and misses that are on the same side of the reflection point; and 3) weight differences between samples and their hits/misses using the reflection point.

Many methods can be used to identify peaks and valleys in a signal. The specific method of peak detection is not significant; we use the one described in [5]. When finding peaks in functions, it is important not to miss the presence of a peak (false negative), or else the bias will remain. Inserting a false peak, on the other hand, has only a small effect on a feature's relevance score.

When updating the relevance weights, only hits and misses on the same side of the reflection point as the target sample are considered. As a result, fewer hits and misses may be used for some samples than others. In extreme cases, some samples may not contribute any hits or misses. That implies that each class weight<sup>3</sup> must be recomputed for each sample.

Finally, Relief normalizes the differences between samples and hits and misses into the interval [0,1]. By restricting hits and misses to be on the same side of the peak as the target sample, our version of Relief reduces the maximum and average distances between samples for the reflected features. To compensate for this, hit and miss distances are divided by the size of the distribution in which they reside.

#### 3.4 Results on synthetic data

The modified algorithm is re-tested for bias, and the result is shown in Table 2 (this compares to Table 1). The maximum difference between the relevance of a feature and the relevance of its reflection is less than 0.009, or approximately 7% of its relevance score. When noisy features are used this difference is 0.025, or 27%. By way of comparison, Kononenko's version of Relief had differences of 98.2% without noise and 98.3% with noise. We also repeated the tests of figures 3 and 5 with the modified algorithm, verifying that reflected functions scores are no longer degraded by including irrelevant features or other reflected features (see figures 6 and 7).

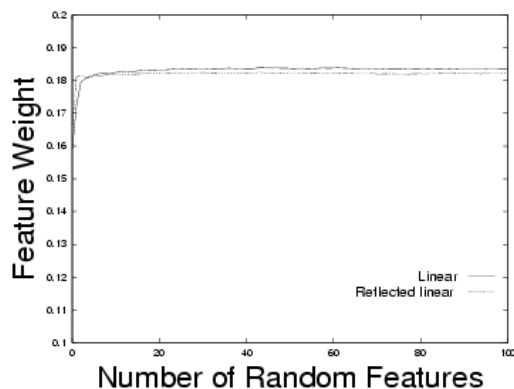
To test how sensitive the modified algorithm is to errors in the peak detector, we assign false reflection points to the non-reflected functions (located at 0.5). The functions are overestimated by amounts ranging from 20% (linear functions) to 60% (Gaussian functions). This represents a bias in favor of non-reflected feature misinterpreted as reflected features, but the bias is much less than the previous bias against reflected features (which was in excess of 5,000%). As a result, even if every non-reflected

<sup>3</sup> For Kononenko's version of Relief these weights are the a-priory probability of each class. We normalize this weight using only the classes that effectively contribute with hits/misses

feature is assigned an erroneous reflection point, the bias is still less than for previous versions of Relief.

### 3.5 Results on real data

When testing on real data, the true relevance of the features is not known *a-priori*. This eliminates the methodology used to evaluate the algorithm on synthetic features. Instead, we use neural nets to test if the modified version of Relief is better than Kononenko's original. We created 300 pairs of features  $\langle A_i, B_i \rangle$ , such.  $A_i$  is a randomly chosen feature where the difference in score between the original and modified versions of Relief exceeds 0.02. (These features are assumed to have internal peaks, since non-peaked features produce similar scores in both versions of the algorithm.)  $B_i$  is a randomly selected feature that is better than  $A_i$  according to the unmodified Relief score and worse than  $A_i$  according to the modified Relief score. A neural net was then trained to predict the training signal from  $A_i$ , and another was trained to predict the training signal from  $B_i$ . The MSE's of the nets were then compared. We considered the test a success if the net for feature A was better than the net for feature B. We achieved a success rate of 60.6% (91/150). According to the binomial distribution, the probability of this result (or better) if the two methods are equivalent is less than 1%. We conclude that the modified version of Relief is better than the original version for this data.

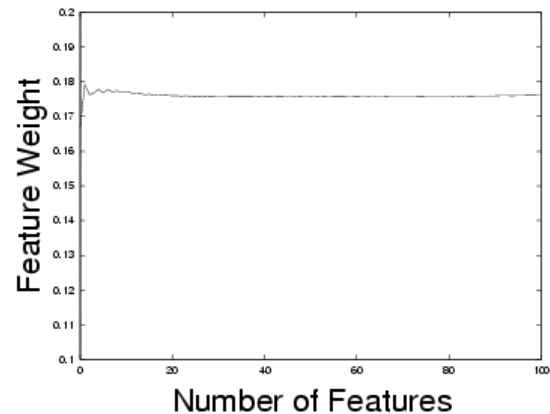


**Figure 6. Modified Relief weight values for a linear and reflected linear feature with increasing numbers of random features.**

### 4. Conclusion

We have identified two sources of bias in Relief against non-monotonic features, and have proposed a simple modification to eliminate them. Our solution does not address multi-peaked functions, but we do not think we could reliably identify them anyway,

given the noise in our data. Instead, we modify Relief so that it can evaluate single mode features as well as monotonic ones.



**Figure 7. Modified Relief weight values for a linear and reflected linear feature with increasing numbers of similar features.**

Feat.	No Reflect	Left Reflect	Right Reflect
L1	0.1210	0.1135	0.1146
L2	0.1213	0.1141	0.1151
Q1	0.0991	0.0801	0.0694
Q2	0.0972	0.0742	0.0776
Q3	0.0920	0.0738	0.0730
Q4	0.0921	0.0732	0.0662
G1	0.0535	0.0470	0.0536
G2	0.0529	0.0539	0.0475

**Table 2. Weights computed by modified Relief for equivalent monotonic & non-monotonic features.**

### 5. References

1. Kira, K. and L.A. Rendell. *A Practical Approach to Feature Selection*. in *9th International Workshop on Machine Intelligence*. 1992. Aberdeen, Scotland: Morgan-Kaufman.
2. Kononenko, I. *Estimation Attributes: Analysis and Extensions of RELIEF*. in *European Conference on Machine Learning*. 1994. Catania, Italy: Springer-Verlag.
3. Caruana, R. and D. Freitag. *Greedy Attribute Selection*. in *International Conference on Machine Learning*. 1994: Morgan Kaufman.
4. Bins, J. and B.A. Draper. *Feature Selection from Huge Feature Sets*. in *International Conference on Computer Vision*. 2001. Vancouver: IEEE Vol II, p. 159-165.
5. Bins, J., *Feature Selection of Huge Feature Sets in the Context of Computer Vision*, Ph.D. thesis, 2000, Colorado State University: Fort Collins, CO. p. 156.

