Bayes and Pseudo-Bayes Estimates of Conditional Probabilities and Their Reliability

James Cussens

Centre for Logic and Probability in IT
King's College, Strand, London, WC2R 2LS, UK
Phone: +44 71 873 2291, Fax: +44 71 836 1799
email: j.cussens@uk.ac.kcl.cc.elm

Abstract. Various ways of estimating probabilities, mainly within the Bayesian framework, are discussed. Their relevance and application to machine learning is given, and their relative performance empirically evaluated. A method of accounting for noisy data is given and also applied. The reliability of estimates is measured by a significance measure, which is also empirically tested. We briefly discuss the use of likelihood ratio as a significance measure.

1 Introduction

The importance of conditional probability estimation in machine learning has been rightly stressed in, for example, [4]. Learning algorithms generally output rules of the form $C \leftarrow A$. An obvious measure of the quality of such a rule is simply $P(C|A) = p$, the probability that a randomly chosen example, given that it is covered by the rule, is correctly classified by the rule.[1] We will sometimes informally refer to p as the *probability of the rule* $C \leftarrow A$.

The exact value of p will generally be unknown—but it can be estimated. Standard Bayesian techniques for the estimation of probabilities are well known [13,25,15]. In recent years they have been used successfully in machine learning [4,5,9,8,16] using m-estimation. In this paper, we use, as well as standard Bayes estimates, *pseudo-Bayes* estimates of conditional probabilities, drawing heavily on the work of Bishop, Fienberg and Holland [2].

Given that we have an estimate for $P(C|A) = p$, it is useful to have a measure of the reliability of this estimate. This reliability is often termed the *significance* of the rule $C \leftarrow A$. A measure of significance is proposed in Sect. 7, and in Sect. 8.4, we map significance against estimate accuracy to check that it is an adequate measure of estimate reliability.

2 Bayesian Estimation of Probabilities

In the Bayesian approach to the estimation of an unknown quantity, for example the probability p, a *prior* distribution is selected which represents information

[1] Throughout this paper, $P(X)$ represents the probability that a randomly chosen example satisfies X.

concerning possible values of p. As we gather data relevant to the value of p, this distribution is updated via a continuous version of Bayes theorem to give a *posterior* distribution. We can then take the mean of this posterior distribution as a point estimate of p. To estimate a probability, the prior is invariably constrained to be a *beta distribution*, since this makes updating it extremely easy. Beta distributions are parameterised by two values r_0 and n_0 ($n_0 > r_0 > 0$) as follows:

$$f_{r_0,n_0}(x) = \begin{cases} \frac{(n_0-1)!}{(r_0-1)!(n_0-r_0-1)!} x^{r_0-1}(1-x)^{n_0-r_0-1} & \text{if } 0 \leq x \leq 1 \\ 0 & \text{elsewhere} \end{cases} \quad (1)$$

The mean of such a distribution is r_0/n_0. n_0 determines the spread or variance of the distribution, the distribution becoming flatter as n_0 decreases; consequently n_0 has been called a 'flattening constant' [10] or a 'smoothing constant' [2,23].

Assume our prior distribution is parameterised by r_0 and n_0. Suppose that out of n examples, the event whose probability we are attempting to estimate, occurs r times. The posterior distribution is then a beta distribution with parameters $r_1 = r_0 + r$ and $n_1 = n_0 + n$. The mean of this distribution is $\frac{r_1}{n_1} = \frac{r_0+r}{n_0+n}$ and we can use this to estimate the probability in question.

3 Choosing the Mean of the Prior Distribution

On the choice of mean for the prior distribution, I concur with the value used in m-estimation. If we are to estimate $P(C|A) = p$, we use $P(C)$ as the prior mean. This choice can be justified as follows.

Suppose the domain about which we are learning is finite and of size D. Hence the number of examples which satisfy A is $D.P(A)$. Identify A, C and $A \wedge C$ with the sets of examples which satisfy them. Prior to gathering any training data, we have no indication about which particular examples fall within A, hence we model this by supposing the members of A are chosen at random. We have $D.P(A)$ elements to chose, at random, from a finite domain where the proportion of elements in C is $P(C)$. Such a scenario is modelled by the *hypergeometric distribution*, so the mean number of elements of A which fall within C, i.e. the mean number in $A \wedge C$, will be $D.P(A).P(C)$ [11, p.139]. This means the mean value of $P(A \wedge C)$ is $D.P(A).P(C)/D = P(A).P(C)$ and finally the mean value of $P(C|A) = P(A \wedge C)/P(A)$ is $P(C)$.

Note, however, that $P(C)$ is generally not known—so it too must be estimated. Fortunately, since the whole of the training data can be used as a sample with which to estimate $P(C)$, we can get very reliable estimates of it. In this paper we use Laplace estimation (see below) to estimate $P(C)$.

4 Selecting the Prior Distribution

Recall that the prior distribution is fixed by the values of r_0 and n_0, and that the prior mean is given by r_0/n_0. So, from above, we have that $r_0/n_0 = P(C) \leftrightarrow r_0 = n_0.P(C)$. All that remains is to fix a value for n_0.

4.1 Bayesian Estimation

m-Estimation: Finding n_0 Experimentally In m-estimation [4,5,9,8,16], the value n_0 above is designated by m. Bayesian estimation is then done in the normal way, giving the usual estimate $(r + r_0)/(n + n_0)$ for p, where r and n are as above. With $r_0 = n_0.P(C)$ and $n_0 = m$, the estimate becomes

$$p_m = \frac{r + m.P(C)}{n + m} = \left(\frac{n}{n+m}\right)\left(\frac{r}{n}\right) + \left(\frac{m}{n+m}\right)P(C) \qquad (2)$$

The best value for m in a given domain is currently found experimentally—'...several different values for m should be applied. At the end, after measuring the performance of the induced classifiers, the value of the m which gives the best performance can be selected' [9]. Such an approach has been found to be very successful. The extra effort required to find the best m value is outweighed by better estimates once it is found and put to use.

As can be seen from (2), the value m 'controls the balance between relative frequency and prior probability' [9]. A high value of m indicates that we are very confident in our prior estimate for $P(C|A)$, namely $P(C)$. The variance of the prior distribution is then small. If noise is expected in the examples, m is set higher so that the (noisy) value for r/n plays less of a rôle in the final estimate.

Non-Experimentally Derived Values for n_0 Various non-experimentally derived values for n_0 have been discussed in the statistical literature, a list can be found in [10]. In this paper we use the a priori value $n_0 = 1$ and also the data-dependent value $n_0 = \sqrt{n}$. For the properties of these values see [2, p.407].

4.2 Pseudo-Bayes Estimation

Pseudo-Bayes estimation can be seen as a variety of Bayesian estimation where n_0 is strongly data-dependent, being a function of both r and n. This explains what is 'pseudo' about it. The prior distribution, determined by n_0, is no longer truly prior, since it contains relative frequency information from the training data. By the same argument, one can see $n_0 = \sqrt{n}$ as semi-pseudo-Bayes, since it depends on n; however, it is generally categorised as a plain Bayesian estimator.

There is an infinite family of pseudo-Bayes estimators, as described in [23]. Here, however, we consider only the one described in [2]. There, the value $\hat{K}$, which is the maximum likelihood estimator of the optimal flattening constant K, plays the rôle of n_0. It is defined as follows:

$$\hat{K} = \frac{\frac{r}{n}(1 - \frac{r}{n})}{(P(C) - \frac{r}{n})^2} = \frac{r(n-r)}{(n.P(C) - r)^2} \qquad (3)$$

The pseudo-Bayes estimator of p is hence given by

$$p^* = \frac{r + \hat{K}.P(C)}{n + \hat{K}} = \left(\frac{n}{n+\hat{K}}\right)\left(\frac{r}{n}\right) + \left(\frac{\hat{K}}{n+\hat{K}}\right)P(C) \qquad (4)$$

5 A Non-Bayesian Approach to Estimation

In all the experiments given below, we have that a set of training data consisting of N examples is drawn at random from the domain. n examples of this training data are in A, s are in C and r are in $A \wedge C$—so the rule $C \leftarrow A$ has training accuracy r/n. Assume the domain in question contains D examples in total. After the collection of training data, there are $D.P(A) - n$ examples left in the domain which are members of A. Clearly we do not know how many of these are also in C, otherwise our estimation problem would be solved. However, by assuming that the remaining members of A are distributed at random (which models our ignorance about A), we can easily calculate the *mean* number number which fall in C.

Our domain has $D - N$ members left from which to pick. The proportion left in C is $\frac{D.P(C)-s}{D-N}$ and we are to pick $D.P(A)-n$ examples at random. The number of these examples that are in C is governed by the hypergeometric distribution, since we are sampling examples without replacement from a finite domain. The mean number of new examples in A that are also in C is thus given by

$$(D.P(A) - n)\frac{D.P(C) - s}{D - N} \tag{5}$$

We can now add in the original r examples, to get the mean number of examples in the domain which satisfy $A \wedge C$. Dividing by D, gives the mean value for $P(A \wedge C)$; and, finally, dividing by $P(A)$ gives the following mean value for $P(C|A)$.

$$\frac{(D.P(A) - n)(D.P(C) - s) + r(D - N)}{D(D - N)P(A)} \tag{6}$$

This gives an alternative non-Bayesian way of estimating p and has been tested by the author empirically, giving generally poor results (Table 4). One big problem is that to get optimal estimates, the value of D had to be set far lower than the known sizes of domains.

The most likely explanation for the poor performance of (6) as an estimator, is that the modelling assumption that A is randomly distributed will generally not be appropriate. The choice of antecedents for learnt rules is, in fact, subject to considerable bias depending on a number of factors [20,7,14,24]. Incorporating such bias into the estimation of the probability of rules is an important research task, but is beyond the scope of this paper.

6 Accounting for Noise

6.1 Winkler and Franklin's Approach to Accounting for Noise in Bayesian Estimation

Rather than account for noise by increasing the value of n_0, we alter the method of updating the prior distribution. Winkler and Franklin [27] show how to account for noise using both alternatives. For both cases, Winkler and Franklin

calculate approximations to the exact posterior distribution[2] and find that with both methods, the approximation is generally good.

We adopt the 'modified-updating' approach because it is consonant with the philosophy behind Bayesian inference. The presence of noise means that the observed values of empirical data are unreliable. Since empirical data affects estimation via the updating of the distribution, it is natural that noise be accounted for in the updating procedure. Prior distributions, on the other hand, are meant to represent prior knowledge.[3] There is no particular reason for this prior knowledge to be affected by the noise level in the training set. Another advantage of this approach is that the parameters that emerge have a natural interpretation in terms of reduced sample size, as we shall see below. The following approach is, with notational changes, that of Winkler, as given in [26].

We have, for the rule $C \leftarrow A$, a set of training examples with r 'successes' $(A \wedge C)$ and $n - r$ 'failures' $(A \wedge \neg C)$. Let the probability that a success is misclassified as a failure be ϕ and the probability that a failure is misclassified as a success be ψ. The maximum likelihood estimator of p (that value of p which makes r successes and $n - r$ failures most likely) is $\hat{p}$, where $\hat{p}_B = r/n$ and,

$$\hat{p} = \begin{cases} 0 & \text{if } r/n < \psi \\ (\hat{p}_B - \psi)/(1 - \phi - \psi) & \text{if } \psi \leq r/n \leq 1 - \phi \\ 1 & \text{if } r/n > 1 - \phi \end{cases} \tag{7}$$

Note that setting $\phi = \psi = 0$ (the noise-free case) entails $\hat{p} = \hat{p}_B = r/n$, as expected, since r/n, relative frequency, is the maximum likelihood estimator for p in the absence of noise.

Winkler shows that, as long as

$$\psi < r/n < 1 - \phi \tag{8}$$

we can set

$$n^* = \frac{n\hat{p}(1 - \hat{p})}{(\hat{p} + c_1)(1 - \hat{p} + c_2)} \tag{9}$$

and

$$r^* = n^*\hat{p} \tag{10}$$

where

$$c_1 = \frac{\psi}{1 - \phi - \psi}, \; c_2 = \frac{\phi}{1 - \phi - \psi} \tag{11}$$

The observation of r successes from n examples with noise parameters ϕ and ψ, is then approximately[4] equivalent to having r^* successes out of n^* *noise-free* examples. We have, $n^* < n$, as long as the initial condition (8) is satisfied. So '[a]s anticipated, then, the noise leads to an effective reduction in sample size.' [26]. We can then view $1 - (n^*/n)$ as 'a rough measure of the proportion of information lost as a result of noise' [26].

[2] An *exact* posterior distribution in the presence of known amounts of noise is possible to calculate, but the calculations tend to be cumbersome [27].

[3] The qualities of pseudo-Bayes estimates show that it is sometimes worth 'cheating' in this respect!

[4] The approximation is particularly good when $\phi = \psi$, which is the case we will consider. See [27] for various graphs showing the precision of the approximation.

6.2 Accounting for Noise in Machine Learning

If noise accords to the *Classification Noise Process* model as described in [1], then Winkler and Franklin's method is directly applicable. In this model we assume that examples are generated by an oracle, and that the examples are subject to independent and random misclassification with some probability $\eta < 1/2$. This is clearly equivalent to the situation described above with $\phi = \psi = \eta$. Quinlan in [19] introduces $\eta \times 100\%$ noise by replacing, with probability η, the correct classification of an example by a random one. In this case the above is applicable with $\phi = \psi = \frac{1}{2}\eta$.

For the noisy data used in this paper, a measured amount of noise was introduced using the *Classification Noise Process*, so we set $\phi = \psi = \eta$. Such noise affects the observed value of r, whilst having no effect on the observed value of n (we have *classification noise* but no *description noise*—these two forms of noise are discussed in [21]). Also, since the noise was artificially introduced, the value η is known. We do not the address the issue of estimating unknown noise levels here—for a discussion of this issue see [1,22].

Rewriting the above equations for n^* and r^* in terms of r, n and η, gives the following:

$$n^* = n\frac{(r-\eta n)(n-r-\eta n)}{r(n-r)} \tag{12}$$

$$r^* = n^*\hat{p} = \frac{(r-\eta n)^2(n-r-\eta n)}{(1-2\eta)r(n-r)} \tag{13}$$

The above equations are only applicable if $\eta < r/n < 1-\eta$, an inequality that does not always hold. However, it is clear that $n^* \to 0$ as $r/n \to 1-\eta$, i.e. the amount of information lost due to noise tends to be total as r/n approaches $1-\eta$. Hence for values of r/n such that $r/n \geq 1-\eta$, we set n^* and subsequently r^* to zero.[5] In this case, all Bayesian and pseudo-Bayesian estimates equal the prior probability $P(C)$, reflecting the fact that we have gained no real information from the data.

To summarise, except for the cases mentioned immediately above, given $\eta \times 100\%$ noise and observing r successes out of n examples covered by the rule in question $(C \leftarrow A)$, we can estimate the probability $P(C|A)$ using Bayesian estimation with n^* and r^* as updating parameters.

7 The Reliability of Estimates

It is clearly desirable not only to have good estimates of probabilities, but also some measure of how good a given estimate is. We shall use the posterior distribution to find $P(|\text{estimate} - p| < t)$, which is the (posterior) probability that the estimate is within a certain distance (t) of the true probability. This value is found by integrating the posterior distribution between (estimate $- t$) and

[5] The case when $r/n \leq \eta$ can be dealt with similarly, but is not of interest for machine learning.

(estimate$+t$).[6] We shall call $P(|\text{estimate}-p| < t)$ a *significance measure*. In our experiments, t was set to 0.025, since this gave convenient significance values.

8 Empirical Results

8.1 Experimental Procedure

The data used here is that used in [18]. There, the value of *HP Compression* as a significance measure was considered. Unfortunately we do not have space to examine this issue here.

Rules were learnt using Golem, an Inductive Logic Programming algorithm [17]. The learning domains were as follows

PROTEINS Prediction of protein secondary structure. We have rules which predict when a given residue is part of an α-helix.

DRUGS Modelling drug structure-activity relationships. Rules relate the structure of a drug to its chemical activity.

KRK Rules for characterising illegality in two Kings and a Rook chess endgames. We do estimation for the cases of 5%, 10% and 20% added noise.

The estimates used can be split into two groups. In the first group, estimation is undertaken without reference to testing data (Table 1) and in the second, testing data is used to find the best possible estimate (Table 2).

Table 1. Non-empirically derived estimators

p^*	Pseudo-Bayes estimation as given by (4).
$p_{n_0=1}$	Estimation by setting $n_0 = 1$. This approach is discussed in [2].
$p_{n_0=\sqrt{n}}$	Estimation by setting $n_0 = \sqrt{n}$. This approach is also discussed in [2].
p_L	Laplace estimation. This amounts to choosing the uniform distribution as a prior. Estimates are given by $(r+1)/(n+2)$.
$\hat{p}$	Training accuracy as an estimate. This is the same as fixing $n_0 = 0$. It is the maximum likelihood estimator.

In a given domain, the probability $P(C|A)$ for each rule $C \leftarrow A$ was estimated using all of the above estimates. The significance of each rule was also calculated for the Bayes/pseudo-Bayes estimates, as described above. Noise, for the Bayes/pseudo-Bayes estimates, was accounted for using Winkler's approach.

[6] In the empirical results that follow, this integration was done by a Numerical Analysis Group subroutine within a Fortran program. Indeed, all results are the output of various Fortran programs.

Table 2. Empirically derived estimators

p_m m-estimation. The best value for m was found using the testing data.

p_D Estimation using (6). The best value for D was found using the testing data.

Some Special Cases After we have accounted for noise, it sometimes occurs that the values for n and r are reduced to zero (see above). In this case, $p_{n_0=\sqrt{n}}$ and $\hat{p}$ are undefined. Since $n = 0$ is equivalent to having no training data, we use the prior mean $P(C)$ as an estimate in these cases, and since this estimate is based on effectively no training data, significance is set of zero.

From (3), we see that, $r/n = 1 \Rightarrow \hat{K} = 0$. Recall that $\hat{K}$ is used as a flattening constant n_0 for a prior beta distribution, and that we must have $n_0 > 0$. In this case, we consider what happens as $\hat{K} \to 0$ and $r/n \to 1$; we have that $p^* \to r/n = 1$ and significance $\to 1$. So in this case, we set both estimate and significance to 1. This seems the only consistent way of dealing with this case, but it gives rise to anomalous behaviour. For example, if $r = n = 1$, p^* returns an estimate of 1 with maximum significance! This reveals a clear weakness with pseudo-Bayesian estimation as used in this paper.

There are two more special cases for pseudo-Bayes. If $r/n = P(C)$, then $\hat{K}$ becomes infinite. Similarly to above, we consider what happens as $\hat{K} \to \infty$ and set p^* to $P(C)$ and significance to 1. Finally if $r = n = 0$, $\hat{K}$ is undefined. Now $\hat{K} = 1$ when $n = 0$, but $r \neq 0$ so we set $\hat{K} = 1$ in this case, giving an estimate of $P(C)$, as above.

8.2 Looking at Mean Squared Error

Our goal in estimation, since we are using posterior *means* as point estimates, is to minimise $(\text{estimate} - p)^2$ (see [3, Appendix A5.6]). Since the value p is unknown, this expression can not be evaluated. So, in the following, the true probability value, p, for any rule is simply estimated by the *testing accuracy* of that rule. In other words, we use relative frequency in the testing set as a (second) estimator of p. Although, as our results show, relative frequency is a poor estimator, we use testing accuracy since this is the standard method of evaluating rule performance in machine learning.

For each domain and each choice of estimate, we found the mean value of $(\text{estimate} - \text{training accuracy})^2$ over all the rules from a given domain. The results for estimates which used only the training data are given in Table 3. Those for empirically tunable estimates are given in Table 4, with the optimal parameter values which gave these results. In both tables, the estimate with smallest mean squared error is in **bold**.

We would like more domains on which to test the various estimates, but there are already some significant results. If we exclude the Drugs domain (for why, see below), $\hat{p}$ is always the worst estimator of all, and $p_{n_0=\sqrt{n}}$ is always the best from amongst those that take $P(C)$ into account. The probability of this

Table 3. Mean squared errors for p^*, $p_{n_0=1}$, $p_{n_0=\sqrt{n}}$, p_L and $\hat{p}$

	p^*	$p_{n_0=1}$	$p_{n_0=\sqrt{n}}$	p_L	$\hat{p}$
Proteins	7.491×10^{-2}	7.633×10^{-2}	$\mathbf{6.038 \times 10^{-2}}$	7.040×10^{-2}	8.269×10^{-2}
Drugs	3.204×10^{-3}	3.162×10^{-3}	7.079×10^{-3}	3.883×10^{-3}	$\mathbf{2.808 \times 10^{-3}}$
KRK (5%)	8.125×10^{-3}	9.758×10^{-3}	$\mathbf{7.813 \times 10^{-3}}$	9.920×10^{-3}	1.0876×10^{-2}
KRK (10%)	6.796×10^{-2}	6.458×10^{-2}	6.062×10^{-2}	$\mathbf{4.324 \times 10^{-2}}$	7.400×10^{-2}
KRK (20%)	3.464×10^{-2}	3.538×10^{-2}	$\mathbf{2.718 \times 10^{-2}}$	3.300×10^{-2}	4.982×10^{-2}

Table 4. Mean squared errors for p_m and p_D

	p_m	p_D
Proteins	2.703×10^{-2} ($m = 114$)	$\mathbf{2.610 \times 10^{-2}}$ ($D = 1 \times 10^5$)
Drugs	$\mathbf{2.808 \times 10^{-3}}$ ($m = 0$)	4.251×10^{-3} ($D = 1743$)
KRK (5%)	$\mathbf{8.545 \times 10^{-3}}$ ($m = 4$)	3.327×10^{-1} ($D = 998$)
KRK (10%)	6.144×10^{-2} ($m = 3$)	$\mathbf{4.216 \times 10^{-2}}$ ($D = 1000$)
KRK (20%)	$\mathbf{1.884 \times 10^{-2}}$ ($m = 10$)	2.831×10^{-2} ($D = 1001$)

occurring by chance is very small—these are significant results. (This could be proved rigourously using nonparametric rank-order statistical tests [12]). On the other hand, there is not enough evidence to say that p_m is superior to the other Bayes/pseudo-Bayes estimates, since it only the best estimate of this class three times out of five.

8.3 Domain Peculiarities

In the protein domain, the data used was unsatisfactory, since the training and testing sets had significantly different proportions of positive examples (residues which really were in α-helices). This means that estimates derived from training data could be expected to be unsuccessful on the given testing data. This probably explains why p_D performed well here, as opposed to most of the other domains.

The Drugs domain is remarkable in that many rules have 100% accuracy on training or test data, and frequently on both. This explains why $\hat{p}$ (equivalently m-estimation with $m = 0$) was the most successful on this domain, whilst being the worst estimator on all other domains. $p_{n_0=\sqrt{n}}$ did particularly badly on the Drugs domain. Since in the Drugs domain, the prior mean was exactly 1/2, $p_{n_0=\sqrt{n}}$ is the unique constant risk minimax estimator [2, p.407]. This estimator has high expected error when the true value of the probability to be estimated is

close to 0 or 1 (see [2, Fig. 12.4-1, p.416]) and this explains its poor performance in the Drugs domain.

The KRK (10%) domain is surprising since all estimators perform more badly there than on the KRK (20%) domain, and the optimal m value is smaller than on the KRK (5%) domain. Again p_D is most successful here, where all other estimators do badly. What has happened here is that Golem has generated a number of rules of low significance that have very large errors, thus increasing the total mean squared error considerably. If we cut out these rules, and look, for example, at only the 20 most significant rules for KRK(10%) and KRK(20%), we find that estimators have lower error on the KRK(10%) domain, as expected (see Fig. 1). Relative frequency ($\hat{p}$) performs badly in the KRK domains, since it cannot account for noise. The Bayes and pseudo-Bayes estimators react to noise by giving the prior mean, $P(C)$, greater weight. In contrast, relative frequency can take no account of $P(C)$.

8.4 Comparing Squared Error and Significance

In Fig. 1, we compare the performance of the four Bayes/pseudo-Bayes estimators p_m, $p_{n_0=1}$, p^* and $p_{n_0=\sqrt{n}}$ on the three KRK domains. We plot their mean squared errors over subsets of the rule base, as we progressively eliminate less significant rules. We will represent the number of rules left on the x-axis, and mean squared error on the y-axis, so the rightmost point for each estimator represents the mean squared error over all the rules. These graphs demonstrate a number of important points.

1. The significance measure is working as it should. Significant rules have lower mean squared error, whatever the particular estimator.
2. The value of m used in m-estimation has been chosen to be optimal over the whole rule set. Such a value of m is successful over the whole set because it has been chosen to be reasonably successful on even quite insignificant rules (rules where an estimate of the probability is unreliable). The graphs for 20% noise demonstrate this ably: the errors for less significant rules are all large with the exception of those estimated by $m = 10$. On significant rules, however, m-estimation is not always superior.
3. The nth most significant rule is often the same for all estimates, as can be seen by the similar shape of the graphs for each estimator.

Our significance measure is meant to give the probability that an estimate diverges from the true probability by a given amount. We now put this to the test by mapping significance against squared error for each rule in each domain. We also do a scatter diagram for the combined results of three KRK domains. That this diagram has the outline of the sort of curve we would expect for a single domain is evidence that noise is being accounted for properly within the three domains. Firstly we do all this for p^* (Fig. 2) and then for p_m (Fig. 3). In the combined KRK domain, the highest point has been omitted so as not to interfere with the KRK(10%) graph above.

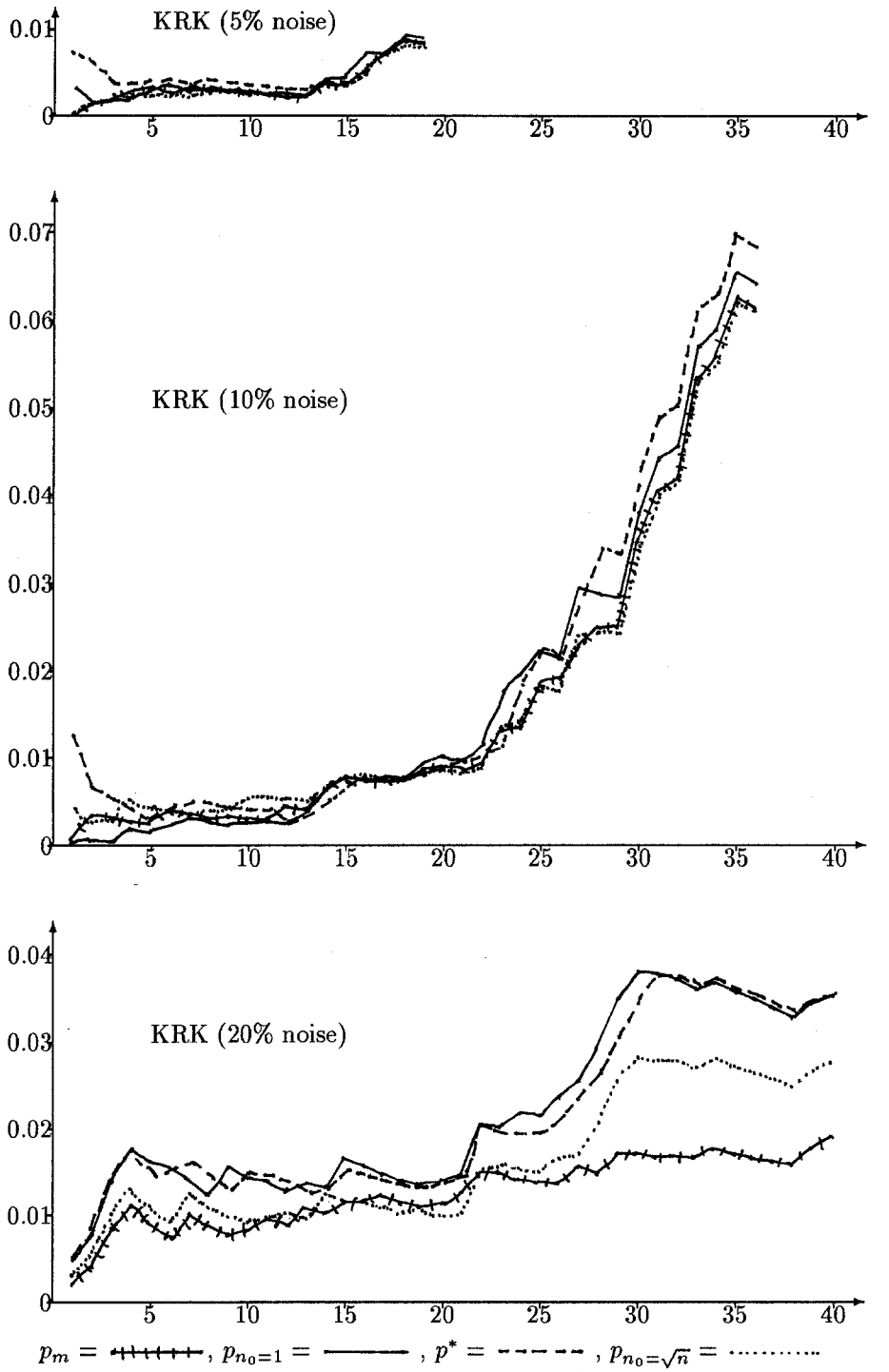

Fig. 1. Mean squared error (y-axis) against number of rules (x-axis)

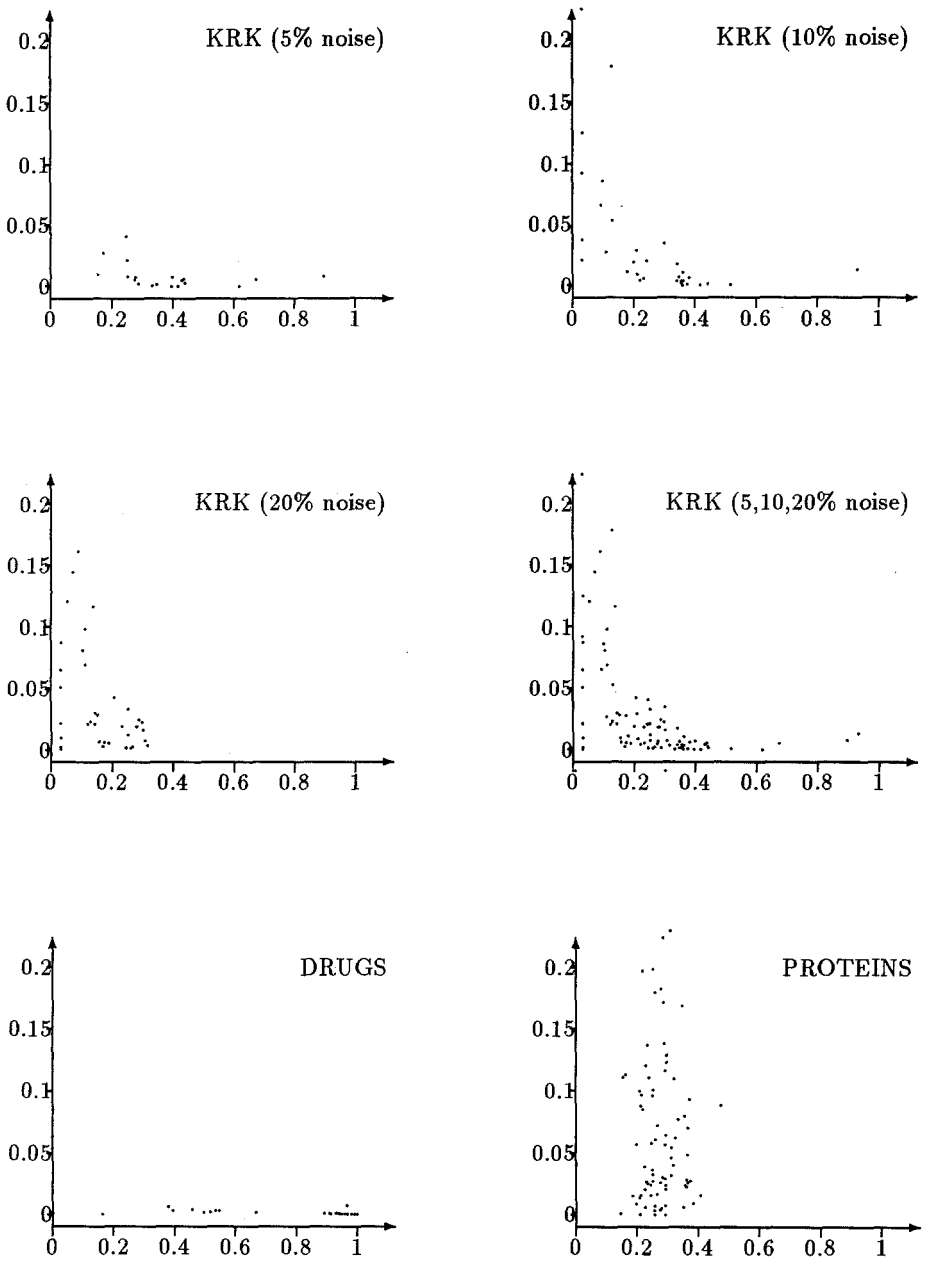

Fig. 2. Scatter diagrams of squared error (y-axis) against significance (x-axis) for p^*

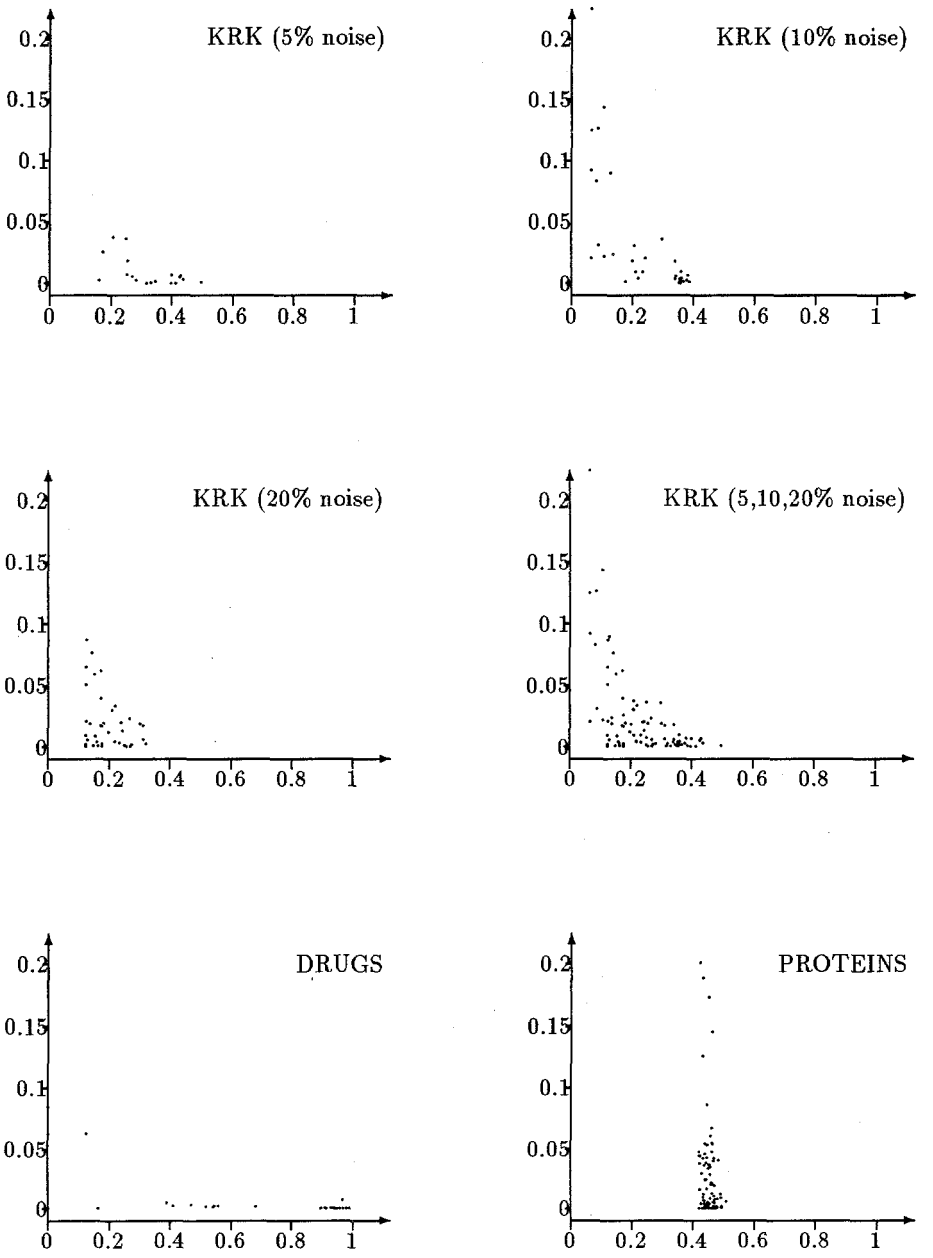

Fig. 3. Scatter diagrams of squared error (y-axis) against significance (x-axis) for p_m

In the Protein domain, training and testing sets are qualitatively different, so both measures give unimpressive results. In the Drugs domain, the data is highly concentrated around the x-axis, as expected.

Recall that the significance measure is $P(|\text{estimate} - p| < t)$ and we have $t = 0.025$. So significance is measuring $P(|\text{estimate}-p| < 0.025) = P((\text{estimate}-p)^2 < 6.25 \times 10^{-4})$. In Table 5 we give the number of points above and below 6.25×10^{-4} for several significance value intervals for the combined domain KRK(5%, 10%, 20%). We also give, in brackets, the values expected for that interval, to the nearest integer. For example, if there are 18 points with significance in the interval $[0.3, 0.4)$, we expect about $\frac{0.3+0.4}{2} \times 18 = 6.3$ points to be below 6.25×10^{-4} and 11.7 above.

Table 5. Number (expected number) of squared errors above and below 6.25×10^{-4}

Significance		0–0.1	0.1–0.2	0.2–0.3	0.3–0.4	0.4–0.5	0.5–0.6	0.6–0.7	0.7–0.8
p^*	$> 6.25 \times 10^{-4}$	21 (22)	19 (16)	24 (18)	15 (12)	4 (3)	0 (0)	1 (1)	1 (0)
	$\leq 6.25 \times 10^{-4}$	2 (1)	0 (3)	0 (6)	3 (6)	2 (3)	1 (1)	1 (1)	0 (1)
$p_{n_0=1}$	$> 6.25 \times 10^{-4}$	15 (15)	18 (16)	25 (19)	13 (10)	4 (4)	1 (1)	1 (0)	0 (0)
	$\leq 6.25 \times 10^{-4}$	1 (1)	1 (3)	0 (6)	2 (5)	4 (4)	1 (1)	0 (1)	0 (0)
$p_{n_0=\sqrt{n}}$	$> 6.25 \times 10^{-4}$	31 (31)	37 (35)	23 (17)	17 (14)	6 (5)	0 (0)	0 (0)	0 (0)
	$\leq 6.25 \times 10^{-4}$	2 (2)	4 (6)	0 (6)	4 (7)	3 (4)	0 (0)	0 (0)	0 (0)
p_m	$> 6.25 \times 10^{-4}$	11 (10)	29 (27)	23 (19)	14 (13)	5 (4)	0 (0)	0 (0)	0 (0)
	$\leq 6.25 \times 10^{-4}$	0 (1)	3 (5)	2 (6)	6 (7)	2 (3)	0 (0)	0 (0)	0 (0)

For all four estimators there is some correspondence between actual and expected values, but it appears that our significance measure is overestimating actual significance, since the expected number of points below 6.25×10^{-4} is nearly always higher than the actual number. m-estimation is the most successful estimator with respect to significance, presumably since it has the advantage of being empirically tunable. It is also notable that the expected number of good estimates was close to the actual number only when significance was very low.

9 Conclusions

The main problem with the above is that much more empirical testing of the various estimates and the significance measure needs to be carried out. However, it is already clear that Bayesian and pseudo-Bayesian estimation are superior to the standard maximum likelihood estimator ($\hat{p}$), especially in the presence of noise. This is because Bayesian estimation can use our prior estimate, $P(C)$, for $P(C|A)$. The success of estimation in the noisy KRK domains show that the proposed method of accounting for noise is working. Also, the results show the given measure of significance behaving roughly as it ought.

A Two Meanings of Significance

My intention here is to disentangle two related but different meanings of the term *significant* which appear to have become conflated in the literature.

Definition 1 *A rule is* significant$_1$ *if there is a high probability that its accuracy on testing data will be close to its training accuracy.*

Definition 2 *A rule is* significant$_2$ *if the distribution over classes of examples covered by the rule is appreciably different to the distribution over classes of examples in the domain as a whole.*

Estimates of a rule's probability are reliable when training accuracy is likely to be close to testing accuracy, so we have been measuring significance$_1$ in this paper. Significance$_2$ can be measured, for example, by likelihood ratio

$$LikelihoodRatio = 2n\left(r/n\log\left(\frac{r/n}{P(C)}\right) + (1-r/n)\log\left(\frac{1-r/n}{1-P(C)}\right)\right) \quad (14)$$

which is used in [6]. It is also used in [18] to measure significance$_1$. Since it is not designed to measure significance$_1$, it is not surprising that it gives unimpressive results there.

Suppose our domain is the set of people who attended ECAI-92. Consider the three rules:

$$H_A = \text{native\_english\_speaker} \rightarrow \neg\text{multilingual}$$
$$H_B = \neg\text{native\_english\_speaker} \rightarrow \text{multilingual}$$
$$H_C = \text{brown\_eyes} \rightarrow \text{multilingual}$$

Let us assume that in a randomly chosen set of training examples, H_A and H_B get training accuracy 95% and H_C gets 90%—fairly realistic figures! Assume that the number of training examples is reasonably large; so all rules have reasonably good cover. This means that all will be significant$_1$. This agrees with what we know about the domain—we would expect all rules to be similarly accurate on any possible testing set. H_A is significant$_1$ because most native English speakers, at ECAI-92 as elsewhere, are monoglots. H_C is significant$_1$, because most brown-eyed people *at ECAI-92* are multilingual. However it is clear that although there is a correlation between brown eyes and multilingualism in this domain, it is of a rather uninteresting nature. H_C is significant$_1$ merely because it covers a reasonably large number of individuals in a domain where multilingualism is common. H_B, on the other hand, is significant$_1$ for rather more 'genuine' reasons and will be also somewhat *more* significant$_1$ than H_C, due to its higher training accuracy.

Turning to significance$_2$, we find that H_A is highly significant$_2$, H_B is moderately significant$_2$ and H_C is insignificant$_2$. Native English speakers have considerably poorer linguistic abilities than most other subsets of the domain, so H_A is amongst the best rules for predicting $\neg$multilingual. H_B is similarly a good rule, but $P(\text{multilingual} \mid \neg\text{native\_english\_speaker})$ is only slightly higher than $P(\text{multilingual} \mid \top)$, the probability of the default rule $\top \rightarrow$ multilingual,[7] which

[7] which is maximally insignificant$_2$, but highly significant$_1$

decreases its significance$_2$. As for H_C, P(multilingual | brown_eyes) will be very close to P(multilingual | $\top$), so H_C will be insignificant$_2$.

The above argues that the two significances are distinct, but for an important class of rules, high significance$_1$ and high significance$_2$ coincide. If a rule $C \leftarrow A$ has high training accuracy and high training cover ($r/n \approx 1$, n is big), we have

1. Estimates of the probability of the rule will be high.
2. Significance$_1$ will be high.
3. Significance$_2$ will be high, *as long as the relevant default rule ($C \leftarrow \top$) has not also equally high training accuracy.*

This indicates that if we are looking for rules with 'a genuine correlation between attribute values and classes' as Clark and Niblett are in [6], we should focus on rules such as above. The question remains: which is the 'best' measure of 'genuine correlation', the probability estimate, significance$_1$ or significance$_2$?

Significance$_2$ (likelihood ratio) certainly can be used to find 'genuine correlation' as demonstrated by its successful employment in [6]. However, I feel that a *combination* of probability estimate and significance$_1$ measure has much to recommend it. For a rule $C \leftarrow A$, we can see the probability estimate as a 'best guess' at the extent of correlation between C and A. However, there is always the problem that even our best guess may be unreliable, so we can qualify the probability estimate with a significance$_1$ measure. We are only confident in those rules which have both high probability estimate and high significance$_1$.

Acknowledgements This work was funded by UK SERC grant GR/G 29854 for the Rule-Based Systems Project. Special thanks are due to Ashwin Srinivasan, who provided all of the necessary data, and to an anonymous reviewer for some helpful comments. Thanks are also due to Anthony Hunter, Donald Gillies, Stephen Muggleton and Dov Gabbay.

References

1. Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988.
2. Yvonne M. M. Bishop, Stephen E. Fienberg, and Paul W. Holland. *Discrete Multivariate Analysis: Theory and Practice.* MIT Press, Cambridge, Mass., 1975.
3. George E. P. Box and George C. Tiao. *Bayesian Inference in Statistical Analysis.* Addison-Wesley, Reading, Mass., 1973.
4. Bojan Cestnik. Estimating probabilities: A crucial task in machine learning. In Aiello, editor, *ECAI-90*, pages 147–149. Pitman, 1990.
5. Bojan Cestnik and Ivan Bratko. On estimating probabilities in tree pruning. In Yves Kodratoff, editor, *Machine Learning—EWSL-91*, pages 138–150. Lecture Notes in Artificial Intelligence 482, Springer, 1991.
6. Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
7. Luc de Raedt. *Interactive Theory Revision: An Inductive Logic Programming Approach.* Academic Press, London, 1992.

8. Sašo Džeroski and Ivan Bratko. Using the m-estimate in inductive logic programming. In *Logical Approaches to Machine Learning*, August 1992. ECAI-92 Workshop Notes.
9. Sašo Džeroski, Bojan Cestnik, and Igor Petrovski. The use of Bayesian probability estimates in rule induction. Turing Institute Research Memorandum TIRM-92-051, The Turing Institute, Glasgow, 1992.
10. Stephen E. Fienberg and Paul W. Holland. On the choice of flattening constant for estimating multinomial probabilities. *Journal of Multivariate Analysis*, 2:127–134, 1972.
11. Marek Fisz. *Probability Theory and Mathematical Statistics.* John Wiley, New York, third edition, 1963.
12. Jean Dickinson Gibbons. *Nonparametric Statistical Inference.* McGraw-Hill, New York, 1971.
13. I. J. Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Techniques.* M. I. T. Press, Cambridge, Mass, 1965.
14. David Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.
15. Colin Howson and Peter Urbach. *Scientific Reasoning: The Bayesian Approach.* Open Court, La Salle, Illinois, 1989.
16. Nada Lavrač, Bojan Cestnik, and Sašo Džeroski. Search heuristics in empirical inductive logic programming. In *Logical Approaches to Machine Learning*, August 1992. ECAI-92 Workshop Notes.
17. Stephen Muggleton and Cao Feng. Efficient induction of logic programs. In *Proceedings of the First Conference on Algorithmic Learning Theory*, pages 473–491, Tokyo, 1990.
18. Stephen Muggleton, Ashwin Srinivasan, and Michael Bain. Compression, significance and accuracy. In *IML 92*, 1992.
19. J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
20. Stuart J. Russell and Benjamin N. Grosof. Declarative bias: An overview. In D. Paul Benjamin, editor, *Change of Representation and Inductive Bias*, pages 267–308. Kluwer, Boston, 1990.
21. Cullen Schaffer. When does overfitting decrease prediction accuracy in induced decision trees and rule sets? In Yves Kodratoff, editor, *Machine Learning—EWSL-91*, pages 192–205. Lecture Notes in Artificial Intelligence 482, Springer, 1991.
22. Ashwin Srinivasan, Stephen Muggleton, and Michael Bain. The justification of logical theories. In Stephen Muggleton, editor, *Machine Intelligence 13.* Oxford University Press, 1993. To appear.
23. Michael Sutherland, Paul W. Holland, and Stephen E. Fienberg. Combining Bayes and frequency approaches to estimate a multinomial parameter. In Stephen E. Fienberg and Arnold Zellner, editors, *Studies in Bayesian Econometrics and Statistics*, pages 275–307. North-Holland, Amsterdam, 1974. volume 2.
24. Paul E. Utgoff. *Machine Learning of Inductive Bias.* Kluwer, Boston Mass., 1986.
25. Robert L. Winkler. *Introduction to Bayesian Inference and Decision.* Holt, Rinehart and Winston, New York, 1972.
26. Robert L. Winkler. Information loss in noisy and dependent processes. In J. M. Bernardo, M. H. Groot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics*, volume 2, pages 559–570. North-Holland, 1985.
27. Robert L. Winkler and Leroy A. Franklin. Warner's randomized response model: A Bayesian approach. *Journal of the American Statistical Association*, 74(365):207–214, March 1979.