On the Chance Accuracies of Large Collections of Classifiers

Keywords: order statistics, extreme value, feature selection, multiple hypothesis testing

Mark Palatucci MARKMP@CMU.EDU
Andrew Carlson ACARLSON@CS.CMU.EDU
Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract

We provide a theoretical analysis of the chance accuracies of large collections of classifiers. We show that on problems with small numbers of examples, some classifier can perform well by random chance, and we derive a theorem to explicitly calculate this accuracy.

We use this theorem to provide a principled feature selection criterion for sparse, high-dimensional problems. We evaluate this method on microarray and fMRI datasets and show that it performs very close to the optimal accuracy obtained from an oracle. We also show that on the fMRI dataset this technique chooses relevant features successfully while another state-of-the-art method, the False Discovery Rate (FDR), completely fails at standard significance levels.

1. Introduction

There are many real world problems in which a large number of *experts* predict the outcome of a small number of events. For example, we may ask one hundred football fans to predict the outcome of twenty games, or we may ask fifty political pundits to predict the outcome of ten elections.

With only a small number of events to predict, there may be a reasonable chance that some expert may predict all the outcomes perfectly, even if the outcomes are chosen at random.

For example, suppose we ask a person to predict the outcome of five coin flips where the probability of obtaining heads is 0.5. Since the flips are independent, this person has a $(0.5)^5 = \frac{1}{32}$ chance of guessing the outcome of all flips correctly. Now, if we ask ten people to predict the outcome of the five flips, there is a much higher chance that *someone* will predict all outcomes perfectly. With thirty-two people, someone would (in expectation) guess correctly each time.

Suppose we repeated this experiment again but asked our participants to predict the outcome of thirty coin flips. In this case, the chance of obtaining a perfect prediction would be nearly 1 in 1 billion. Given any number of participants less than 1 billion, we would not expect any participant to perfectly predict all the outcomes. But some participant will predict a series of outcomes that is most similar to the true flips.

How accurate should we expect this participant's predictions to be?

We consider this question and its relevance to machine learning. In our setting, we consider *experts* that are not people, but rather classification algorithms that predict labels for a set of examples.

When a large number of classifiers predict labels for a small number of examples, some classifiers will predict the labels well purely by random chance. This may lead us to believe that a subset of the classifiers are actually good predictors, when in fact they may be just guessing randomly.

This effect is commonly seen in discriminative feature selection, where a feature is selected based on the accuracy of a classifier trained on that single feature and tested on a held-out set of validation examples. In modern high-dimensional machine learning applications such as fMRI or microarray analysis, there are typically thousands of features with less than one hundred examples. Classification tasks in such settings often have *sparse* solutions, meaning that only a small subset of the features are useful for predicting the correct class.

To determine which features are relevant, it would be

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

useful to know how well *some* classifier could perform if all classifiers just chose labels at random. We would like to know how this accuracy changes with both the number of features and number of examples. This paper poses and answers the following question:

Given M classifiers that each produce labels randomly for N examples, what is the highest accuracy that we would expect some classifier to achieve?

1.1. Related Work

Our work is closely related to the multiple-testing problem in the statistics community. In statistics, hypothesis tests are the standard way to test if some assertion is true with high probability. While a single test has a low probability of making an error, when multiple hypothesis tests are performed simultaneously, the probability of at least one of the tests making an error can be much higher. It is common to *correct* the tests by making them more conservative to compensate.

Two of the most popular methods for correcting multiple tests are the Bonferroni Correction and the False Discovery Rate (Benjamini & Hochberg, 1995). We can apply these methods to the problem of feature selection, but in practice they are often too conservative at standard significance levels (e.g. 5%). See Wong et al. (2002) and Frank and Witten (1998). With many high-dimensional classification problems they may simply state that no feature is significant. This is not particularly helpful when building a classifier.

We could lower the significance level so more features are considered relevant, but it is unclear what significance level to choose. Since different learning algorithms have different tolerances to noisy, irrelevant features, there is no single significance level that is appropriate for all learning algorithms.

This fact, along with the large number of available tests and correction methods, makes hypothesis testing a difficult task for non-experts.

In our work, we approach the problem of significance from a different angle. Using *order statistics*, we explicitly model small chance events in a group setting. These techniques are relatively unknown in the machine learning literature although the *multiple comparison procedures* described in Jensen (2000) are similar in spirit.

We feel an order statistic approach is much more intuitive than hypothesis testing, and is well suited to problems in machine learning.

One such problem is discriminative feature selection. This feature selection technique is often called a *wrapper* method in contrast to more recent *embedded* methods like the $\mathcal{L}_1$ regularized Lasso (Tibshirani, 1996). While a full comparison of wrapper and embedded methods is beyond the scope of this paper, we believe that wrapped methods will continue to play a role in machine learning due to their simplicity and tractability. An excellent overview of the feature selection literature is available in Guyon (2003).

The work most similar to ours is by Li and Grosse (2003), which uses extreme value distribution theory to choose a significance threshold for selecting relevant features. While the general theme is similar, we do not use asymptotic results of extreme value theory, nor do we use simulation to compute moments of order statistics. By contrast, we focus on classification problems and show exact solutions that do not require any simulation.

2. Preliminaries

2.1. Order Statistics

We use *order statistics* extensively in this paper, thus we begin with a small introduction to define some basic concepts and notation. Consider M samples (i.i.d.) drawn from some distribution: $X_1, \ldots, X_M \sim F_X(x)$. If we order these samples from smallest to largest we obtain:

$$X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(M)}$$

and we use the notation $X_{(r)}$ to denote the r^{th} smallest sample which we call the r^{th} order statistic. $X_{(1)}$ and $X_{(M)}$ have special meaning which we call the *extreme values*:

$$\begin{aligned} X_{(1)} &= \min(X_1, X_2, \ldots, X_M) \\ X_{(M)} &= \max(X_1, X_2, \ldots, X_M) \end{aligned}$$

Each order statistic $X_{(r)}$ is also a random variable and can be described by a cumulative distribution function $F_{(r)}(x)$ or a density function $f_{(r)}(x)$. We will refer to an order statistic's *parent distribution*, which is the original distribution from which the M unordered samples were drawn. In our example this is $F_X(x)$.

We will use the notation $\mu_{r:M}$ to denote the mean of the r^{th} order statistic for M samples drawn from the parent distribution.

3. Expected Chance Accuracies

Using order statistics we can now answer the question we posed earlier:

Given M classifiers that each produce labels randomly for N examples, what is the highest accuracy that we would expect some classifier to achieve?

To answer this question, first consider a classifier that labels some collection of examples at random. If the classifier labels an example incorrectly with probability p_{err}, we can model the number of errors the classifier makes as a binomial random variable. Formally, let X be defined as the number of errors the classifier makes on some true labeling of N examples. Then:

$$X \sim \text{Binomial}(N, p_{err})$$

and the mean and variance of X are:

$$\begin{aligned} \mathbb{E}[X] &= N \cdot p_{err} \\ \mathbb{V}[X] &= N \cdot p_{err} \cdot (1 - p_{err}) \end{aligned}$$

Now, suppose instead we have M independent classifiers where each produces a set of N labels at random. Once again, the probability that each classifier makes a mistake on a single example is p_{err}. Let X_i be the number of errors made by the ith classifier. We then have:

$$X_1, X_2, \ldots, X_M \sim \text{Binomial}(N, p_{err})$$

One of these classifiers will have the minimal number of errors. Using our order statistic notation we have:

$$X_{(1)} = \min(X_1, X_2, \ldots, X_M)$$

and the expected minimum number of errors is:

$$\mu_{1:M} = \mathbb{E}[X_{(1)}]$$

If we knew the density function of $X_{(1)}$ for M samples from a Binomial(N, p_{err}) we could compute the mean $\mu_{1:M}$ directly:

$$\mu_{1:M} = \sum_{x=0}^{\infty} x f_{(1)}(x)$$

If the parent distribution were a continuous variable, obtaining $f_{(1)}$ would not be difficult and many references show simple methods to compute the density for any order statistic of a continuous distribution (Casella & Berger, 2002). Since our parent distribution is the discrete binomial, computing $f_{(1)}$ and more importantly $\mu_{1:M}$ is more difficult.

We could resort to simulation to find the mean, but this can be quite time consuming for large collections of variables. We will show later, however, that an exact solution does exist.

3.1. The Multiplicity Gap

For any problem with M classifiers and N examples there is a risk that some classifier will perform well by random chance. *What is a good measure of this risk?*

As we showed earlier, $\mathbb{E}[X_{(1)}]$ is the minimum number of errors that we should expect *some* classifier to make. We also know that $\mathbb{E}[X]$ is the expected number of errors an *individual* classifier will make.

Thus, one natural measure of this risk is the difference between these two values. We define the *multiplicity gap* $\mathcal{G}_{M,N}$ for M classifiers and N examples as:

$$\mathcal{G}_{M,N} = \mathbb{E}[X] - \mathbb{E}[X_{(1)}]$$

Reducing the number of examples N or increasing the number of classifiers M *increases* the risk.

4. Derivation

Theorem 4.1. *Highest Chance Accuracy*
Consider a classification problem with M classifiers and N examples. If the probability that a classifier makes a mistake on a single example is p_{err}, the highest expected accuracy $\mathcal{A}_H$ of any classifier is given by:

$$\mathbb{E}[\mathcal{A}_H] = 1 - \frac{1}{N} \sum_{i=0}^{N-1} I_{p_{err}}(i+1, N-i)^M \quad (1)$$

where $I_p(a, b)$ is the incomplete beta function[1]:

$$I_p(a, b) = \frac{1}{\beta(a, b)} \int_0^p t^{a-1}(1-t)^{b-1} dt$$

Proof. Let X_i, $(1 \leq i \leq M)$ be the total number of errors classifier i makes on some true labeling of N examples. If the probability that a classifier makes a mistake on a single example is p_{err}, then:

$$X_1, X_2, \ldots, X_M \sim \text{Binomial}(N, p_{err})$$

Therefore, the expected minimum number of errors is:

$$\mu_{1:M} = \mathbb{E}[X_{(1)}]$$

To compute the value of $\mu_{1:M}$ we utilize a useful result from Feller (1957) that relates the mean of a discrete random variable to its distribution function:

$$\mu_X = \sum_{i=0}^{\infty} [1 - F_X(i)]$$

[1] Some texts refer to this form as the regularized incomplete beta function.

therefore

$$\mu_{1:M} = \sum_{i=0}^{\infty}[1 - F_{(1)}(i)]$$

A result from David and Nagaraja (2003) shows that is equivalent to:

$$\mu_{1:M} = \sum_{i=0}^{\infty}[1 - F_X(i)]^M \quad (2)$$

Now, for a Binomial(N, p_{err}), $F_X(i) = 1$ when $i \geq N$. Therefore, the upper limit of the sum becomes $N-1$:

$$\mu_{1:M} = \sum_{i=0}^{N-1}[1 - F_X(i)]^M$$

Note that the incomplete beta function $I_p(a,b)$ has an expansion that looks similar to the distribution function of a binomial:

$$I_p(a,b) = \sum_{j=a}^{a+b-1} \frac{(a+b-1)!}{j!(a+b-1-j)!} p^j (1-p)^{a+b-1-j}$$

Using this expansion and a few algebraic manipulations we can express the tail of the distribution function in terms of the incomplete beta function[2]:

$$\begin{aligned} 1 - F_X(i) &= \mathbb{P}(X \geq i+1) \\ &= \sum_{j=i+1}^{N} \frac{N!}{j!(N-j)!} (p_{err})^j (1-p_{err})^{N-j} \\ &= I_{p_{err}}(i+1, N-i) \end{aligned}$$

Substituting this form into (2) we have:

$$\mu_{1:M} = \sum_{i=0}^{N-1} I_{p_{err}}(i+1, N-i)^M$$

To put our answer in terms of accuracy rather than errors we rearrange:

$$\begin{aligned} \frac{1}{N}(N - \mu_{1:M}) &= 1 - \frac{1}{N}\mu_{1:M} \\ &= 1 - \frac{1}{N}\sum_{i=0}^{N-1} I_{p_{err}}(i+1, N-i)^M \end{aligned}$$

Note that this theorem depends on the number of classes only through p_{err}. It does not require any modification to adapt to many classes. □

[2]We feel it is numerically advantageous to use the incomplete beta function rather than computing the binomial CDF directly. Many numerical computing environments have fast implementations of the incomplete beta function $I_p(a,b)$. For example, the `betainc(p,a,b)` command in MATLAB can implement Equation 1 in one line of code.

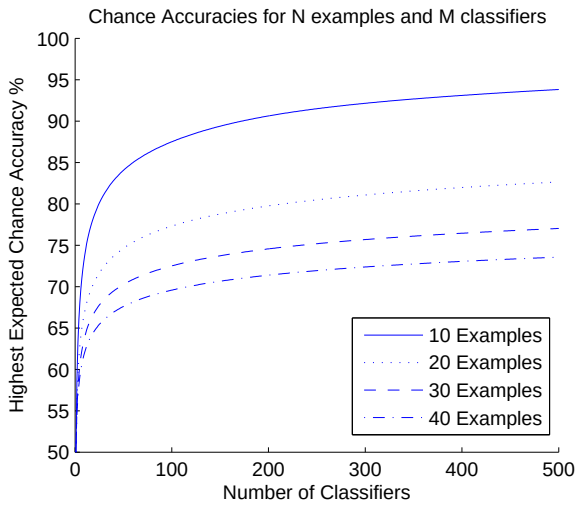

Figure 1. The highest expected chance accuracy as a function of the number of examples and classifiers. Each line represents a different number of examples. The x-axis is the number of classifiers and the y-axis is the accuracy.

Example 4.1 Predicting NFL games
Consider an office football pool with 200 participants betting on the outcome of 20 games. If each participant selects the outcome of a game according to a fair coin flip, how well would we expect the "winner" to perform?

To answer this question, we apply Equation 1 where $M = 200$, $N = 20$, and $p_{err} = 0.5$. In this case, the highest expected accuracy of some participant is 80%.

Although the chance probability of obtaining a *perfect* labeling is extremely small, in this case only $1/2^{20} = 1/1,048,576$, the chance of obtaining a *very good* labeling is much higher. Exactly 1,048,576 participants would be needed for us to expect one to obtain a perfect labeling. Yet, with only 200 participants, the expected accuracy of the top performer is 80%.

This effect can be seen by plotting Equation 1 for a two class problem where $p_{err} = 0.5$ (see Figure 1). The graph shows the highest expected chance accuracy (y-axis) for a given number of classifiers (x-axis). Each line represents a different number of examples N. As we increase the number of examples, the *multiplicity gap* closes, and highest expected chance accuracy for some classifier approaches the expected chance accuracy for a single classifier.

With small numbers of examples and large numbers of classifiers, the chance of obtaining a very good labeling may be very high, even if the chance of obtaining a perfect labeling is very low.

5. Case Study: Discriminative Feature Selection in Sparse, High-Dimensional Problems

A simple and popular method for finding relevant features in a classification task is discriminative feature selection. This method evaluates how well individual features discriminate between different classes and selects features with high predictive accuracy.

For example, if we have M features in a classification task, we train M distinct classifiers, where each classifier is trained using a single feature. After training, we evaluate all the classifiers on a set of *validation examples* and select the top performing features according to some criterion. A final classifier is then trained using only these top performing features, and then evaluated on some set of test examples.

This method is popular because it is simple to implement and often performs well in practice. The main difficulty is: *What are appropriate criteria for selecting significant features?*

One approach is to run a *cross-validation* loop, testing different significance thresholds to find one that has high empirical performance. This loop is computationally expensive and also requires additional validation examples. To avoid these difficulties in practice, it is common to choose some arbitrary threshold, and hope that performance is sufficient for the classification task.

Besides being pedantically unsatisfying, choosing an arbitrary threshold in a high-dimensional problem with a small number of examples is very risky. For example, a simple threshold might choose all features that perform better than 80% accuracy. As we showed earlier, many features may exceed this seemingly high threshold purely by random chance.

In high-dimensional problems with small numbers of examples, the accuracy required for statistical significance is often much higher than intuition might suggest.

A more principled approach for determining significance is to use a hypothesis test. With a hypothesis test, one tries to disprove a certain assertion. For example, one might assume that a classifier performs with a true accuracy of 50%. This assumption is called the *null hypothesis*. The goal then is to reject the null hypothesis if the evidence (e.g. the discriminative accuracy) is sufficiently strong.

Hypothesis testing has a vast literature in the statistics community. A good introduction can be found in Wasserman (2005). The Wald, "t", binomial, permutation, and χ^2 tests are just a few of the possible testing methods available. It is difficult, however, for a non-expert to know when to apply a particular test. To complicate matters, adjustments must be made when multiple tests are considered simultaneously. This is known in the statistics community as the *multiple testing problem*. Several methods such as the Bonferroni correction, family-wise error rate, and the false discovery rate (FDR) are used to compensate for multiple tests (Benjamini & Hochberg, 1995).

For the problem of discriminative feature selection, the use of a binomial test along with a false discovery rate adjustment is an appropriate choice. As we mentioned earlier, however, hypothesis tests require the choice of a significance level α. As is common in the scientific literature, the level $\alpha = 0.05$ is typically considered statistically significant.

For the purpose of feature selection, however, an appropriate choice of α is highly dependent on the classification algorithm used. Some classifiers are more tolerant to irrelevant features than others. Thus, there is no single α value appropriate for all classifiers. We could use a cross-validation loop to search for an appropriate α, but then we could have avoided the hypothesis test altogether and searched empirically for an appropriate threshold.

5.1. The Multiplicity Gap Midpoint (MGM) Method

Earlier in Equation 1 we derived the highest expected chance accuracy of some classifier assuming all classifiers choose their labels according to random chance. In some sense, this accuracy is a *natural significance threshold*, since we would not expect any classifer to perform better than this threshold by random chance.

While this may seem like an intuitive threshold for feature selection, in practice the threshold is overly conservative for several reasons. First, this threshold assumes all features are independent. This rarely holds in practice, and in many high-dimensional datasets it is very common to see strong correlations between features.

Further, the threshold assumes that all features are irrelevant and produce labels at random. In practice, some subset of the features will actually be significant, thereby lowering the effective number of random features. There is also no guarantee that errors for a feature can be modeled as a binomial random variable.

These violations of independence and irrelevance effectively lower the highest expected chance accuracy

(and increase the expected minimum number of errors). While this threshold may be overly conservative, it effectively serves as an upper bound on the highest expected chance accuracy.

At the other extreme, we might consider any feature significant that performs better than the expected chance accuracy of a *single* feature. As we showed before, this will clearly allow many irrelevant features to be considered significant. Note that these two extremes are the endpoints of the multiplicity gap that we defined earlier. If we model the number of errors made by a classifier as a binomial random variable, and we have M classifiers and N examples then the multiplicity gap $\mathcal{G}_{M,N} = \mathbb{E}[X] - \mathbb{E}[X_{(1)}]$

We conjecture that the optimal threshold should fall within the multiplicity gap.

In practice, we can choose any threshold between these two extremes. If we believe that our classifier is sensitive to irrelevant features, we should choose a threshold closer to $\mathbb{E}[X_{(1)}]$. Similarly, if our classifier is robust to irrelevant features, we should choose a threshold closer to $\mathbb{E}[X]$.

Without any knowledge of the particular classifier it is impossible to know what the optimal threshold should be. Therefore, as a simple heuristic we propose the *multiplicity gap midpoint* method, which chooses the midpoint of the extremes of the multiplicity gap. This yields a threshold τ_{MGM} on the maximum number of errors a classifier could make and still be considered significant:

$$\tau_{MGM} = \frac{\left(\mathbb{E}[X] + \mathbb{E}[X_{(1)}]\right)}{2}$$

where $\mathbb{E}[X_{(1)}]$ is computed as in Equation 1:

$$\mathbb{E}[X_{(1)}] \quad = \quad \mu_{1:M} = \sum_{i=0}^{N-1} I_{p_{err}}(i+1, N-i)^M$$

and $\mathbb{E}[X]$ is the number of examples N multiplied by the probability p_{err} that a classifier makes an error on a particular example: $\mathbb{E}[X] = N \cdot p_{err}$

To use this threshold, we perform a discriminative feature selection and select all features that make less than τ_{MGM} errors on a validation set with N examples.

5.2. Experimental Methodology

We perform discriminative feature selection experiments on two high-dimensional classification tasks that have few relevant features and limited training data:

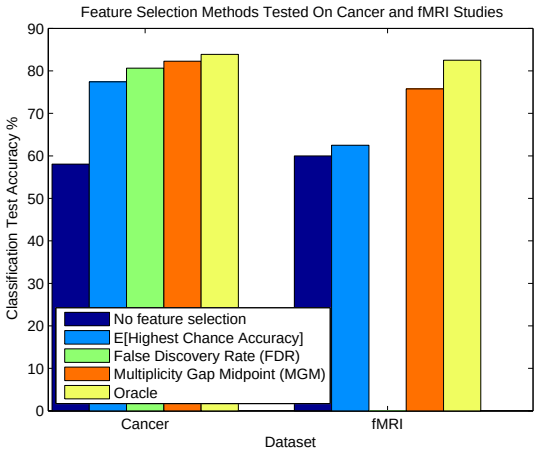

Figure 2. Accuracies for different feature selection methods for two classification tasks: Cancer (left) fMRI (right). The False Discovery Rate (FDR) method selected no features in the fMRI task.

Task 1: Cognitive state classification using functional magnetic resonance imaging (fMRI) In this task, we are given a time series of neural activity from thirteen human subjects. Each feature is the neuro-activation of a particular region of the brain at a given time. The goal is to distinguish between two cognitive states: reading a sentence, and viewing a picture (Mitchell et al., 2004). Each subject has ≈80,000 features and 40 examples.

Task 2: Colon cancer patient classification using microarray gene expression levels (Cancer) In this task, the goal is to predict whether a patient is diagnosed with colon cancer. The data are microarray gene expression levels from tissue samples (Alon et al., 1999). There are 2,000 features and 62 examples.

Testing Method In each experiment, we use a Gaussian Naive Bayes classifier and perform a leave-one-out-cross-validation. On each round, we leave out one example, and split the remaining examples into equal training and validation sets. We train using the first set, and measure classification accuracy on the validation set. We select the best performing features according to a specific criterion. After selecting features, we retrain by combining the validation and training sets. We then test the left out example. We repeat the process for each example.

We tested five different feature selection criteria:

1. **No feature selection** Uses all features.
2. **Highest Expected Chance Accuracy** Selects features that make fewer than $\mathbb{E}[X_{(1)}]$ mistakes.

3. **Binomial Hypothesis Test with False Discovery Rate correction** We select a feature if we reject the hypothesis that a classifier's true accuracy, trained on that feature, is 50%.[3] We use an $\alpha = 5\%$ level in the tests.
4. **Multiplicity Gap Midpoint (MGM) method** The method proposed in Section 5.1.
5. **Oracle Threshold** This is the threshold that would have led to the optimal testing accuracy.

5.3. Results and Discussion

In Figure 2, we see the classification results of five discriminative feature selection methods for both the colon cancer and fMRI datasets (for the fMRI dataset, we averaged the results of the 13 subjects together).

In both datasets, the threshold $\mathbb{E}[X_{(1)}]$ yields an improvement over no feature selection. But the assumptions made in calculating that threshold, namely that all features are independent and irrelevant, result in a very conservative threshold which admits few features.

The multiplicity gap midpoint (MGM) method relaxes these assumptions and performs significantly better. This method comes closest to the accuracy that could have been achieved had an oracle told us the optimal threshold to use[4].

As a state-of-the-art baseline, we tried a binomial hypothesis test with a false discovery rate correction. As is common in the statistical and scientific literature, we chose a significance level $\alpha = 0.05$. This method completely failed to select any features for the fMRI task, indicating that it is overly conservative for very high-dimensional problems. The method performed fairly well on the colon cancer dataset, but did so after selecting fewer than ten features.

It is worth noting that we could tune the α value of the false discovery rate test to admit more features and help performance. But the goal of the midpoint heuristic is to avoid this tuning (in fact, if we were to do tuning, it would make more sense to just tune the threshold for selecting features directly). Thus we feel the midpoint method provides a more appropriate *default threshold* than a specific value of α would in a classical test.

We chose the Gaussian Naive Bayes classifier because it is extremely fast to train and test making it very appropriate for use in a *wrapped feature selector*. This classifier is also robust to noise but is not entirely immune to overfitting. We found that adding additional features increased performance up to a point, but eventually noisy features overwhelmed the classifier, and performance degraded.

Figure 3 shows this effect for three fMRI subjects and the colon cancer dataset. The curves shows test accuracies at various feature selection thresholds. In each plot, the x-axis is the number of errors allowed, and the y-axis is the test accuracy of the resulting classifier. We mark the extremes of the multiplicity gap $\mathbb{E}[X_{(1)}]$ and $E[X]$ on each plot. On all thirteen subjects as well as the colon cancer dataset, the optimal (oracle chosen) threshold falls within this gap.

5.4. Future Work

The goal of this paper was to show how order statistics can be a useful tool for problems in machine learning. While our initial work focused on accuracy, we feel similar techniques can be applied to other measures such as information gain, entropy, and AUC.

Also, in our initial analysis we compute a significance threshold assuming that all features are independent. One natural extension of this work is to develop a method that adjusts for correlations between features.

6. Conclusion

We provided a theoretical analysis of the chance accuracy of large collections of classifiers. We showed that on problems with small numbers of examples and large numbers of features, we should expect some classifier to be highly accurate by random chance. We derived a theorem to directly calculate this accuracy.

We used this theorem to provide a principled feature selection criterion for sparse, high-dimensional problems. This criterion is theoretically well-motivated, simple to implement, and computationally inexpensive.

We demonstrated this method on microarray and fMRI datasets and showed that this method performs very close to the optimal oracle accuracy. We also showed that on the fMRI dataset this technique chooses relevant features while another state-of-the-art method, the False Discovery Rate (FDR), completely fails at standard significance levels.

Acknowledgments

We would like to offer our tremendous thanks to Haikady Nagaraja, Larry Wasserman, Tom Mitchell,

[3] This is appropriate since both datasets have nearly equal class priors.

[4] The oracle is determined by calculating the highest accuracy on a test set for every possible "number of errors" threshold on the validation set.

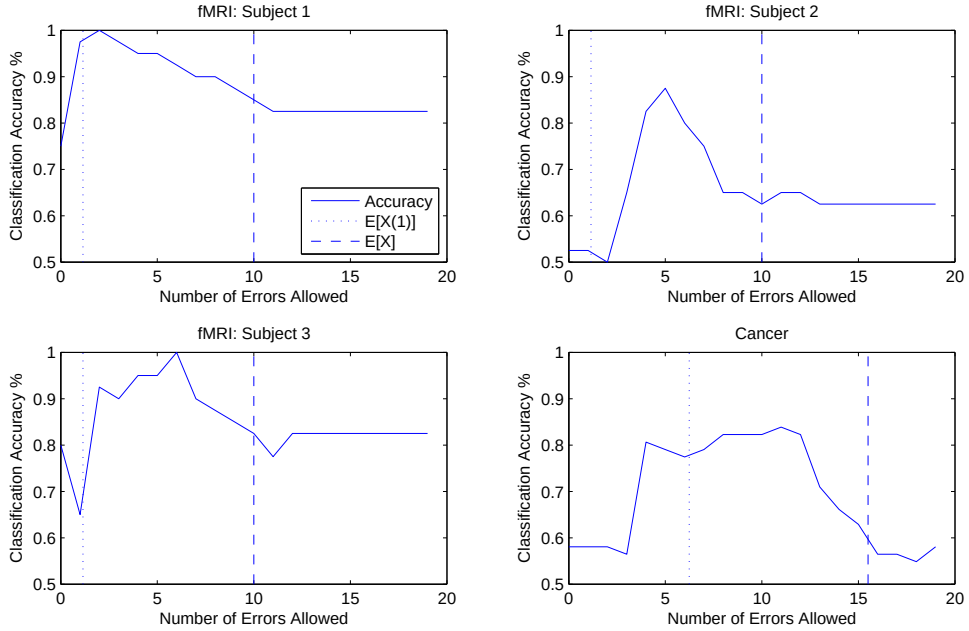

Figure 3. Test accuracies at various feature selection thresholds. In each plot, the x-axis is the number of errors allowed, and the y-axis is the test accuracy of the resulting classifier. We mark the extremes of the multiplicity gap $\mathbb{E}[X_{(1)}]$ and $E[X]$ with vertical lines on each plot.

and Geoff Gordon for their useful comments. We would also like to acknowledge the National Science Foundation, W.M Keck Foundation, and Yahoo! for their generous financial support.

References

Alon, U., et al. (1999). Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, *96*, 6745–6750.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, *57*, 289300.

Casella, G., & Berger, R. L. (2002). *Statistical inference*. Pacific Grove, CA: Duxbury.

David, H., & Nagaraja, H. (2003). *Order statistics*. Hoboken, NJ: Wiley.

Feller, W. (1957). *An introduction to probability theory and its applications*. New York, NY: Wiley.

Frank, E., & Witten, I. H. (1998). Using a permutation test for attribute selection in decision trees. *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 152–160). Morgan Kaufmann Publishers Inc.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Jensen, D., & Cohen, P. (2000). Multiple comparisons in induction algorithms. *Machine Learning*, *38*, 309–338.

Li, W., & Grosse, I. (2003). Gene selection criterion for discriminant microarray data analysis based on extreme value distributions. *RECOMB '03: Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology* (pp. 217–223). New York, NY, USA.

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., & Newman, S. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, *57*, 145–175.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, *58*, 267–288.

Wasserman, L. (2005). *All of statistics*. New York, NY: Springer.

Wong, W.-K., Moore, A., Cooper, G., & Wagner, M. (2002). Rule-based anomaly pattern detection for detecting disease outbreaks. *Eighteenth national conference on Artificial intelligence* (pp. 217–223). Edmonton, Alberta, Canada: American Association for Artificial Intelligence.