Measures of Prediction Accuracy and Ranking

Kyle Payne, Tangwei Li, Manzi Qin

March 1, 2016

Introduction

The following report consists of the Author's Recommendation for Measures of Prediction Quality for the TerraRef Project at the University of Illinois. In this preliminary report we will focus on two particular problems that have been addressed so far:

- How to determine the accuracy of a continuous prediction on a continous target value (e.g. phenotype).
- How to Score Predicted Rankings for some subset of lines (e.g. genotypes).

While we only address these problems in a relatively closed sense, the measures that we propose may be applicable to other settings as well. We will define the our measures, describe their respective numerical and statistical properties, make recommendations for using these measures in practice, and produce functions in both the Python and R languages for their implementation.

Measures

Continuous Phenotype Prediction

An example of the types of problems that fall under 'How to determine the accuracy of a continuous prediction on a continous target value (e.g. phenotype)' would be to demonstrate that predicted values are within 20 percent of ground truth values. Thus, we we need a measure that accounts for the difference between the predicted and ground truth (the true observed phenotype values), while also accounting for the *relative degree* by which the predictions are different from the ground truth values. One metric that appears in the literature (citation) is the Relative Mean Squared Prediction Error, or the RMSPE, which we define in equation (*)

Let $Y_1, ..., Y_n$ be a set of ground truth phenotype values, and let $\hat{Y}_1, ..., \hat{Y}_n$ be the set of corresponding predictions for the ground truth phenotype values, then the RMSPE is defined as.

$$RMSPE = \frac{\sqrt{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}}{\sqrt{\sum_{i=1}^{n} Y_i^2}}$$

The square of the numerator $\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ is a well-studied function within the machine learning and statistics community, known as the Mean Squared Prediction Error, or (MSPE). This is an easily computed, and numerically stable quantity that provides several desirable large sample properties. The denominator of $RMSPE$ can be viewed as the difference of the continuous ground-truth phenotype values from 0. Thus, $RMSPE$ acts a relative measure of the difference between the ground truth continuous phenotype values, expressed in units of the continuous phenotype. Using this measure, an experimenter could make a statment such as, "I demonstrated that the predicted values are within 0.2 of the ground truth values". This type of measure could be helpful for determining prediction accuracy for both Terminal Biomass and 3D Structural Models. Moreover, the $RMSPE$ could also be used to compare the prediction accuracy between competing models, such as comparing algorithm predictions v.s. the Lemnatec Software.

Expected Value of the RMSPE

While the RMSPE is a ratio of two random quantities, and thus may have a distribution that is difficult to derive analytically, it is possible to derive the expected value of RMPSE from a simple bivariate taylor expansion. That is, for any differentiable real-valued function

$$f(x,y) = f(\theta) + f_x^{'}(\theta)(x - \theta_x) + f_x^{'}(\theta)(y - \theta_y) + remainder$$

Then for $\theta = (E(X), E(Y))$, we get

$$f(X,Y) = f(\theta) + f_x^{'}(\theta)(X - E(X)) + f_x^{'}(\theta)(Y - E(Y)) + remainder.$$

Then for $f(x,y) = \frac{x}{y}$, a first-order taylor approximation of $E(\frac{X}{Y}) = \frac{E(X)}{E(Y)} + E(O(n^{-1}))$.

MSPE

Another important property of the RMSPE can be found when looking at the quadratic expansion of the numerator of the ratio, that is, the $MSPE$.

$$MSPE = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \sum_{i}(Y_i - E(Y_i|X_i = x_i) + E(Y_i|X_i = x_i) - \hat{Y}_i)^2 = \sum_{i=1}^{n}(Y_i - E(Y_i|X_i = x_i))^2 + \sum_{i=1}^{n}(E(Y_i|X_i = x_i) + error$$

Where the later error term is irreducbile. One conclusion from this derivation is that if one minimizes the $MSPE$ over a set of models

$M_0, M_1, ..., M_m$ then obviously one minimizes the $bias^2 + variance$ of the set of models. As we assume the restriction that all models $M_0, M_1, ..., M_m$ are trained and tested on the same sets of data, $\|Y\|$ is the same for all and only acts as a scaling factor.

Examples

If a prediction model produces a set of predictions $\hat{Y}_1, ..., \hat{Y}_n$, for a set of ground truth continuous phenotype values $Y_1, ..., Y_n$, then let's state that

$$RMSPE \leq 0.20 \Rightarrow \sqrt{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \leq 0.2\sqrt{\sum_{i=1}^{n} Y_i^2}$$

The equation above describes the case where the $RMSPE$ being less than or equal to 20 percent is equivalent to stating that the mean squared prediction error is bounded by 0.2 the size of the ground truth phenotype values. The quantity on the left-hand side of the inequality is an example of a commonly used measure of distance in mathematics, engineering, statistics, computer science, etc. known as the *Norm* (Weisstein, Eric W. "Norm." From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/Norm.html`).

Let $RMSPE_L$ be the relative mean squared prediction error of Lemnatec software predictions on the ground truth continuous phenotype values $Y_1, ..., Y_n$, which we will denote as $Y_1^*, ..., Y_n^*$. Let $RMSPE_M$ be the relative mean squared prediction error of some prediction model or algorithm. Then, making a determination such as 'algorithm predictions are no less accurate than values predicted via LemnaTec software', would require comparing $RMSPE_L$ and $RMSPE_M$, thus

$$RMSPE_M \leq RMSPE_L$$

Thus, if we compare two prediction models in $RMSPE$ on the same set of data, the comparision is equivalent to just comparing the average squared difference between the predictions and the ground truth values.

Penalized RMSPE

While relative mean squared prediction error is guaranteed to minimize the overall $bias^2 + variance$ over a set of prediction models fit to the same test set, it is not currently equipped to score the cases in which we wish to compare multiple models predictions that include interval estimations. We now introduce a measure that for fixed probability coverage $1 - \alpha$, can evaluate model predictions with interval predictions. We also will present an argument for how it may be improved based on assumptions on the underlying distribution of the data.

Let $C_n(x) \subset \mathbb{R}^1$ be an interval prediction s.t. for all x

$$C_n(x) = C((X_1, Y_1), ..., (X_n, Y_n), x)$$

is an interval prediction based on the training data, a set of pairs

$$(X_1, Y_1), ..., (X_n, Y_n)$$

with $X_k \in \mathbb{R}^d$ and $Y_k \in \mathbb{R}$ with $d < n$. Moreover, $C_n(x)$ is typically constructed such that

$$P(Y_{n+1} \in C_n(X_{n+1})) \geq 1 - \alpha$$

That is, the probability coverage of the prediction interval is at least $1 - \alpha$. Obviously, we want our prediction interval to be as small as possible in lebesgue measure while retaining at least $1 - \alpha$ probability converage of any observation in the test set, i.e. (X_{n+1}, Y_{n+1}). A general evaluation of model performance could be one in which we consider models which have the smallest overall bias, variance, and (if equipped with interval predictions), the smallest size of a prediction interval over a set of models where the probability coverage of the interval is $1 - \alpha$. A measure of efficiency of the interval estimate is the Lebesgue measure, which in any subset of $[a, b] \subset \mathbb{R}$ is just $b - a$. Thus, as $C_n(x)$ is just an interval pointwise, we can measure of the efficiency as defined in (citation) as $C_n^u(x) - C_n^l(x)$ where $C_n(x) = [C_n^l(x), C_n^u(x)]$.

To formalize the measure, let

$$\tau_r = \{(X_1, Y_1), ..., (X_n, Y_n)\}$$

be training data with $X_k \in \mathbb{R}^d$ and $Y_k \in \mathbb{R}$ with $d < n$.

and let

$$\tau_e = \{(X_n, Y_n), ..., (X_{n+k}, Y_{n+k})\}$$

be test data with $X_k \in \mathbb{R}^d$ and $Y_k \in \mathbb{R}$, that is, there are k many observations in τ_e. This notation is without loss of generality as the ordering in the set is entirely arbitrary. Let a model M be fit on τ_r, with point predictions $\hat{\mathbf{Y}}$, and prediction interval estimates $C_n(X_n), ..., C_n(X_{n+k})$. Then

$$PRMSPE_{\tau_e} = \frac{\|Y - \hat{Y}\|}{\|Y\|} + \frac{\sum_{j=1}^{k}(C_n^u(X_{n+j}) - C_n^l(X_{n+j}))}{\|Y\|}$$

A problem with this formulation is that due to the typical case in which the interval has much larger Lebesgue measure when compared to how far off the prediction is from the true value in the test set. The amount of the measure from the RMSPE is relative small compared to the effect of the size of the confidence inteval. A natural response is then to renalize the measure such that the two contributions to the size of the measure are roughly the same. But then, one must ask what kind of penalty should be used. An alternative measure factoring in the size of the prediction interval is defined by Gneiting, Raferty (2007) as follows:

$$S(C_n(x), y)_\alpha = (p_u - p_l) + \frac{2}{\alpha}(p_l - y)\mathbb{1}(y < p_l) + \frac{2}{\alpha}(y - p_u)\mathbb{1}(y > p_u),$$

Suppose your prediction interval has lower limit p_l and upper limit p_u, and the goal is to design a central prediction interval at level α (i.e. you allow for $\alpha/2$ percent of the observations to fall below p_l, and for $\alpha/2$ percent to fall above p_u).

The intuition behind the score formula is in the best of all worlds, we would want to have very short prediction intervals (p_u only slightly larger than p_l), which nevertheless cover the realization y, $p_l \leq y \leq p_u$. In this ideal world, the first summand of the score formula would be close to zero, whereas the second and third summands would be exactly zero.

Of course, in practice there is a tradeoff between producing short prediction intervals and actually covering the realization y. The score above strikes a balance between those two goals.

Characterization of Scoring Rules

S is a proper scoring rule in the sense of Gneiting, Raferty (2007). If we have a general probability space, then any probabilistic forecast or prediction is just a measurable probability function within the class of Probability measures $\forall p \in P$. If a Score is proper then

$$S(Q, Q) \leq S(P, Q)$$

with equality only if $Q = P$. This is know in Casella, Lehmann (1998, pg. 157) as risk unbiasedness. Thus we follow the framework of Gneiting, Raferty (2007) and Casella, Lehmann (1998, pg. 157) in constructing scores for interval estimates that are proper in the sense above.

Criteria for Scoring Prediction Models

Following the intuition for a score that weighs both the point prediction accuracy as well as the interval estimation width and coverage, we arrive at

$$PRMSPE_{\tau_e} = \frac{\|Y - \hat{Y}\|}{\|Y\|} + \frac{1}{k\|Y\|} \sum_{i=1}^{k} S(C_n(x), y_i)_\alpha$$

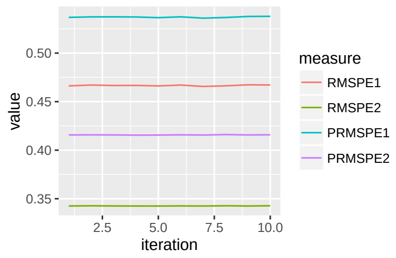
Practically, this score penalization results in a convenient and well-behaved score for measuring the strength of a prediction model while providing an even measure of the interval strength.

```
## Warning: package 'mgcv' was built under R
## version 3.2.4

## Loading required package: nlme

## Warning: package 'nlme' was built under R
## version 3.2.4

## This is mgcv 1.8-12. For overview type 'help("mgcv-package")'.
```



Another interesting aspect of the interval score penalty is that it may work in comparisons for predictions with different probability coverages.

Lemma 1.

If there are two model-based prediction intervals $C_n^{(0)}(x)_\alpha = [p_l^{(0)}, p_u^{(0)}]$, $C_n^{(1)}(x)_\alpha = [p_l^{(1)}, p_u^{(1)}]$, for models M_0 and M_1, then for a test sample (x_{n+1}, y_{n+1}), if $\epsilon = (p_u^{(1)} - p_l^{(1)}) - (p_u^{(0)} - p_l^{(0)})$ and

$$S(C_n^{(0)}(x_{n+1})_\alpha) \leq S(C_n^{(1)}(x_{n+1})_\alpha)$$

then

$$(p_l^{(0)} - y_{n+1})\mathbb{1}(y_{n+1} < p_l^{(0)}) - (p_l^{(1)} - y_{n+1})\mathbb{1}(y_{n+1} < p_l^{(1)}) + (y_{n+1} - p_u^{(0)})\mathbb{1}(y_{n+1} > p_u^{(0)}) - (y_{n+1} - p_u^{(1)})\mathbb{1}(y_{n+1} > p_u^{(1)}) \leq \frac{\alpha\epsilon}{2}$$

The Proof of Lemma 1 is given in the appendix.

One consequence of Lemma 1 is that if for any arbitrary observation in the test set (Y_{n+1}, x_{n+1}), if $S(C_n^{(0)}(x_{n+1})_\alpha) \leq S(C_n^{(1)}(x_{n+1})_\alpha)$ then either

$$(p_l^{(0)} - y_{n+1}) \leq \frac{\alpha\epsilon}{2} + (p_l^{(1)} - y_{n+1})\mathbb{1}(y_{n+1} < p_l^{(1)})$$

or

$$(p_l^{(0)} - y_{n+1}) \leq \frac{\alpha\epsilon}{2} + (y_{n+1} - p_u^{(1)})\mathbb{1}(y_{n+1} > p_u^{(1)})$$

or

$$(y_{n+1} - p_u^{(0)})\mathbb{1}(y_{n+1} > p_u^{(0)}) \leq \frac{\alpha\epsilon}{2} + (p_l^{(1)} - y_{n+1})\mathbb{1}(y_{n+1} < p_l^{(1)})$$

or

$$(y_{n+1} - p_u^{(0)})\mathbb{1}(y_{n+1} > p_u^{(0)}) \leq \frac{\alpha\epsilon}{2} + (y_{n+1} - p_u^{(1)})\mathbb{1}(y_{n+1} > p_u^{(1)})$$

or both intervals cover the target sample y_{n+1}. The meaning of these comparisons seem to indicate that if the prediction band does not cover the actual test realization for the prediction band with the lower (better) score, then the distance between the upper or lower limit of the prediction band and the true test realization is bounded by $\frac{\alpha\epsilon}{2}$ and the difference in the upper or lower prediction band and the true test realization for model M_1. Simply put, if the prediction band $C_n^{(0)}(x_{n+1})$ screws up, then it can't be by much.

One might question the importance of the probability coverage of prediction intervals if for any finite sample then prediction bands cover y_{n+1}. Of course such an interval would be a godsend. However, while we can evaluate the empirical coverage proportion in any test sample $\frac{1}{n}\sum_{i=1}^{n} 1(Y_i \in C_n^{(}x_{n+1}))$, we would hope to use the preidc-tion interval in a context in which we couldn't check the coverage empirically. Thus, we have to consider the probability coverage of the interval in determining optimal intervals. This is also complicated by the fact that one can create arbitrarily good coverage of Y_{n+1} by taking $\alpha \to 0$. Thus the interval score S_n balances size of the overall interval by the empirical coverage, and should work well for models that generalize well.

Performance

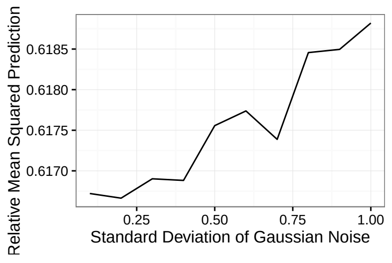
The Relative Mean Squared Prediction Error Performs well in situations in which there is additional noise in the continuous phenotype values $Y_1, ..., Y_n$. For the following example, let's assume that some prediction algorithm has been fitted to a set of training data. In this case, we chose a Random Forest Model to predict Stem Biomass using the plot identifier and the precipitation data on a sub-sample of simulated data. We trained the random forest model on a sub-sample of simulated data. Predictions were then made using a test sample of the data, with additional mean 0 gaussian random noise applied to the Stem Biomass data. The RMSPE measure increases like a polynomial with increasingly variable noise. However, the RMSPE remains relatively robust to deviations from the true Prediction Error, and only increases to very large values as the

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```



Example

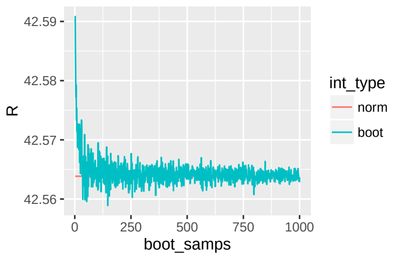
We can also elucidate the behavior of the Scoring Penalty by comparing a single prediction model with two sets of prediction intervals. In this section, we will compare the performance of both the R score and an alternative score, the $MASE$ when comparing a series of increasingly precise bootstrap prediction intervals to that of a normal approximation.

We estimated a Generalized Additive Model (citation) with a 0.95 prediction interval approximation of the type described in (Wasserman, 2012):

$$C_n(x) = (\hat{m}(x) - z_{\alpha/2}\sqrt{sig\hat{m}a^2 + s}, \hat{m}(x) + z_{\alpha/2}\sqrt{sig\hat{m}a^2 + s})$$

Thus, this prediction interval is symmetric about the point predicted value $\hat{m}(x)$, and likewise acheives nominal 0.95 coverage as the sample size becomes arbitrarily large.

```
## No id variables; using all as measure variables
```



Ranking

Another problem of interest is assigning a score to rank predictions, e.g. given a set of lines of sorghum, and we score the predicted rank orders from some prediction algorithm? The literature is replete with measures of rank ordering. We will focus here two measures of rank ordering correlation, Kendall's Tau and the Normalized Discounted Cumulative Gain.

Kendall's Tau

Kendall's Tau is a measure of the correlation between two sets of rankings. Let $Y_1, ..., Y_n$ be a set of observations and let $\hat{Y}_1, ..., \hat{Y}_n$ be a set of predicted observations. Then Any pair of observations $(Y_i, \hat{Y}_i)$ and $(Y_j, \hat{Y}_j)$, where $i \neq j$, are said to be concordant if the ranks for both elements agree: that is, if both $Y_i > Y_j$ and $\hat{Y}_i > \hat{Y}_j$ or if both $Y_i < Y_j$ and $\hat{Y}_i < \hat{Y}_j$. They are said to be discordant, if $Y_i > Y_j$ and $\hat{Y}_i < \hat{Y}_j$ and $Y_i < Y_j$ and $\hat{Y}_i < \hat{Y}_j$. If $Y_i = Y_j$ and $\hat{Y}_i = \hat{Y}_j$, the pair is neither concordant nor discordant.

Kendall's Tau is defined as:

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{1/2n(n-1)}$$

If both sets, the ground truth and predicted values are completely concordant in rank, then $\tau = 1$. Likewise, if both sets are in perfect disagreement, then $\tau = -1$. Kendall's Tau is analogous to a correlation coefficient in this manner. While Kendall's τ is easy to interprete, τ does not weight discordance differently depending on where in the rankings the discordance occurs. Therefore, one may be interested in a different ranking score, which we describe below. For examples of Kendall's Tau, please see the github repository.

Normalized Discounted Cumulative Gain

Normalized Discounted Cumulative Gain (NDCG) has gained prominence in the Learning to Rank literature across many fields as a robust and power answer to the problem of measuring the performance of ranking functions. First we start with a explation of the discounted cumulative gain, then we will introduce the normalize version of the measure.

$$DCG_{n,p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

where n is the total number of items to rank. rel_i is the relevancy of the predicted rank, which is a value that can take on varying interpretations depending on the context. For this work, we set $rel_i = n - |p_i - t_i|$, where p_i is the predicted rank for the i^{th} ranked value and t_i is the true rank for the i^{th} value. Obviously $t_i = i$.

$$DCG_p = \sum_{i=1}^{p} \frac{2^{n-|p_i-i|} - 1}{log_2(i+1)}$$

Examples

Let's say we wish to predict the ranks of the top 10 lines of sorhgum, based on the overall biomass. Then,

```r
library(knitr)
true_rank <- 1:10
pred_rank <- c(1, 2, 4, 3, 5:10)

ranks <- matrix(c(true_rank, pred_rank), byrow = T,
    nrow = 2)
row.names(ranks) <- c("Ground Truth", "Predicted")

kable(ranks)
```

Ground Truth	1	2	3	4	5	6	7	8	9	10
Predicted	1	2	4	3	5	6	7	8	9	10

In the table, we have the true ranks for the top 10 lines of sorghum out of 20 total lines, which is obviously just a sequence of the first 10 integers. At the 3rd and 4th positions, our prediction algorithm get's the predicted ranks wrong, and swaps 3 and 4. Therefore, the term in the summand of the discounted cumulative gain, that is

$$(2^{rel_3} - 1)/log_2(3+1) = (2^{20-|(3-4)|} - 1)/log_2(4) = 262144$$

$$(2^{rel_4} - 1)/log_2(3+1) = (2^{20-|(4-5)|} - 1)/log_2(5) = 225798.6$$

The mis-ranking at the 4th position had a slightly smaller effect on the scoring than that of the misprediction at the 4th position.

```r
library(knitr)
true_rank <- 1:10
pred_rank <- c(1, 2, 3, 4, 5, 6, 7, 10, 9, 8)

ranks <- matrix(c(true_rank, pred_rank), byrow = T,
    nrow = 2)
row.names(ranks) <- c("Ground Truth", "Predicted")

kable(ranks)
```

Ground Truth	1	2	3	4	5	6	7	8	9	10
Predicted	1	2	3	4	5	6	7	10	9	8

In this example, the prediction algorithm mixed up the bottom 3 in our list of the top 10 ranks. Let's look at the first term in the summands that correspond to those positions in the list:

$$(2^{rel_8} - 1)/log_2(9) = (2^{20-|(2)|} - 1)/log_2(9) = 82696.91$$

As the mis-ranking occurs much farther down in the list of the rankings, it contributes much less to the overall score. Thus, if the objective is the maximize the overall ranking score, then mis-rankings in positions far along in the list cost less than mis-rankings that occur in the front of the list.

A perfect prediction ranking would lead to a maximized DCG, for this example if would be

$$(2^{20} - 1)\sum_{k=1}^{10} \frac{1}{log(k+1)}$$

The Normalized Discounted Cumulative Gain is the ratio of a prediction algorithm's Discounted Cumulative Gain over the maximum possible Discouted Cumulative Gain.

$$NDCG = \frac{DCG_p}{max\{DCG_p\}}$$

One reasonable question to ask is why do define $rel_i = n - |p_i - i|$? instead of $rel_i = k - |p_i - i|$, where k is the length of the list of rankings compared. We chose this form as it allows for mis-rankings outside of the "top k many lines" list. E.g.

```r
library(knitr)
true_rank <- 1:10
pred_rank <- c(1, 2, 3, 4, 50, 6, 7, 8, 9, 10)

ranks <- matrix(c(true_rank, pred_rank), byrow = T,
    nrow = 2)
row.names(ranks) <- c("Ground Truth", "Predicted")

## kable(ranks)
```

We can still score the predicted ranking relative to the true ranking even if the $5th$ position of the list of predicted ranks has a value that is way outside of the top 10 values.