Robust Uncertainty Estimates for Unbalanced Data Sets

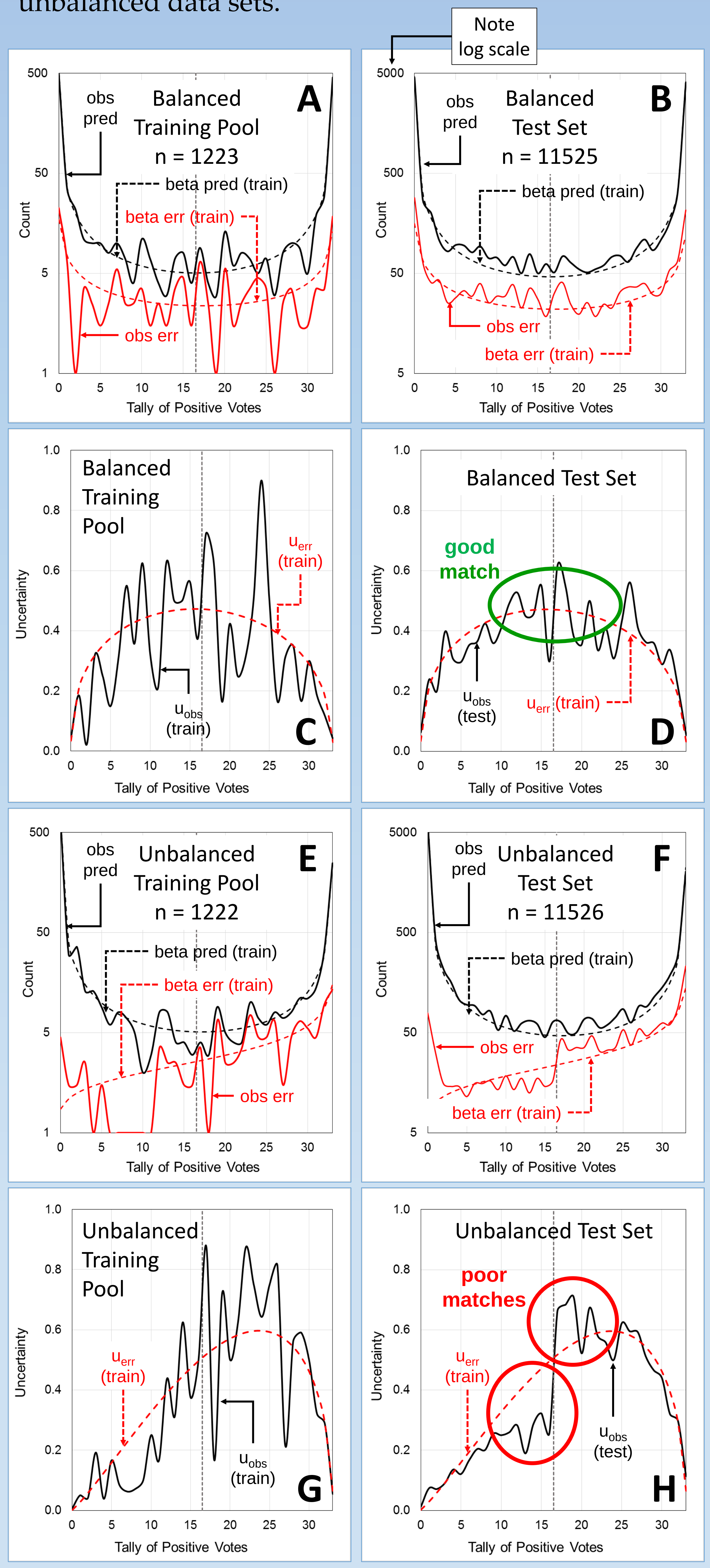
Robert D. Clark and Marvin Waldman

Simulations Plus, Inc., 42505 10th Street West, Lancaster CA 93534 USA

Summary

Mathematical models of quantitative structure-activity relationships (QSARs) play a key role in qualifying synthesis ideas, drug candidates and leads. In many cases (e.g., solubility), these are regression models but classification models – e.g., for mutagenicity or CYP inhibition – are also important. Overall sensitivity and specificity are commonly used as performance metrics for such models. Some predictions are more clear-cut than others, however, so it is often also important to obtain robust confidence estimates for individual predictions, especially in regulatory contexts.

We recently showed that predictions and errors from artificial neural network ensemble (ANNE) classification models follow beta binomial distributions with respect to the degree of consensus within the ensemble, and that those distributions yield a robust estimate of uncertainty for individual binary classification predictions [1]. The method works remarkably well when applied to balanced and moderately unbalanced data sets. Its performance can be suboptimal when applied to unbalanced data sets, however, where there are many more examples in one class or where examples from one class are much more informative than examples from the other class. Here we describe a variation of the method where the positives and negatives are fit to separate (split) beta binomial distributions. Doing so yields more accurate estimates of predictive uncertainty for most unbalanced data sets.



General Methods

All modeling work was done in the ADMET Modeler Module™ of ADMET Predictor™ v8.5. Descriptors are predominantly molecular attributes calculated from chemical structure.

Models are artificial neural networks ensembles (ANNEs), where each network has a single hidden layer and a single logistic output neuron. Weighting is by $1/n_{class}$ in the objective function, which maximizes a generalized version of Youden's index (J = sensitivity + specificity – 1).

Ensemble output is determined by tallying the number of "positive" votes across an ensemble. An option to use average network outputs rather than voting is also available but is not discussed here.

The 33 networks in each ensemble are each optimized against a different ~2:1 split of the training pool into train & verify subsets. They share inputs, however, and have the same number of hidden neurons. Moreover, they are all trained to do the same thing – classify compounds correctly – so their predictions agree most of the time. The networks are therefore not statistically independent, and the ensemble predictions follow *beta binomial distributions* as a result [1].

Results

Figure 1 (left) shows the results of fitting the error beta binomial to combined positive and negative errors from the training pool. **Figure 2** (right) shows the result of splitting predictions for positive and negative examples then building a split uncertainty from the distribution of positive examples whose vote tally falls below the decision threshold (false negatives) and the distribution of negatives that fall above the threshold (false positives).

A unified uncertainty profile tends to work well for balanced data sets (green circle in **1D**). For unbalanced data sets, on the other hand, the simpler approach dos not deal well with the transition between classes that occurs at the threshold (red circles in **1H)**.

A split uncertainty profile handles uncertainty well for an unbalanced data set (green circle in **2H**), but overestimates the uncertainty at the transition for the balanced case (red circle in **2D**).

Conclusion

Estimating uncertainty using beta binomial analysis of the distribution of combined positive and negative errors works well for balanced data sets but treating positive and negative error distributions separately may work better for unbalanced data sets. In either case, beta binomial uncertainty analysis of ANNE training pool data provides excellent estimates of prediction uncertainty when applied to large external test sets.

Figure 1 (left): Distributions of prediction ("pred") and error ("err") counts and of uncertainties ("*u*") as a function of the number of "positive" votes by the 33 networks in each ensemble. The beta binomials and uncertainties derived from them are also shown. Beta binomials ("beta") were fit only to predictions and errors *for the training pool*. Observed uncertainty profiles (u_{obs}) for the training pool and test sets equal the respective ratio of observed errors to observed predictions at each tally. Estimated uncertainties for the errors (u_{err}) equal the ratio of the error and prediction beta binomials evaluated at each tally *for the training pool.*

(A-D) Results for a balanced data set of 5535 compounds with logP ≥ 2 (positives) and 5990 compounds with logP < 2 (negatives). The model takes 50 input and has 1 hidden neuron.

(E-H) Results for an unbalanced data set of 3961 compounds with logP ≥ 3 (positives) and 8787 compounds with logP < 3 (negatives). The model takes 40 inputs and has 5 hidden neurons.

Green and red circles highlight good and poor matches.

Figure 2. (right). As in Figure 1 except that the full distribution of tallies for prediction made for positive ("pos") and negative ("neg") examples are also shown, not just the ones on the "wrong" side of the threshold. The uncertainty estimated from separated error beta binomials ("u_{split}") is shown as well. Other details are as for **Figure 1**.

Confidence Estimation

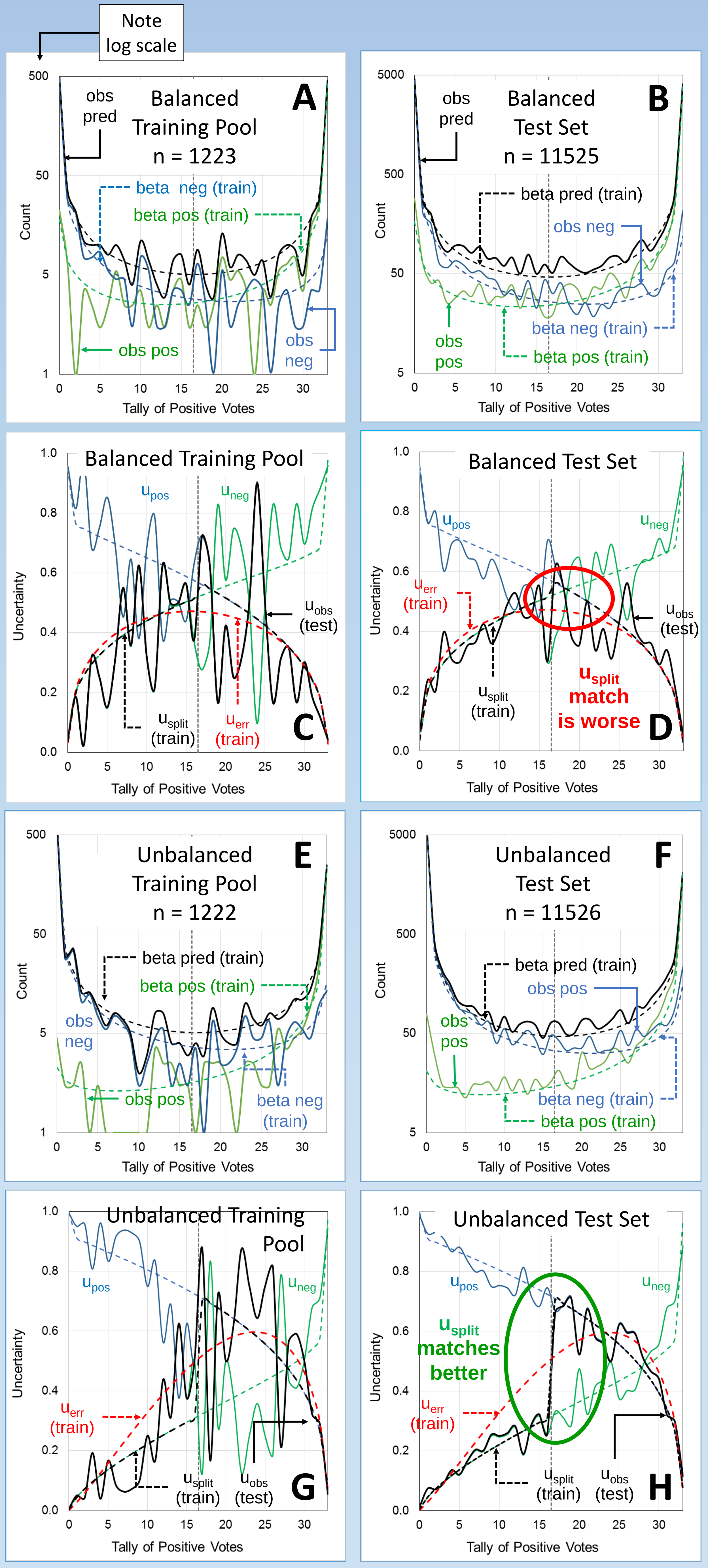
Fit beta binomials to the distributions of predictions.

1. Fit a beta binomial distribution g(k) to all training pool predictions as a function of the number k of networks casting "positive" votes
   - k = 0 to K; K = 33 by default in ADMET Modeler.
2. Fit a beta binomial distribution f(k) to all (positive and negative) training pool errors.
3. Fit a beta binomial $f_0(k)$ to the negative examples
4. Fit beta binomial $f_1(k)$ to the positive examples

Estimate uncertainties and predictive confidences from the ratio of the predicted frequencies at each tally.

5. Calculate the unified uncertainty u_{err} (k) = f(k)/g(k)
6. Estimate the split uncertainty $u_{split}(k) = f_1(k)/g(k)$ if k < k* and $u_{split}(k) = f_0(k)/g(k)$ if k > k*, where k* is the classification threshold (i.e., the number of positive votes required for a "positive" prediction).
7. Choose the profile that yields the smallest sum of squared deviations between the observed and estimated uncertainties between k = K/3 and 2×K/3.
8. Set confidence = 1 – uncertainty.

1. Clark *et al*. Using beta binomials to estimate classification uncertainty for ensemble models. *J Cheminfo* 2014, **6**, 34.


Note
log scale
A
Balanced Training Pool
n = 1223
obs pred
beta neg (train)
beta pos (train)
obs neg
obs pos
B
Balanced Test Set
n = 11525
obs pred
beta pred (train)
obs neg
obs pos
beta neg (train)
beta pos (train)
C
Balanced Training Pool
u_pos
u_neg
u_obs (test)
u_split (train)
u_err (train)
D
Balanced Test Set
u_pos
u_neg
u_err (train)
u_obs (test)
u_split (train)
u_split match is worse
E
Unbalanced Training Pool
n = 1222
beta pred (train)
beta pos (train)
obs neg
obs pos
beta neg (train)
F
Unbalanced Test Set
n = 11526
beta pred (train)
obs pos
obs neg
beta neg (train)
beta pos (train)
G
Unbalanced Training Pool
u_pos
u_neg
u_err (train)
u_split (train)
u_obs (test)
H
Unbalanced Test Set
u_pos
u_neg
u_split matches better
u_err (train)
u_split (train)
u_obs (test)
Count
Uncertainty
Tally of Positive Votes