Multiple Outcomes
A 'Reverse Regression' Approach

(Hyderabad) Pipeline Documentation

Last Updated: October 6, 2017

1 Data Preparation [Hyderabad-Specific]

1.1 Data Formatting

Read in/format the raw data - correct 1 miscoded variable[1]

Note. Using the generated dataset we are *with a few exceptions* able to replicate the original results
Note. The below analysis focuses on the categorical outcomes only, i.e while we replicate the results for the continuous *Credit* outcome family (12 non-index outcomes across W1/W2) these are not used in any other part of the analysis

1.2 Prediction Task Set-up

Set-up 17 treatment (T) prediction tasks i of the form $T_i = f(Predictors_i)$

- **Tasks #1 - #15**

  - 1 prediction task for each family of outcomes[2]

  - Each prediction task is defined by:

  (a) **Outcome Predictors (Y_i)**: Vector of individual outcomes in the outcome family including the family-specific Kling outcome index [3,4] (*3-9 outcomes per family*)

[1] Miscoded variable - 'biz-stop-2' (*"Has closed a business in the last 12 months"*)

[2] Outcome families - Credit W1/W2, Employment W1/W2, Employment (families with existing businesses) W1/W2, Employment (families with new businesses) W1, Consumption W1/W2, Social W1/W2, Income W1/W2, Hours Worked W1/W2) (*Note: W1 / W2 outcomes are measured 15-18 months / 2 years after program rollout)* - There are a total of 117 outcomes (102 non-index outcomes (86 unique outcomes (duplicates arise from outcome 'duplication' across the 3 *Employment Outcome* Families and a duplication of the *bizprofit* outcome across the *Employment* and Income Outcome Family) & 15 index outcomes)

[3] Outcome indexes are provided in the original dataset

[4] In the case of the *Consumption* outcome family the original regressions treat one of the individual outcomes (*Monthly Total Per Capita Expenditure*) as the 'index' - this does not affect the set-up of the prediction task which either way includes *all* outcomes in the outcome

(b) **Sample Size Restrictions - Sample Related**: Sample restrictions where applicable, i.e. in the case of 3 *Employment* outcome families the sample is limited to households with old/new businesses. (*Note: Sample size is not restricted for outcome families containing individual outcomes which are naturally only available for a subset of observations (e.g. child/teen specific-outcomes in the Social and Hours Worked family). Unavailable observations for these outcomes are treated as missing)*

(c) **Sample Size Restrictions - Attrition Related**: Sample size restrictions arising from attrition between the two endline surveys, i.e. wave 2 prediction tasks are associated with fewer observations than wave 1 prediction tasks. (*Note that this restriction can be ignored in all cases where only complete observations are used (the relevant observations will automatically be omitted) - as is the case in e.g. the original paper - but cannot be ignored in cases where missingness dummies/median imputation are used*)

- **Task #16 - #17**

  - 2 prediction tasks incl. *all* 'meaningful' outcomes, i.e. using as the outcome predictor (Y_i) a vector of *all unique* outcomes and outcome indexes with the exception of the *Credit W1/W2* outcomes and the 3 sample-restricted *Employment* outcome families (W1/W2) (which effectively represent subsets of the non-sample restricted *Employment* outcome family)

  - Prediction task #16 uses all 76 unique outcomes (incl. 10 index outcomes) / Prediction task #17 uses only those 59 outcomes (incl. 10 index outcomes) which are available, theoretically, for the full sample (i.e. excludes 1 out of the 16 individual *Employment* outcomes (specific to old businesses in W1), 12 out of the 16 individual *Social* outcomes and 4 out of the 16 individual *Hours Worked* outcomes (these are child/teen-specific))

Each of the 17 prediction tasks is further defined by:

- **Sampling Weights**: A (separate) observation-level sampling weight for W1 and W2 prediction tasks

- **Clusters**: Observation-level neighbourhood identifier

- **Control Variables (X)**: 6 cluster-level control variables capturing neighbourhood (cluster) baseline characteristics [5]. *Note: Treatment (T) is assumed to be (by design) orthogonal to the control variables, i.e. the control variables (X) are assumed to be balanced across treatment conditions – an assumption which is a necessary condition for the experimental impact estimates to be valid*

family

[5] Control variables - Area Population, Area Total Debt, Area Total Business, Area Mean Expenditure Per Capita, Area Literacy Rate (Household Heads), and Area Literacy Rate (All Adults)

- **Missingness Dummies ($X0_i$)**: Vector of observation-level binary missingness indicators (0 if non-missing, 1 if missing) for each outcome predictor/control variable in Y_i/X with at least 1 missing observation [6]. *Note: Treatment (T) is not assumed to be orthogonal to missingness, i.e. the missingness dummies ($X0_i$) are not assumed to be balanced across treatment conditions. It is assumed that conditional on $X0_i$ treatment (T) is randomly assigned.*

2 Ensemble Construction [Generic]

2.1 Overview

Construct 3 ensembles for each of the 17 prediction tasks (i):

- **Signal Ensemble**: $T = f(Y_i, X0_i, X)$

- **Control Ensemble (1)**: $T = f(X)$ [Using only complete observations]

- **Control Ensemble (2)**: $T = f(Y_i, X0_i)$ [Using only the subset of Y_i related missingness dummies, i.e. a subset of the $X0$ covariates]

2.2 Approach

To construct the ensembles - adopt one of the following permutation strategies:

- **Hold-Out Approach**: Holdout-set & OOS predictions

  Learners are fitted using all training observations / Each learner (requiring tuning) is fitted $((k^2+k)*n+k+1)*1$ times(1,806 times with default settings)*

- **CV Approach**: *No* Holdout-set & CV-based (OOS) predictions

  Learners are fitted using $(k-1/k)*100\%$ of training observations / Each learner (requiring tuning) is fitted $((k^2)*n+k)*p$ times (752,500 times with default settings)*

- **'Optimized' Hold-Out Approach**. Holdout-set & OOS predictions

  Learners are fitted using all training observations / Each learner (requiring tuning) is fitted $((2k)*n+k+1)*1$ times(606 times with default settings)*. *(Note: This approach results in a biased loss/performance measure (permutation test validity is unaffected))*

- **'Optimized' CV Approach**. *No* Holdout-set & CV-based (OOS) predictions

  Learners are fitted using $(k-1/k)*100\%$ of training observations / Each

[6] Missing observations in the original variables are replaced with the variables' median value (using only the training observations (*where applicable*))

learner (requiring tuning) is fitted $((k) * n + k) * p$ times (152,500 times with default settings)*. *(Note: This approach results in a biased loss/performance measure (permutation test validity is unaffected))*

* In a scenario with (i) random grid tuning with n draws and k-fold CV, (ii) p permutations, and (iii) a single '(Baseline) Signal Permutation Test' with no unbalanced, non-outcome covariates ($X0$) [Default settings: n=60, k=5 and p=500)]. *See Section 3 for more detail*

2.3 Construction

(A) **Hold-Out Set/Fold Generation - All Approaches**

(1) **[Hold-Out Set]** IF using a hold-out set: Split the dataset (taking into account sample size restrictions) into training and testing data (keeping cluster observations together)

IF *not* using a hold-out set: Designate the entire dataset (taking into account sample size restrictions) as training data

(2) **[Folds]** Split the training data into k-folds (keeping cluster observations together)

(B) **Estimation - Option #A: Hold-Out / CV Approach**

(1) [**Inner Tuning**] For each learner (j):

(I) For each fold (m) of the training data:

(i) Tune $learner_j$ using all $observations_{Fold!=m}$ and k-fold CV (obj: minimize the chosen loss function) (*Note: If no tuning is to be performed - select the default parameters (no estimation needed)*)

(ii) Fit tuned $learner_j$ using all $observations_{Fold!=m}$

(iii) Use tuned/fitted $learner_j$ to generate (OOS) predictions for all $observations_{Fold=m}$

(II) Combine $learner_j$'s predictions to generate $\hat{T}_j$ for all training observations

(2) [**Stacking**] Stack the j learners by fitting a logistic / non-negative least squares 'meta-leaner' using the individual learners' predictions ($\hat{T}_j$) as predictors (obj: minimize the log-loss/constrained L2 loss)

(3) [**Outer Tuning** - *Hold-Out Approach only*] For each learner j:

(i) Tune $learner_j$ using all training observations and k-fold CV (obj: minimize the chosen loss function) (*Note: If no tuning is to be performed - select the default parameters (no estimation needed)*)

(ii) Fit tuned $learner_j$ using all training observations

(4) Return the fitted 'meta-learner' derived in *Step-2* and the tuned/fitted individual learners derived in (i) *Step-3* IF using the *Hold-Out Approach* / (ii) derived in *Step-1* IF using the *CV Approach* (*k versions of each learner (different tuning parameters & different coefficients, etc.) (1 per fold)*)

(B) **Estimation - Option #B: 'Optimized' Hold-Out / CV Approach**

(1) [**Outer Tuning**] For each learner j:
   (i) Tune $learner_j$ using all training observations and k-fold CV (obj: minimize the chosen loss function) (*Note: If no tuning is to be performed - select the default parameters (no estimation needed)*)

(2) Split the training data into k *different* folds (keeping cluster observations together)

(3) [**Outer Tuning**] For each learner j:
   (I) For each fold (m) of the training data:
      (i) Fit tuned $learner_j$ using all $observations_{Fold!=m}$
      (ii) Use tuned/fitted $learner_j$ to generate (OOS) predictions for all $observations_{Fold=m}$
   (II) Combine $learner_j$'s predictions to generate $\hat{T}_j$ for all observations

(4) [**Stacking**] Stack the j learners by fitting a logistic / non-negative least squares 'meta-leaner' using the individual learners' predictions ($\hat{T}_j$) as predictors (obj: minimize the log-loss/constrained L2 loss)

(5) [**Outer Tuning** - *Hold-Out Approach only*] For each learner j:
   (i) Fit tuned $learner_j$ using all training observations

(6) Return the fitted 'meta-learner' derived in *Step-4* and the tuned/fitted individual learners derived in (i) *Step-5* IF using the *Hold-Out Approach* / (ii) derived in *Step-3* IF using the *CV Approach* (*k versions of each learner (same tuning parameters & different coefficients, etc.) (1 per fold)*)

(C) **Prediction - All Approaches**

(1) Combine the 'meta-learner' and the individual learners into a function that generates $\hat{T}_{Ensemble}$ for any dataset containing the predictors (Y, X and $X0$) following the below steps:
   (I) Predict T using (separately) each of the tuned/fitted learners ($\hat{T}_{Learner(j)}$) (applied separately to each of the (given) k folds of the dataset in the case of the 'CV Approach' / the 'Optimized CV Approach')
   (II) Generate $\hat{T}_{Ensemble}$ using the fitted 'meta-learner' to predict T based on the $\hat{T}_{Learner}$ predictors

3 (Permutation) Tests [Generic]

3.1 Overview

Depending on the structure of the covariates Y_i, $X0_i$, and X (which may vary across prediction tasks) different testing procedures are appropriate.

The below section outlines (i) a set of permutation tests representing the theoretically optimal but in some cases unfeasible testing strategy (*Section 3.2*) and (ii) a number of alternative testing strategies applicable across a broader range of covariate structures (*Section 3.3*)

3.2 Permutation Tests - The Theoretical Optimum

3.2.1 Overview

Given (balanced) control variables (X) *and* unbalanced covariates ($X0_i$) (e.g. in the form of missingness indicators) whereby *X **and** $X0_i$ are constant within clusters, i.e. vary only across clusters* - perform the following permutation tests:

- **(Baseline) Balance Test** (*H1: X predicts T better than chance*) [KEY TEST # 1]
  - *Tasks/Ensemble*: Prediction tasks #16, #17 only / Control Ensemble (1)
  - *Permutation*: Permute T against X
  - *Motivation*: Fail to reject H0 ⇒ Baseline balance
- **(Baseline) Signal Test** (*H1: Y, X and X0 jointly predict T better than X0*)
  - *Tasks/Ensemble*: All prediction tasks (#1 - #17) / Signal Ensemble
  - *Permutation*: Permute (T, X0) (jointly) against (Y, X)
  - *Motivation*: Reject H0 ⇒ Predictability (*not* driven (solely) by unbalanced, non-outcome covariates ($X0$))

  The below 3 permutation tests serve as a robustness check and provide additional insights into the relative importance of the different covariates
- **Supplementary Signal Test (a)** (*H1: Y, X and X0 jointly predict T better than chance*) [KEY TEST # 2]
  - *Tasks/Ensemble*: All prediction tasks (#1 - #17) / Signal Ensemble
  - *Permutation*: Permute T against (X, X0, Y)
  - *Motivation*: Reject H0 ⇒ Predictability (driven by the outcome predictors (Y) and/or the covariates (X, $X0$))
  - *Note*: Test of limited practical utility
- **Supplementary Signal Test (b)** (*H1: Y, X and X0 jointly predict T better than X0 and X (jointly)*)
  - *Tasks/Ensemble*: All prediction tasks (#1 - #17) / Signal Ensemble

- *Permutation*: Permute (T, X0, X) (jointly) against Y

- *Motivation*: Reject H0 ⇒ Predictability (driven solely by the outcome predictors (Y))

- **Supplementary Signal Test (c)** (*H1: Y and X0 jointly predict T better than X0*)

- *Tasks/Ensemble*: All prediction tasks (#1 - #17) / Control Ensemble (2)

- *Permutation*: Permute (T, X0) (jointly) against Y

- *Motivation*: Reject H0 ⇒ Predictability (driven solely by the outcome predictors (Y))

3.2.2 Details & Qualifications

- **Experimental Design Motivation**: In an experimental setting the assumption that the X covariates are orthogonal to treatment T - an assumption which the *Baseline Balance Test* is designed to test - is *necessary* for the experimental impact estimates to be valid.

  The (non-)orthogonality of the $X0$ covariates with respect to T on the other hand - explored by the 4 supplementary tests - is *not* critical for the validity of the estimates. Instead the (non-)orthogonality of the $X0$ covariates is of interest to the extent that it helps to explain *why* there is signal (i.e. is it because the distribution of outcomes is actually different, or merely because the pattern of missingness (for Y and/or X) is different across treatment groups).

- **Permutation Structure.** Permutations are (i) performed p times and are (ii) performed at the cluster level[7]

- **Baseline Balance & Test Interpretation.** A failure to reject H0 in the *(Baseline) Balance Test* effectively implies that there is no predictability of T based on the X covariates. This has implications for the interpretation of e.g. the *(Baseline) Signal Test* which can, conditional on baseline balance, be interpreted as a test of the predictability of T based on the outcome predictors (Y) (effectively the equivalent of the *Supplementary Signal Test (b)*) (Note: In interpreting different permutation tests jointly 'pre-testing' needs to be guarded against)

- **Absence of Non-Outcome Covariates.** The outlined permutation tests remain appropriate in set-ups where one or both of types of non-outcome covariates (X and $X0_i$) are non-existent. In such cases the 4 tests effectively collapse into a smaller number of distinct tests, e.g. with no unbalanced covariates ($X0_i$) the *Supplementary Signal Test (a)* corresponds to the *(Baseline) Signal Test* (and would not need to be performed separately)

[7] Values of each variable being permuted are re-assigned at the cluster level (e.g. observa-

3.2.3 Implementation

Each of the permutation tests results in a p-value given by $P(Loss_{Ensemble} >= Loss_{Permuted})$.

- $Loss_{Ensemble}$

  - *Hold-Out / 'Optimized' Hold-Out Approach:* Function of (i) the ensemble predictions ($\hat{T}_{Ensemble}$) for the hold-out set and (ii) the realized value of $T_{Holdout}$

  - *CV / 'Optimized' CV Approach:* Function of (i) the ensemble predictions ($\hat{T}_{Ensemble}$) for the training data and (ii) the realized value of $T_{Training}$

  Note: The ensemble predictions are derived using the *original* Ensemble Prediction Function, i.e. the ensemble constructed using the (training) data as described in Section-2.3

- $Loss_{Permuted}$

  - *Hold-Out / 'Optimized' Hold-Out Approach:*

  Function of (i) the ensemble predictions ($\hat{T}_{Ensemble}$) *for the permuted holdout data* and (ii) the realized value of $T_{Holdout-Permuted}$. In the case of the *(Baseline) Balance Test* and the *(Supplementary) Signal Test (a)* the ensemble predictions are constructed using the *original* Ensemble Prediction Function whilst in the case of the *(Baseline) Signal Test*, *(Supplementary) Signal Test (b)* and the *(Supplementary) Signal Test (c)* the ensemble predictions are constructed using a *permutation-specific* Ensemble Prediction Function, i.e. an ensemble constructed separately for each permutation using the permuted training data[8]

  - *CV / 'Optimized' CV Approach:*

  Function of (i) the ensemble predictions ($\hat{T}_{Ensemble}$) *for the permuted training data* and (ii) the realized value of $T_{Training-Permuted}$. Across all tests the ensemble predictions are constructed using a *permutation-specific* Ensemble Prediction Function, i.e. an ensemble constructed separately for each permutation using the permuted training data

3.3 Alternative Tests

The permutation tests described in Section 3.2 are not applicable in the context of certain covariate structures, notably in the case where $X0_i$ *is* ***not*** *constant within clusters,i.e. varies both within and across clusters* (while X varies only at the cluster level as previously assumed). In this case all tests, with the

tions in cluster m are all (re-)assigned the value of T in cluster h)

[8] Re-estimation of the ensemble is necessary given that the permutation of X and/or $X0$ (alongside T) gives rise to a new DGP by altering the co-variance structure of the predictors. Given that the permuted hold-out set cannot be seen as a draw from the same DGP as the

exception of the *(Baseline) Balance Test* are unfeasible[9] and alternative testing procedures - associated with drawbacks - need to be adopted. This section outlines 2 potential strategies.

- **Complete-Case Analysis**: Use only the subset of observations which are complete to construct and assess the ensemble (using the 4 signal-related permutation tests).

  - *Motivation*: Dropping all incomplete observations effectively conditions on missingness, i.e. removes the need for missingness indicators ($X0$)

  - *Potential Problems*: If missingness is spread across predictors dropping all incomplete observations can result in a substantial reduction in sample size (with implications for the power of the permutation tests)

- **Mean Difference Test**: Replace each of the 4 signal-related permutation tests with a 'mean difference test': For each of the tests construct a *single* ensemble using the permuted data and use this ensemble to derive the $Loss_{Permuted}$. Obtain the observation-level differences between the $Loss_{Ensemble}$ and the $Loss_{Permuted}$. Finally, perform a two-sided parametric t-test on the loss-difference (deriving $\mu_{loss-difference}$ and $\sigma_{loss-difference}$ from the data) to obtain a p-value for each test

  - *Motivation*: Standard test for the difference in means

  - *Potential Problems*: (i) This approach requires the use of the Hold-Out / 'Optimized' Hold-out Approach[10], (ii) The test is based on a comparison of the ensemble to a *single* 'permuted predictor' the construction of which is non-deterministic introducing the risk of false negatives, i.e. the risk of a failure to find signal when there is signal in cases where the 'permuted predictor' happens by chance be a poor predictor, and (iii) Different tests are needed for different loss functions, i.e. the above outlined procedure using a t-test works only if using an L2 loss

4 Assessment [Hyderabad-Specific]

Assess the performance of the reverse regression approach by comparing the obtained p-values to the p-values obtained by the authors using classical methods

4.1 Comparison #1 - Baseline Balance

- *Reverse Regression Approach*: P-value derived from the *Baseline Balance Test*

- *Classical Approach*: P-values derived from a balance test for each control variable (X) (Two-sided t-test on the mean treatment-control difference)

training data the ensemble needs to be re-estimated

[9]This follows from the observation that there is, in this case, no way of permuting the unbalanced covariates ($X0$) at the cluster level

[10]The test is not applicable in the case of the CV / 'Optimized' CV approach given the

4.2 Comparison #2 - Family-Level Significance

For each outcome family:

- *Reverse Regression Approach*: P-value derived from the relevant *Baseline Signal Test*

- *Classical Approach*: (a) Uncorrected p-values for each of the individual outcomes in the outcome family, (b) Corrected p-values for each of the individual outcomes (Hochberg Step-Down Correction (1988))[11], and/or (c) Uncorrected p-value for the family-specific outcome index

4.3 Comparison #3 - Overall Significance

- *Reverse Regression Approach*: P-values derived from the relevant *Baseline Signal Tests* (Prediction task #16 and #17)

- *Classical Approach*: (a) Corrected p-values for each of the outcome indexes with the exception of the *Credit W1/W2* outcome indexes (Hochberg Step-Down Correction (1988))

correlation of losses across folds

[11] Corrected p-values are not reported in the original paper. The correction implemented follows the procedure outlined in the paper (and used to correct the outcome index p-values)