Permutation Test for Distribution Shift via Random Forest

Heqiao Ruan, Lucas Mentch

Department of Statistics, University of Pittsburgh

Overview

- For an already deployed model, we will not know how the next batch of data differs from the existing data, so understand the **distribution shift** would be one of the most important problems for **machine learning practitioners**.
- We propose a computationally efficient, easy-to-implement permutation test **PermOOB** to detect **distribution shift** using random forest.
- The proposed permutation test for distribution shift can **extend to arbitrary machine learning model** with a separate validation set.

Motivation

- Data coming in batches, $\mathbf{D}_1 = (X_1, Y_1)$ and $\mathbf{D}_{new} = (X_2, Y_2)$. Types of shifts can vary:
  - Covariate Shift: $P(X_1) \neq P(X_2)$ where $P(Y_1|X_1) = P(Y_2|X_2)$, only covariates distribution change where the conditional dependence remain the same.
  - Concept Drift: $P(Y_1|X_1) \neq P(Y_2|X_2)$ where $P(X_1) = P(X_2)$, only conditional dependence change where the covariate distribution remain the same
  - Target Shift: $P(Y_1) \neq P(Y_2)$ where $P(X_1|Y_1) = P(X_2|Y_2)$, only response distribution will change where the conditional dependence remain the same.
  - Distribution Shift: $P(Y_1|X_1) \neq P(Y_2|X_2)$ and $P(X_1) \neq P(X_2)$, both changes.
- In practice, the distribution shift is prevalent - both the data generating process and the covariate distribution will change across batches of data.
- **Existing methods [2, 3, 4, 5] only focus on certain type of shift. We aim to develop hypothesis testing procedure efficient in most of scenarios.**

Hypothesis Testing Procedure

- Random forest is among one of the most popular machine learning algorithms. Out-of-bag(OOB) error provides a readily available validation set.
- **The intuition of the test**: We fit a model on the current batch of data and perform prediction on the next batch of data. We aim to compare the OOB error(MSE) and the Prediction Error(MSE) on observation level.

Algorithm 1 Permutation Test for Distribution Shift(**PermOOB**)

BEGIN We start from a training data $DTrain = \mathbf{D}(X^T_{train,i}, Y_{train,i}), i = 1, ..., n_{train}$ and the testing data $DTest = \mathbf{D}(X^T_{test,i}, Y_{test,i}), i = 1, ..., n_{test}$ where $X^T_{train,i} = (X_{train,i1}, ..., X_{train,ip})^T$.

S1 Standardize the training data X_{Train} and test data X_{Test} both. Then train a model on $DTrain$ with n_{tree} trees.

S2 For $i = 1, 2, ..., n_{train}$, get the average prediction error on the OOB sample for that observation $MSE_{i,OOB} = (Y_i - \frac{1}{n_{oob,i}} \sum_{j \in OOB} f_j(X_i))^2$ with T trees $f_1, ..., f_T$.

S3 For $i = 1, 2, ..., n_{test}$, get the average prediction error on the predicted sample, denote as $MSE_{i,Pred}$, calculated as $MSE_{i,Pred} = (Y_i - \frac{1}{T} \sum_{j=1}^{T} f_j(X_i))^2$.

S4 Perform the hypothesis testing $H_0 : MSE_{OOB} = MSE_{Pred}$ vs. $H_a : MSE_{OOB} < MSE_{Pred}$: Construct a permutation test based on $L_1 = MSE_{i,OOB}, i = 1, 2, ..., n_{train}$ and $L_2 = MSE_{i,Pred}, i = 1, 2, 3, ..., n_{test}$. Denote the original difference as $d_0 = mean(L_1) - mean(L_2)$ Doing a permutation test for B(by default 5000) times, denote the difference as $d_i = L_{\pi-i,1} - L_{\pi-i,2}$. Calculate the p-value as p-value = $ecdf_{L_\pi}(d_0)$.

END

Simulation Setting

Model Name	Data Generating Process
Mars Model	$Y = 50 * sin(\pi X_1 X_2) + 5 * (X_3 - 0.05)^2 + 5 * X_4 + 5 * X_5$, $X_i \sim N(0, 1)$
Linear Model	$Y = X\beta + \epsilon$, $X \sim N(0, \Sigma_{(i,j)} = \rho^{\lvert i-j \rvert})$, $SNR = \frac{\beta^T \Sigma \beta}{Var(\epsilon)}$
Concept Drift	$Y = (X_5 + X_6 + X_7 + X_8) * (1 - T * \lambda) + X_1 + X_2 + X_3 + X_4$, T is indicator of test data
Mean Shift Mean Shift	Training Data $Mean(X_1, X_2, X_3, X_4, X_5) \sim (1 + \delta, 1 + 2\delta, 1 + 3\delta, 1 + 4\delta, 1 + 5\delta)$ Testing Data $Mean(X_1, X_2, X_3, X_4, X_5) \sim (1 + 5\delta, 1 + 4\delta, 1 + 3\delta, 1 + 2\delta, 1 + \delta)$

Table 1. List of Simulation settings, each of the setting will impose varying number of random noise features, noise levels $var(\epsilon) = \sigma^2$ and dependence level among features ρ.

Simulation Results

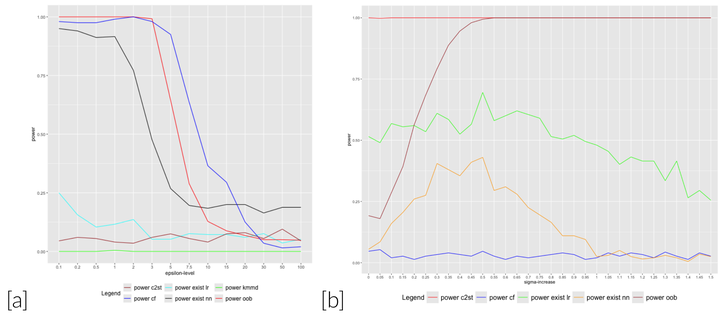

Figure 1. (a) Power Comparison for Concept Drift, $\rho = 0.3$ (b) Power Comparison for Covariate Shift Only, Means Shift on top of Variance Shift

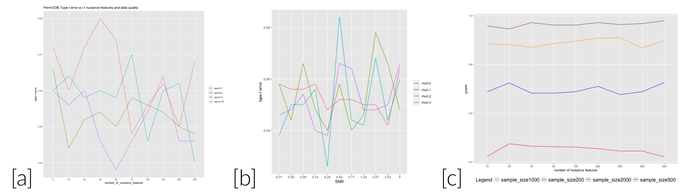

Figure 2. (a) Type-I error for Mars Model w.r.t. different number of random noise features (b) Type-I error for Linear Model w.r.t. different Signal-to-Noise Ratio(SNR) (c) Power for Linear Model w.r.t. different number of random noise features

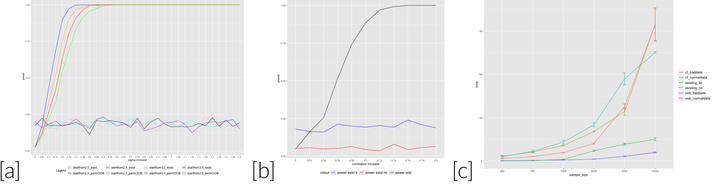

Figure 3. (a) Power Comparison for drop of data quality (b) Power Comparison for increase of dependence level ρ (c) Time Complexity Comparison, for bad data with very high noise level and normal data with moderate noise level.

Real Data Application

We evaluate the performance of permutation tests on a variety of UCI datasets where we use features to split the data into training set and testing set. To select the features to split onto, we regress response on all predictors and select those features with the lowest p-value. Then split based on the median value of that feature. **By construction, distribution shift exists.**

Dataset - Feature	PermOOB	Existing-NN	Existing-LL	CFPerm
AToxic - V1	**1.000**	0.370	0.02	0.125
AToxic - V2	**1.000**	0.265	0.035	0.310
AToxic - V4	**1.000**	0.468	0.026	0.595
Autoprice - V7	**1.000**	0.445	0.000	0.175
Autoprice - V8	**1.000**	0.500	0.063	0.220
Autoprice - V10	**1.000**	0.410	0.038	0.295
Boston - rm	**1.000**	0.483	0.115	0.565
Boston - nox	**1.000**	0.469	0.027	0.110
Boston - crim	**1.000**	0.37	0.020	0.275
StudentPerf - higher	**1.000**	0.493	**1.000**	0.475
StudentPerf - famsup	**1.000**	0.225	0.007	0.110
ethanol - X	**0.622**	0.580	0.440	0.555
airfoil - X5	**1.000**	**1.000**	0.868	0.985
airfoil - X4	**1.000**	0.996	0.764	0.885
airfoil - X2	0.066	**0.068**	0.046	0.060

Table 2. Comparison for power of detecting the covariate shift on more **UCI datasets** - made to split the dataset based on the features that contributes most significantly to the response: lowest p-value in regress response on the predictors

PermOOB achieve higher power than existing methods in most cases.

Theoretical Guarantee

Asymptotic Normality of the random forest Prediction via U-statistics.

Theorem1: Asymptotic Normality for random forest on a single observation:
Suppose that $X_1, ..., X_n$ are i.i.d from F_X and that $U_{n,\alpha*n=s,n_{tree},w}$ is a generalized incomplete U-statistic with kernel $h = h(X_1, ..., X_n; w)$. Assume η is the subsampling proportion in growing the tree, n_{train} is the number of observations in the training data and w is the randomization parameter. Let $\theta = E[h], \xi_s = var(h)$ and $\xi_{1,w} = E[g^2(Z_1)]$ where $g(z) = E[h(z, Z_2, ..., Z_s; w)] - \theta$. Suppose further that $\xi_s < \infty$ and $\xi_{1,w} > 0$. If $|h - \theta|^2$ is sub-Gaussian after standardization(bounded variance/third moment - reasonable assumption in most of the data) for k = 2, 3 and all s, then for any $0 < \eta, \eta_0 < \frac{1}{2}$, Then the prediction value on the i-th observation in the test data follows the asymptotic normality, this is for the out-of-bag(OOB) error version where each observation is expected to appear in $B(1 - \frac{s}{n})$ trees for OOB error and $\Phi(z)$ is the cdf of standard normal distribution. Here $C > 0$ is some constant and $N_i \xrightarrow{P} B(1 - \frac{s}{n})$

$$sup_{z \in R} |\frac{U_{n,s,N_i,w}(X_{test,i}) - \theta}{\sqrt{s^2 \xi_{1,w}/n + \xi_s/N_i}}| - \Phi(z)| + o_p(1)$$

$$\leq C\{\frac{E|g|^3}{\sqrt{n(E|g|^2)^{3/2}}} + \frac{E[|h - \theta|^3]}{N_i^{1/2}(E[|h - \theta|^2])^{3/2}} + [\frac{s}{n}(\frac{\xi_s}{s * n\xi_{1,w}} - 1)]^{1/2}$$

$$+ (N_i)^{-1/2+\eta_0} + (\frac{s}{n})^{-\frac{1}{2}+\eta}\}$$

Asymptotic Validity of Permutation Test

Theorem2: Given the following assumptions on top of those in the previous theorem: (1)Subsampling proportion in random forest $\eta = k_n = o(\sqrt{n})$ (2)Prediction value for random forest value $RF_B(X_i)$ and $RF_B(X_j)$ asymptotically independent. Then for the following null hypothesis: $H_0 : E[MSE_{OOB}(X_{Train}, Y_{Train})] = E[MSE_{Pred}(X_{Test}, Y_{Test})]$ using the statistic $\hat{\Delta} = MSE_{Pred}(X_{Test}, Y_{test}) - MSE_{OOB}(X_{Train}, Y_{Train})$. Under H_0 the permutation distribution $\sqrt{n_1 + n_2}\hat{\Delta} \sim N(0, \sigma^2)$, $\sigma^2 = f'(E[RF_B(x)])^2 \sigma^2_{RF-B} + f'(E[RF_{OOB}])^2 \sigma^2_{RF-(OOB)}$ where $f(RF_B(X), Y) = (Y - RF_B(X))^2$. The permutation test attains the asymptotic Type-I error rate.

Conclusion & Practical Considerations

Conclusion

- PermOOB is very easy to implement in practice and very fast in computation.
- PermOOB can achieve good type-I error control where the existing method may have slightly high type-I error under some component methods.
- PermOOB works better than existing methods in detecting **concept drift**. PermOOB achieves higher power than existing methods on most of the benchmark datasets.
- PermOOB can achieve decent power under **covariate shift only**, especially under mean shift. It can help detect **drop of data quality** as well as the increase of **dependence level** ρ where none of the existing method can do this.

Practical Considerations

- Practically, the proposed permutation test can serve as a pre-screening procedure for **any already deployed Machine Learning Models** - If **permOOB** not reject H_0 then we will make a concrete argument that there's no distribution shift among batches of data.
- Extend beyond the scope of random forest - as long as one have a separate validation set, the permutation test can be extended to **any machine learning model on regression task.**
- In the future, we aim to extend the permutation test to classification problems.

References

[1] Tim Coleman, Wei Peng, and Lucas Mentch.
Scalable and efficient hypothesis testing with random forests.
Journal of Machine Learning Research, 23(170):1–35, 2022.

[2] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola.
A kernel two-sample test.
The Journal of Machine Learning Research, 13(1):723–773, 2012.

[3] Xiaoyu Hu and Jing Lei.
A two-sample conditional distribution test using conformal prediction and weighted rank sum.
Journal of the American Statistical Association, 119(546):1136–1154, 2024.

[4] David Lopez-Paz and Maxime Oquab.
Revisiting classifier two-sample tests.
arXiv preprint arXiv:1610.06545, 2016.

[5] Jian Yan and Xianyang Zhang.
A nonparametric two-sample conditional distribution test.
arXiv preprint arXiv:2210.08149, 2022.

Email: [1] :her76@pitt.edu, [2] :lkm31@pitt.edu