`pact`: Predictive Analysis of Clinical Trials

Richard Simon
Jyothi Subramanian

April 15, 2016

1 Introduction

The objective of this vignette is to demonstrate the `pact` R package. The methodology behind the functions in `pact` will also be described briefly. For a more thorough scientific description of `pact` including applications, the reader is referred to [1].

2 Outline of Methodology

Based on response and covariate data from a randomized clinical trial comparing a new experimental treatment E versus a control C, the purpose behind the functions in `pact` is to develop and internally validate a classifier that can identify subjects likely to benefit from E rather than C. Currently, survival and binary response types are permitted. Covariate data can be high-dimensional as well, and currently, the dimension reduction techniques lasso and univariate variable selection are implemented. These dimesion reduction options can be used with low-dimensional covariates too if the user so desires. The user can optionally specify a second (small) set of prognostic variables to always remain in the model. This set of variables will not be subjected to variable selection.

In the case of a survival response, a Cox proportional hazards (PH) regression model is developed using data from all subjects in both E and C groups. Main effect of treatment, main effect of the fixed prognostic covariates, main and treatment by covariate interactions for the remaining covariates are considered for the model development:

$$\log\left[\frac{h(t)}{h_0(t)}\right] = \alpha z + \bar{\beta}_0' \bar{x_f} + \bar{\beta}_1' \bar{x_v} + \bar{\gamma}' z \bar{x_v} \tag{1}$$

Here z is a treatment indicator with $z = 0$ for subjects assigned to group C and $z = 1$ for subjects assigned to group E. $\bar{x_f}$ denotes the vector of prognostic covariates that are fixed to remain in the model and $\bar{x_v}$ denotes the vector of the second set of covariates for which variable selection may be applied. So, $\bar{x_v}$ can be high-dimensional as well. Model (1) can be fit by maximizing the penalized log partial likelihood.

The evaluation of the model is done using K-fold cross-validation (CV). The difference in the log hazard for a subject with covariate vectors $\bar{x_f}$ and $\bar{x_v}$ receiving treatment E as compared to receiving treatment C can be estimated by $\delta(\bar{x_v}) = \hat{\alpha} + \hat{\bar{\gamma}}'\bar{x_v}$. $\delta(\bar{x_v})$ is referred as the *predictive score* for the subject with covariate vectors $\bar{x_f}$ and $\bar{x_v}$. Lower predictive scores are indicative of benefit with E. Note that the expression for the predictive score does not explicitly contain terms involving $\bar{x_f}$. In each CV fold, the PH model (1) is developed from the training set. Variable selection, if any, is performed within the training set and estimates $\hat{\alpha}$ and $\hat{\bar{\gamma}}$ are found. These estimates are used to calculate the predictive scores for subjects in the test set. This is repeated for all the CV folds. Cross-validated predictive scores are thus obtained for all the subjects in the dataset. Various evaluation statistics can be calculated from these cross-validated predictive scores to give unbiased estimate of the performance of the model (1) for future samples.

In the case of a binary response variable, a logistic regression model is developed instead of a PH regression model:

$$\log\left[\frac{p}{1-p}\right] = \alpha z + \bar{\beta}_0'\bar{x_f} + \bar{\beta}_1'\bar{x_v} + \bar{\gamma}' z\bar{x} \tag{2}$$

Here p is the probability of a response. The other steps are the same as for survival response. Also, in the case of a binary response, higher predictive scores are indicative of benefit with E.

3 Usage and Examples

3.1 Survival Response and High-dimensional Covariates

The GSE10846 dataset is used to illiustrate the application of `pact` with a survival response variable and high-dimensional covariates. This dataset contains data on survival, treatment and gene expression of 1000 genes for 412 subjects with diffuse large B cell lymphoma. The subjects were randomized to treatment with either CHOP (control treatment, C) or CHOP+Rituximab (new treatment, E).

We first load the pact package and the GSE10846 dataset.

```
> library("pact")   ### Load the "pact" R-package
> data("GSE10846")  ### Load the dataset
> GSE10846[1:5,1:5] ### Display a piece of the data
```

```
          time status Treatment 224588_at 224590_at
GSM274895 2.68      1         0     3.838     2.070
GSM274896 0.82      1         0     2.868     5.049
GSM274897 2.54      1         0    14.525    11.337
GSM274898 9.67      0         0    13.112     9.792
GSM274899 4.83      0         0     3.973     4.278
```

The next step is to prepare the response Y, the covariates X_f, X_v and the $Treatment$ variables. For a survival type response, Y should be a two-column matrix with the columns named 'time' and 'status'. 'time' is the column of survival times and 'status' is a binary variable, with '1' indicating death, and '0' indicating right censored. Here, we do not have info on any prognostic variables that should be kept fixed in the model, hence X_f is NULL (which is the default). X_v is the *nobs* by p dataframe of covariates to be used for model development. Each row in Y and X_v corresponds to the data for a subject and each column in X_v is a covariate. Additionally, $Treatment$ is a *nobs* length treatment indicator, which is a factor, with a 1 indicating that the subject was assigned to treatment E and 0 indicating that the subject was assigned to treatment C.

```
> Y <- GSE10846[,1:2]    ## Response, survival status
> Treatment <- as.factor(GSE10846[,3]) ## Treatment information
> Xv <- GSE10846[,-c(1:3)]    ## Covariates
> ## No Xf. So Xf=NULL, the default
```

Once the variables are defined, a predictive model can be fit to the full dataset using function pact.fit. Two variable selection method options are currently provided in pact.fit to facilitate analysis of data with high-dimensional ($p > nobs$) covariates. The variable selection options can be specified using the varSelect argument. The possible options for varSelect are c("none", "univar", "lasso"). Note that these variable selection can be used with low-dimensional data too, if the user so desires.

No variable selection is performed if varSelect = "none". If varSelect = "univar", univariate variable selection is performed. For each covariate Xv_i, a regression model is developed that includes, $Treatment$, Xv_i and $Treatment * Xv_i$ interaction (if Xf is not

NULL, the main effect of variables in Xf too are included in this regression model). The nsig Xv_is that have the lowest $Treatment * Xv_i$ interaction p-values are then used to develop the final predictive model. The variable selection parameter, nsig is set by the user.

The output from pact.fit is an object of class pact. Objects of class pact have summary, print and predict methods defined.

```
> ### Fit predictive model using univariate variable selection
> p1 <- pact.fit(Y=Y, Xv=Xv, Treatment=Treatment, family="cox",
       varSelect="univar", nsig=5)
```

```
Variable selection: Univariate method...
```

```
> summary(p1) ## Display model coefficients
```

```
              T1    `1552531_a_at`      `1553604_at`     `219737_s_at`
     -2.73090690        0.08112818       -0.18694323       -0.11036970
     `242334_at`       `243905_at` T1:`1552531_a_at`   T1:`1553604_at`
     -0.04400722       -0.04255307       -0.27369178        0.24522308
 T1:`219737_s_at`    T1:`242334_at`    T1:`243905_at`
      0.30161537       -0.02197103        0.31287453
```

```
> print(p1)   ## Print the classification function
```

```
Call:  pact.fit(Y = Y, Xv = Xv, Treatment = Treatment, family = "cox",      varSelect = "univar",



family:  cox



Classification function for classifying future subjects:
 f =  ( -2.7309 )
  +  (-0.2737 )*`1552531_a_at`
  +  (0.2452 )*`1553604_at`
  +  (0.3016 )*`219737_s_at`
  +  (-0.022 )*`242334_at`
  +  (0.3129 )*`243905_at`
```

```
> ### Model can be used to predict score for new subjects
> r <- rnorm(ncol(Xv))   ## Generate dummy covariate data for one new subject
> newXv <- Xv[1,]+r
> rownames(newXv) <- "New"
> predict(p1, newXv)     ## Now predict scores for this subject
```

```
      New
-1.132003
```

If the option `varSelect = "lasso"` is chosen, a penalized regression is carried out with the penalty factor chosen through a cross-validation procedure. The R functions `glmnet` and `cv.glmnet` [2, 3] are used in this case. The variable selection parameter, `penalty.scaling` decides the amount of penalty to be applied to main effect coefficients as compared to interaction coefficients for Xv. The default value for `penalty.scaling` is 0.5, which implies that the main effect coefficients are penalized half as much as the interaction coefficients. This default can be changed by the user. Variables, if any, in Xf are not penalized.

```
> ### Fit predictive model using "lasso" with peanlty.scaling = 2
> p2 <- pact.fit(Y=Y, Xv=Xv, Treatment=Treatment, family="cox",
       varSelect="lasso", penalty.scaling=2)
```

```
Variable selection: lasso...
```

```
> summary(p2) ## Display coefficients
```

```
17 x 1 sparse Matrix of class "dgCMatrix"
                              1
T1                -0.5408140673
`1553499_s_at`    -0.0387235573
`236981_at`       -0.0059501356
`208168_s_at`     -0.0078692660
`231049_at`       -0.0129423502
`210546_x_at`      0.0020755888
`240898_at`       -0.0029054260
`240777_at`       -0.0014960815
`237493_at`       -0.0222474024
`223484_at`       -0.0110271197
`216233_at`        0.0297970065
`212353_at`       -0.0513873141
T1:`1552531_a_at` -0.0182393371
T1:`231898_x_at`   0.0050718672
T1:`231391_at`    -0.0080538839
```

```
T1:`1563001_at`    0.0007773538
T1:`242107_x_at`   0.0107091506
```

```
> print(p2)   ## Print classification function
```

```
Call:  pact.fit(Y = Y, Xv = Xv, Treatment = Treatment, family = "cox",      varSelect = "lasso", p



family:  cox



Classification function for classifying future subjects:
 f =  ( -0.5408 )
  +  (-0.0182 )*`1552531_a_at`
  +  (0.0051 )*`231898_x_at`
  +  (-0.0081 )*`231391_at`
  +  (8e-04 )*`1563001_at`
  +  (0.0107 )*`242107_x_at`
```

3.2 Cross-validation and Model Evaluation

The function `pact.cv` computes the cross-validated predictive score for each subject using K-fold cross-validation, with the same model development parameters as in `pact.fit`. Evaluations of the cross-validated scores are performed using function `eval.pact.cv`.

```
> ### Cross-validate the 'pact' model, p1
> cv1 <- pact.cv(p1, nfold=5)
```

```
> ### Evaluate with method="discrete" (Figure 1)
> e1 <- eval.pact.cv(cv1, method="discrete", g=log(0.80), perm.test=FALSE)
```

Two methods are currently implemented for computing the evaluation statistics from the cross-validated predictive scores, specified by setting the value for `method`. In `method="discrete"`, the user specifies a value for the cutpoint `g` to be applied to the cross-validated score to determine whether a subject can be considered to benefit from E or not. For `family="cox"`, the predictive scores represent the change in the log hazard with treatment E as compared to treatment C. Hence, a cutoff `g = log(0.80)`, for example, implies that subjects predicted to receive at least 20% reduction in HR with E are classified to 'benefit' from E. Kaplan-Meier curves by *Treatment* are plotted for the subjects predicted to be in

```
Call:  eval.pact.cv(out.cv = cv1, method = "discrete", g = log(0.8),      perm.test = FALSE)



family:  cox

Log-rank statistic (LR) comparing E and C in group predicted to benefit from E: 14.5984
Log-rank statistic (LR) comparing E and C in group predicted to not benefit from E: 1.6191
```

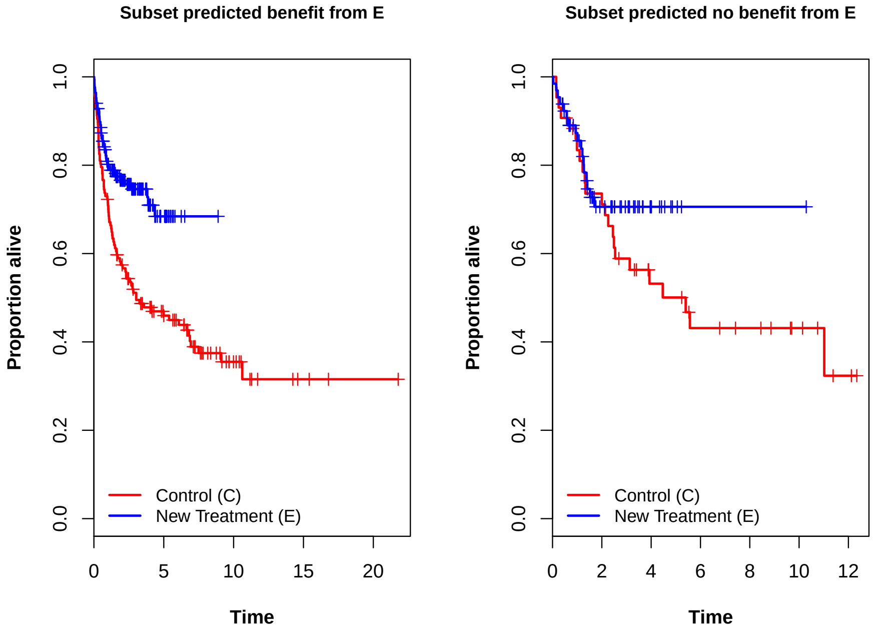

Figure 1: Figures after evaluation with `method="discrete"`

the ’benefit’ and ’no benefit’ groups (Figure 1). Log-rank statistics are computed for the ’benefit’ and ’no benefit’ groups.

In `method="continuous"`, no cutoff is applied to the cross-validated scores. Instead, a PH model is developed that includes terms for the main effect of *Treatment*, main effect of cross-validated score and interaction effect of *Treatment* by cross-validated score. From this model, two plots can be generated. The first plot (obtained by specifying `plot.score=TRUE` in `eval.pact.cv`) consists of KM curves by *Treatment* for the 20th, 40th, 60th and 80th percentiles of the cross-validated predictive scores and depicts the differential effect of treatment as function of increasing cross-validated scores (Figure 2). The second plot that can be generated is the plot of the probability of survival beyond a (user specified) landmark time as a function of the cross-validated score and *Treatment* (obtained by specifying the landmark time for `plot.time` in `eval.pact.cv`) (Figure 3).

```
Call:  eval.pact.cv(out.cv = cv1, method = "cont", plot.score = TRUE,      perm.test = FALSE)



family:  cox


Coefficients from the regression model with Treatment, cross-validated score
and Treatment*score interaction
          T1     predscore T1:predscore
     -0.4342       -0.1188       0.2307
```

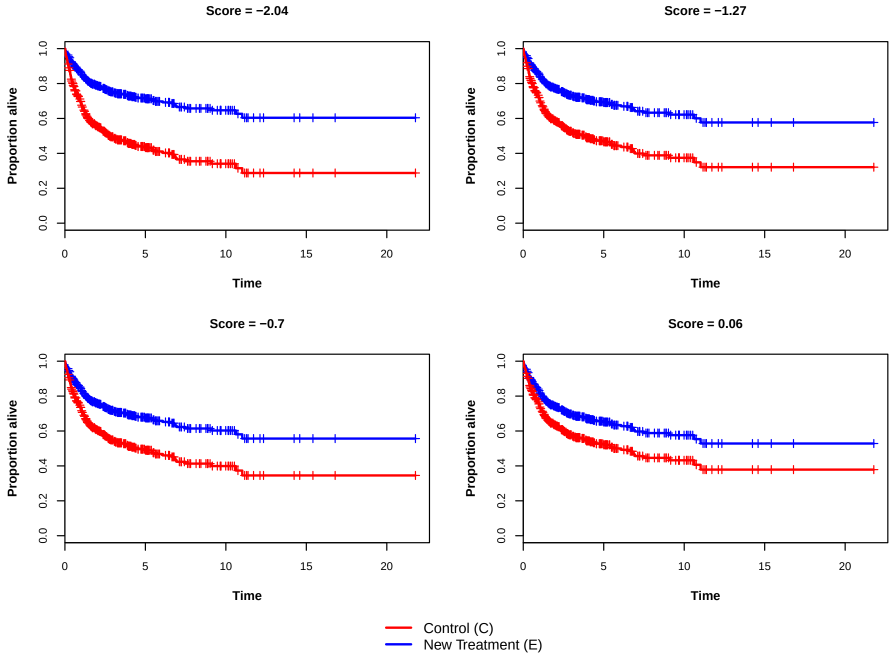

Figure 2: Figures after evaluation with `method="continuous"` (type 1, plots at specified percentiles of cross-validated scores)

```
> ### Evaluation with method="continuous". No cut-offs here.
> ### Plot type 1: KM curves are plotted at the 20th, 40th, 60th and 80th
> ### percenttiles of the cross-validated treatment scores (Figure 2)
> e21 <- eval.pact.cv(cv1, method="cont", plot.score=TRUE, perm.test=FALSE)
```

```
> ### Evaluate with method="continuous". Plot type 2: Prob[surv] beyond user
> ### specified landmark time as a function of the predictive score (Figure 3)
> e22 <- eval.pact.cv(cv1, method="cont", plot.score=FALSE, plot.time=12)
```

```
Call:  eval.pact.cv(out.cv = cv1, method = "cont", plot.score = FALSE,      plot.time = 12)


family:  cox

Coefficients from the regression model with Treatment, cross-validated score
and Treatment*score interaction
          T1    predscore T1:predscore
     -0.4342      -0.1188       0.2307
```

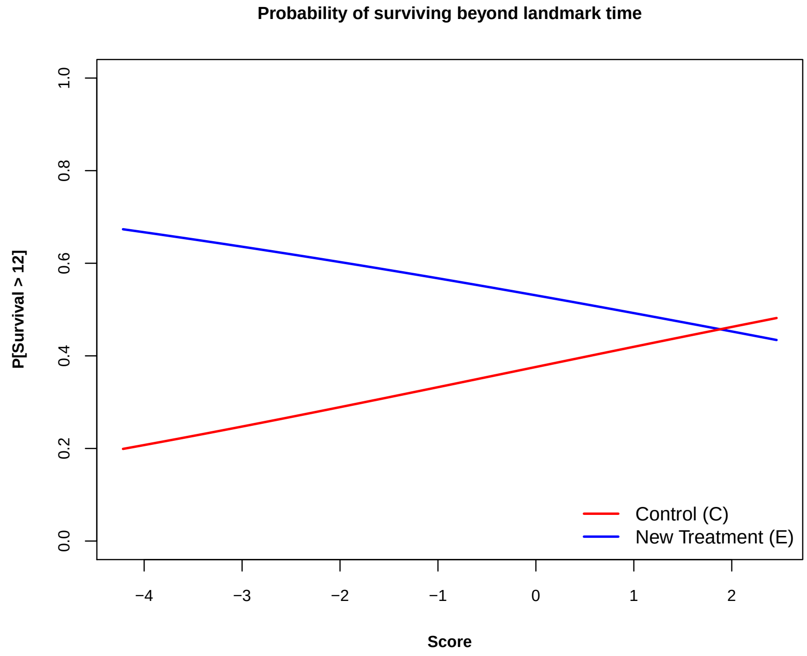

Figure 3: Figures after evaluation with `method="continuous"` (type 2, probability of survival beyond landmark time)

3.3 Specifying fixed prognostic covariates, i.e., x_f

The user can specify a small set of prognostic covariates to always remain in the model using option Xf. This would result in a predictive model that is adjusted for these prognostic covariates. We illustrate this application of pact using the prostate cancer dataset.

```
> data("prostateCancer")
> head(prostateCancer)
```

```
  ID Treatment time status age               pf sz sg        ap
1  1         0   72      0  75  Normal.Activity  2  8 0.2999878
2  3         1   40      1  69  Normal.Activity  3  9 0.2999878
3  4         0   20      1  75 Limited.Activity  4  8 0.8999023
4  5         0   65      0  67  Normal.Activity 34  8 0.5000000
5  6         0   24      1  71  Normal.Activity 10 11 0.5999756
6  7         0   46      1  75  Normal.Activity 13  9 0.7999268
```

Then specify the treatment, response and covariates.

```
> Y <- prostateCancer[,3:4]  ## Survival response
> Xf <- prostateCancer[,7:8]  ## Prognostic covariates always in the model
> Xv <- prostateCancer[,c(5:6,9)]  ## Covariates for the predictive score
> Treatment <- as.factor(prostateCancer[,2])
```

Then fit the model.

```
> ### Fit predictive model, variable selection with "univar"
> p11 <- pact.fit(Y=Y, Xf=Xf, Xv=Xv, Treatment=Treatment, family="cox",
                 varSelect="univar")
```

```
Variable selection: Univariate method...
```

```
> ### And display it
> summary(p11)
```

```
                  T1                  age    pfNormal.Activity
        -3.176178312          0.002025321         -0.303748037
                  ap               T1.age T1.pfNormal.Activity
         0.002516760          0.050223885         -0.650114371
               T1.ap                   sz                   sg
        -0.003785730          0.014738644          0.075746535
```

```
> ### Print
> print(p11)
```

```
Call:  pact.fit(Y = Y, Xf = Xf, Xv = Xv, Treatment = Treatment, family = "cox",      varSelect = '



family:  cox



Classification function for classifying future subjects:
 f =  ( -3.1762 )
  +  (0.0502 )*age
  +  (-0.6501 )*pfNormal.Activity
  +  (-0.0038 )*ap
```

```
> ### Model can be used to predict score for new subjects
> ### We only need to specify variables in Xv for new subjects
>
> newXv <- data.frame(age=c(60,70),
                    pf=c("Normal.Activity","Limited.Activity"),
                    ap=c(0.5,0.5))
> predict(p11, newXv)
```

```
         1          2
-0.8147525  0.3376008
```

3.4 Binary Response

The application of pact for a data with binary response variable is illustrated with the EORTC10994 data set. This dataset contains treatment, response and covariate information for 125 subjects with breast cancer. The covariate dimension is low, as there are only 4 covariates. The binary response Y, the predictor Xv and the $Treatment$ variables are first defined. Xf is not present (equals NULL).

```
> data("EORTC10994")
> head(EORTC10994, n=4)
```

```
  ID Age Treatment Response TumorSize Node ERBB2Log2
1  2  54         0        0     Small   No      6.75
2  5  47         0        0     Small  Yes      6.46
3  7  54         0        0     Large   No      8.30
4 10  55         0        0     Small   No      6.62
```

```
> Y <- EORTC10994[,4]   ## Response
> Xv <- EORTC10994[,c(2,5,6,7)]  ## Variables in Xv
> Treatment <- as.factor(EORTC10994[,3])   ## Treatment
```

For fitting the predictive model for a binary response, the option is `family="binomial"` in `pact.fit`. Cross-validated predictive scores can be obtained using `pact.cv` and evaluation statistics can be obtained through `eval.pact.cv`. With `method="discrete"` option in `eval.pact.cv`, the cross-validated estimates of response rates with E and C are displayed for the subset predicted to 'benefit', as well as the subset predicted 'no benefit' from E. If `method="continuous"` is chosen in `eval.pact.cv`, a logistic regression model is developed that includes the main effect of *Treatment*, main effect of cross-validated score and interaction effect of *Treatment* by cross-validated score. From this model, a graph is produced depicting the probability of response as a function of the cross-validated score and *Treatment* (Figure 4).

```
> ### Fit predictive model, no variable selection
> pbin <- pact.fit(Y=Y, Xv=Xv, Treatment=Treatment, family="binomial",
                varSelect="none")
```

```
No variable selection: All variables in X used in the model...
```

```
> ### Evaluate the model using K-fold CV and method="discrete"
> cvbin <- pact.cv(pbin, nfold=5)
> e3 <- eval.pact.cv(cvbin, method="discrete", g=log(1), perm.test=FALSE)
> e3
```

```
Call:  eval.pact.cv(out.cv = cvbin, method = "discrete", g = log(1),      perm.test = FALSE)



family:  binomial


Response rate (RR) with E in group predicted to benefit from E: 0.4375
Response rate (RR) with C in group predicted to benefit from E: 0.5152
```

```
Call:  eval.pact.cv(out.cv = cvbin, method = "continuous", perm.test = FALSE)



family:  binomial


Coefficients from the regression model with Treatment, cross-validated score
and Treatment*score interaction
 (Intercept)          T1    predscore T1:predscore
     -0.2927      0.1287       0.3225      -0.2751
```

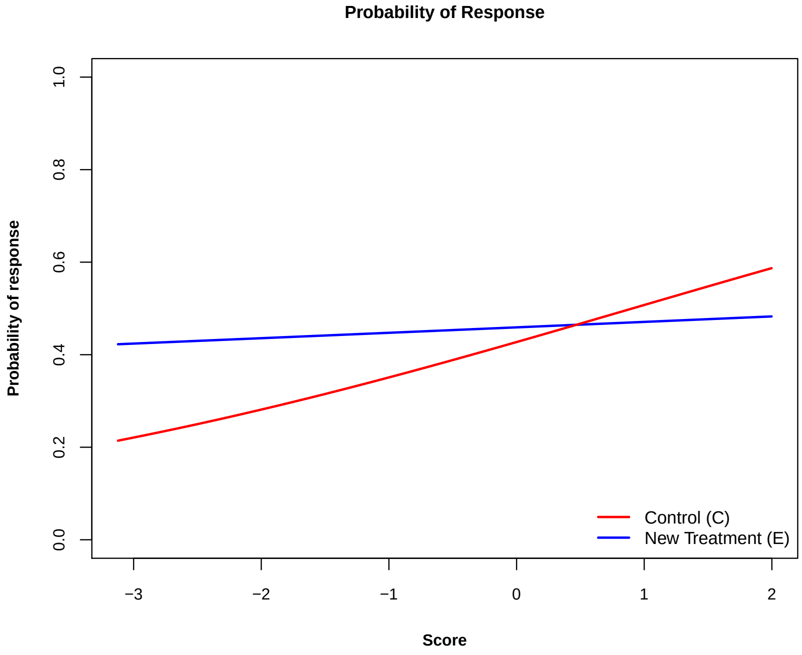

Figure 4: Figure after evaluation with `method="continuous"`: probability of response as a function of predictive score

```
Response rate (RR) with E in group predicted no benefit from E: 0.4815
Response rate (RR) with C in group predicted no benefit from E: 0.3333
```

```
> ### Evaluation for binary response with method="continuous".
> ### Plot: Probability of response as a function of cross-validated
> ### predictive score (Figure 4)
> e4 <- eval.pact.cv(cvbin, method="continuous", perm.test=FALSE)
```

3.5 Permutation Tests for Treatment Effects

Permutation based testing for statistical significance of interaction effects of cross-validated scores and *Treatment* can be carried out by specifying `perm.test=TRUE` in `eval.pact.cv`. The number of permutations can be set using the `nperm` option. At least 500 to 1000 permutations are recommended.

3.5.1 Evaluation Method: Discrete

In the case of a survival response, permutation based p-values for differential treatment effects are computed separately for the subset predicted to 'benefit' as well as for the subset predicted 'no benefit' from E. The statistic used is the log rank statistic.

In the case of a binary response, permutation based p-values are computed for testing the null hypothesis that response rates are the same with treatments E and C. A chi-square test statistic is used. Permutation p-values are computed for subsets predicted to 'benefit' as well as predicted 'no benefit' from E.

3.5.2 Evaluation Method: Continuous

If `method="continuous"` is chosen in `eval.pact.cv`, a permutation based test is performed to test the null hypothesis that the interaction coefficient of *Treatment* and cross-validated score is zero in the regression model that was developed using main effect of *Treatment*, main effect of cross-validated score and interaction effect of *Treatment* and cross-validated score.

```
> ### Permutation test examples (survival response): method="discrete"
> e5 <- eval.pact.cv(cvbin, method="discrete", g=log(1),
           perm.test=TRUE, nperm=100)
```

```
Start Permutations..May take a few minutes to complete..


End Permutations
```

```
> e5  ### (or print(e5))
```

```
Call:  eval.pact.cv(out.cv = cvbin, method = "discrete", g = log(1),      perm.test = TRUE, nperm
family:  binomial

Response rate (RR) with E in group predicted to benefit from E: 0.4375
Response rate (RR) with C in group predicted to benefit from E: 0.5152
Response rate (RR) with E in group predicted no benefit from E: 0.4815
Response rate (RR) with C in group predicted no benefit from E: 0.3333

p-value for the difference in RR (E vs C) in group predicted to benefit from E
based on 100 permutations: 0.505

p-value for the difference in RR (E vs C) in group predicted to not benefit from E
based on 100 permutations: 0.2475
```

```
> ### Permutation test examples (survival response): method="continuous"
> e6 <- eval.pact.cv(cvbin, method="continuous", perm.test=TRUE, nperm=100)
```

```
Start Permutations..May take a few minutes to complete..

End Permutations
```

```
> e6  ### (or print(e6))
```

```
Call:  eval.pact.cv(out.cv = cvbin, method = "continuous", perm.test = TRUE,      nperm = 100)



family:  binomial

Coefficients from the regression model with Treatment, cross-validated score
and Treatment*score interaction
 (Intercept)           T1    predscore T1:predscore
     -0.2927       0.1287       0.3225      -0.2751

Two-sided p-value for the Treatment*score interaction coefficient
based on 100 permutations: 0.6634

One-sided p-value for the Treatment*score interaction coefficient
based on 100 permutations: 0.495
```

References

[1] Simon R (2012), *Clinical Trials for Predictive Medicine.* Statistics in Medicine, 31(25): 3031-40.

[2] Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). *Regularization Paths for Generalized Linear Models via Coordinate Descent.* Journal of Statistical Software, 33(1): 1-22. URL http://www.jstatsoft.org/v33/i01/.

[3] Noah Simon, Jerome Friedman, Trevor Hastie, Rob Tibshirani (2011). *Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent.* Journal of Statistical Software, 39(5), 1-13. URL http://www.jstatsoft.org/v39/i05/.