Vignette for the `anoint` package

Stephanie Kovalchik and Ravi Varadhan

April 20, 2012

This document gives a brief introduction to the main functions of the `anoint` package. The package provides a set of tools for the "analysis of interactions" to investigate consistency of treatment effect with data from a parallel-group clinical trial.

The examples will make use of a simulated clinical trial data set, motivated by the structure of the Studies of Left Ventricular Dysfunction Trial (SOLVD-T), a placebo-controlled trial of the angiotensin-converting-enzyme inhibitor enalapril for patients with congestive heart failure (Yusuf 1991). The simulated data set is included with the package.

Creating a `anoint` object

Specifying the prognostic factors

The fundamental object of the package is a `anoint` object. This is what specifies family of model to be fit in all regression methods, the candidate prognostic factors for investigating effect modification, and whether a selection procedure is used for identifying prognostic factors.

In the following, we create a `anoint` object for the mock SOLVD-T data, specifying age, pulse, lower ejection fraction, cardiothoracic ratio, and sodium as prognostic factors. The regression model for all the `anoint` regression methods will be a logistic model on the outcome of time to hospitalization or death within two years of study entry.

```
> library(anoint)
> data(simsolvd)
> simsolvd$event <- 1-simsolvd$censor
> obj <- anoint(event~(age+beat+lvef+cardratio+sodium)*trt,
+                   family = "binomial", data = simsolvd)
> obj
```

```
event ~ (age + beat + lvef + cardratio + sodium) * trt
```

The option family can be any one of the canonical models in the GLM family or a Cox regression model which is specified by setting `family` to 'coxph'.

Model-based prognostic selection

If we were uncertain whether all of the factors were prognostic, we could perform a prognostic selection step prior to any interaction testing. When the `anoint` option `select` is set to `TRUE`, a lasso variable selection is performed on the control data in order to identify factors that have a marginal influence on the outcome. The default settings use a ten-fold cross-validation to estimate the penalty parameter λ.

For example, suppose we were uncertain about the importance of the variables `noise` and `sodium`. We could require all variables known to be prognostic to be selected by setting their penalty factor in the lasso `glmnet` call to zero.

```
> obj <- anoint(event~(age+beat+lvef+cardratio+sodium+noise)*trt,
+                   family = "binomial", data = simsolvd, select=TRUE,
+                   penalty.factor=c(rep(0,5),1,1))

Performing selection procedure for prognostic model...
Selected MIM:

event ~ (age + beat + lvef + cardratio + sodium + noise) * trt

> obj

event ~ (age + beat + lvef + cardratio + sodium + noise) * trt
```

Note that the first term of the `penalty.factor` vector is always an intercept for GLM and Cox models. In this case, `noise` was dropped from the set of prognostic factors based on the empirical evidence in the control group.

PR-plot

Before formal testing of multiple interactions, we might be interested to see whether a proportional interaction model is reasonable. The proportional interaction model (PIM) postulates a constant effect modification on a score of the prognostic treatment-response factors. The score is linear for the specified regression model. To inspect proportional interactions, we use the PR-plot, for 'proportional response'. This bins the outcomes of the logistic model based on the baseline prognostic score then computes the treatment effect within the decile groups. If the treatment effects are linear with respect to the prognostic score, this would be evidence of proportional interactions.

To generate the PR-plot, apply the `plot` method of the `anoint` class.

```
> plot(obj, ylab="treatment effect (odds ratio)")
```

The shaded region of Figure 1 denotes the 95% confidence region for the overall treatment effect. Thus, the plot can give an idea about heterogeneity as a function of baseline prognosis, as determined by the candidate prognostic treatment-response factors.

Forest plot

We can also create a more traditional plot of subgroup effects using the function `forest`. All of the factors included must be categorical, so we first create categorical representations of the continuous variables. Because all of the continuous variables have been centered and scaled, a value greater than zero corresponds to an individual that has a greater than average value for the specified variable.

```
> simsolvd$age.cat <- factor(simsolvd$age>0)
> simsolvd$beat.cat <- factor(simsolvd$beat>0)
> simsolvd$lvef.cat <- factor(simsolvd$lvef>0)
> simsolvd$sodium.cat <- factor(simsolvd$sodium>0)
> obj <- anoint(event~(age.cat+beat.cat+lvef.cat+cardratio+sodium.cat)*trt,
+                   family = "binomial", data = simsolvd)
> forest(obj)
```

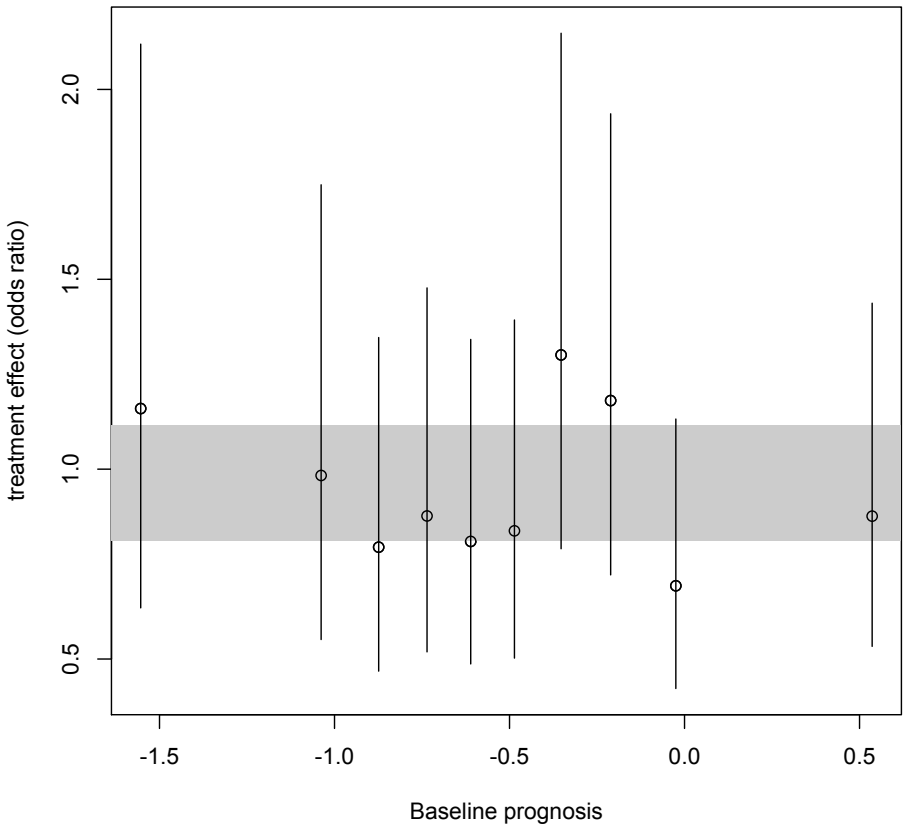

Figure 1: PR-plot

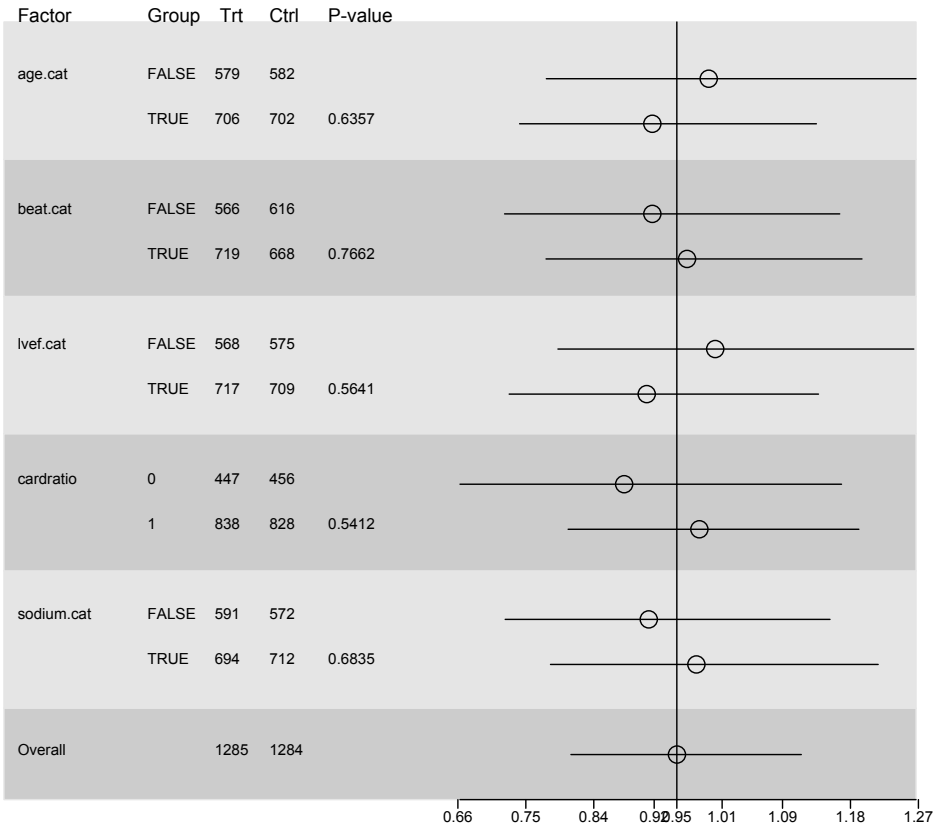

Figure 2: Forest plot of `anoint` object.

The result is a forest plot that shows the treatment effect estimate within each of the subgroups defined by the set of candidate prognostic treatment-response factors (Figure 2). The default settings also include the number of treated and control subjects in each subgroup and the p-value for the univariate test of interaction, based on the likelihood ratio statistic. Additional arguments can be supplied to modify the plot labeling and the number of terms included.

Global tests of effect modification

Several tests for the presence of effect modification among the candidate treatment-response factors are provided by the `anoint` package. These include tests based on one-by-one (OBO), unrestricted interactions (UIM), and proportional interactions (PIM) models of treatment-covariate interaction. OBO tests each prognostic factor one-at-a-time in a model of treatment, prognostic factor, and interaction effects. Thus, each model in the OBO is univariate with respect to treatment-covariate interaction. The UIM is a full regression model with all main effects and pairwise treatment-covariate interactions. The PIM is a proportional interactions model and is fit with either an exact or approximate method.

Likelihood ratio tests for any interaction (any proportional interaction in the case of PIM) is provided with the `anoint.fit` function. This includes one-stage tests for each method and two-stage tests which have the exact PIM global test at the first stage and, if no evidence of proportional interactions is found, OBO or UIM at the second stage. For the two-stage testing, each stage uses a one-half of the global α, which is specified by the `level` option.

In what follows, we obtain an `anoint.fit` object at the 5% global level of significance. The `summary` method gives a matrix with logical indicators for whether a given method rejected the null hypothesis of no interaction. Note that for PIM, the null is no *proportional* interaction. Thus, if the null is not rejected with the PIM test, it does not exclude the possibility that non-proportional interaction is present.

```
> obj <- anoint(event~(age+beat+lvef+cardratio+sodium)*trt,
+                  family = "binomial", data = simsolvd)
> fit <- anoint.fit(obj, level=.05)
> summary(fit)

               Global null rejected
OBO                           FALSE
OBO (adj.)                    FALSE
UIM                           FALSE
PIM (exact)                   FALSE
PIM (approx)                  FALSE
PIM/OBO (adj.)                FALSE
PIM/UIM                       FALSE

             Global LRT (p-value)
OBO (max)                0.2623014
UIM                      0.5242248
PIM (exact)              0.6308764
PIM (approx)             0.6344116
```

The `summary` method also reports each likelihood-ratio test p-value. The global test is based on the maximum LRT among the univariate tests for OBO. When corrected for multiplicity using a Bonferroni correction, this is compared to the $\chi^2(1)$ at level α/K, where K is the number of univariate tests. The p-value for UIM is for the LRT comparing a model with all pairwise treatment-covariate interactions, to a model with only main effects. The PIM models are one degree of freedom tests of the responsiveness parameter θ.

Model fits

Any of the `anoint` models can be extracted using the function `fits` and indicating the `type` (any one or more of 'obo', 'uim', 'pim.exact', 'pim.approx').

```
> pim.fit <- fits(fit, type = "pim.exact")
> pim.fit

$pim.exact

Baseline/treatment effects:

             Estimate
Control   -0.76750591
```

```
Treatment -0.05183811

Prognostic effects:

             Estimate
age        0.23326849
beat       0.07101775
lvef      -0.32476581
cardratio  0.40897389
sodium    -0.19560137

Responsiveness parameter:

       Estimate       LRT   p-value
theta 0.9120764 0.2308742 0.6308764
```

This is a list with an object of class 'pim'. Each type of model has the standard set of methods including: `print`, `summary`, `coef`, `vcov`, `predict`, and `confint`. For example, if we were interested in the confidence intervals for the intercept in the control group and for the `sodium` term of the exact PIM, we could obtain them with the `confint` method.

```
> confint(pim.fit$pim.exact,parm=c("Control","sodium"))

             2.5 %     97.5 %
Control -0.5984761 -0.9365358
age      0.3219879  0.1445491
```

In the above, the default variance-covariance for the model terms regards the responsiveness parameter as fixed. This could underestimate the standard error. As an alternative, the variance-covariance can be estimated using bootstrap resampling with the `n.boot` option set to some positive integer in the `pim` call. An example of this is provided in the next section.

Direct fitting with `anoint` object

We could also obtain any fit directly from the `anoint.object`. For example, to fit the one-by-one regression models:

```
> obo.fit <- obo(obj) # A list of fits
> obo.fit$fit

$`event ~ age *  trt`

Call:  glm(formula = f, family = anoint@formula@family, data = anoint@data)

Coefficients:
(Intercept)          age          trt      age:trt
   -0.49402      0.18015     -0.04885     -0.03173

Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
Null Deviance:            3397
Residual Deviance: 3380         AIC: 3388
```

```
$`event ~ beat *  trt`

Call:  glm(formula = f, family = anoint@formula@family, data = anoint@data)

Coefficients:
(Intercept)         beat          trt     beat:trt
   -0.48795      0.05157     -0.05906      0.08247

Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
Null Deviance:            3397
Residual Deviance: 3390         AIC: 3398

$`event ~ lvef *  trt`

Call:  glm(formula = f, family = anoint@formula@family, data = anoint@data)

Coefficients:
(Intercept)         lvef          trt     lvef:trt
   -0.49932     -0.31053     -0.05289      0.01288

Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
Null Deviance:            3397
Residual Deviance: 3341         AIC: 3349

$`event ~ cardratio *  trt`

Call:  glm(formula = f, family = anoint@formula@family, data = anoint@data)

Coefficients:
  (Intercept)      cardratio            trt  cardratio:trt
      -0.7530         0.4016        -0.1276         0.1072

Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
Null Deviance:            3397
Residual Deviance: 3369         AIC: 3377

$`event ~ sodium *  trt`

Call:  glm(formula = f, family = anoint@formula@family, data = anoint@data)

Coefficients:
(Intercept)       sodium          trt   sodium:trt
   -0.48368     -0.21537     -0.06489      0.09205

Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
Null Deviance:            3397
Residual Deviance: 3378         AIC: 3386
```

The `obo` function, like all of the `anoint` models, returns a list with the fitted model `fit`, the likelihood ratio test statistics for each univariate test of interaction, and the corresponding unadjusted pvalue.

Here is an example of fitting the UIM.

```
> uim.fit <- uim(obj)
> summary(uim.fit$fit)

Call:
glm(formula = object@formula@formula, family = object@formula@family,
    data = object@data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5610  -0.9729  -0.7839   1.2652   1.9456

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.71208    0.10266  -6.936 4.03e-12 ***
age            0.24240    0.06211   3.903 9.52e-05 ***
beat           0.03021    0.05982   0.505   0.6135
lvef          -0.33347    0.06039  -5.521 3.36e-08 ***
cardratio      0.32495    0.12647   2.569   0.0102 *
sodium        -0.25026    0.06016  -4.160 3.19e-05 ***
trt           -0.16963    0.14741  -1.151   0.2498
age:trt       -0.03458    0.08663  -0.399   0.6897
beat:trt       0.08278    0.08540   0.969   0.3324
lvef:trt       0.04132    0.08597   0.481   0.6307
cardratio:trt  0.14342    0.17971   0.798   0.4248
sodium:trt     0.13120    0.08386   1.565   0.1177
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3396.7  on 2568  degrees of freedom
Residual deviance: 3270.1  on 2557  degrees of freedom
AIC: 3294.1

Number of Fisher Scoring iterations: 4

> uim.fit$LRT

[1] 4.176995

> uim.fit$pvalue

[1] 0.5242248
```

In the following, we fit an exact PIM and specify that a bootstrap estimate of the variance-covariance for all model terms be taken using 100 resamples.

```
> pim.fit <- pim(obj, n.boot = 100)
> summary(pim.fit)
```

```
             Estimate         SE    t value      p-value
Control   -0.76750618 0.09455315 -8.1171931 7.328784e-16
Treatment -0.05183732 0.10080200 -0.5142489 6.071223e-01
age        0.23326879 0.04992098  4.6727611 3.125548e-06
beat       0.07101779 0.03808366  1.8647836 6.232604e-02
lvef      -0.32476621 0.05568035 -5.8326895 6.140323e-09
cardratio  0.40897424 0.09874797  4.1415964 3.560641e-05
sodium    -0.19560171 0.05673503 -3.4476358 5.746412e-04
theta      0.91207398 0.18336794  4.9740102 6.992362e-07
```

References

Follmann DA, Proschan MA. A multivariate test of interaction for use in clinical trials. *Biometrics* 1999; 55(4):1151-1155

Yusuf, S. et al. Effect of Enalapril on Survival in Patients with Reduced Left-Ventricular Ejection Fractions and Congestive-Heart-Failure. *NEJM* 1991; 325:293-302.