Published in final edited form as:

Biometrics. 2014 September ; 70(3): 713–716. doi:10.1111/biom.12189.

Discussion of Combining Biomarkers to Optimize Patient Treatment Recommendations

Ying-Qi Zhao^{1,*} and Michael R. Kosorok^{2,**}

¹Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, Madison, WI, 53792, U.S.A

²Department of Biostatistics, Department of Statistics and Operations Research University of North Carolina at Chapel Hill, Chapel Hill, NC, 27599, U.S.A

1. Introduction

The problem of combining markers to optimize treatment selection has recently received significant attention among statistical researchers. We congratulate Kang, Jane, and Huang (2014; hereafter KJH) on an elegant contribution to this important area. KJH provide a novel application of boosting to find the best marker combination. In particular, with a binary outcome, the optimal treatment rule is determined by the conditional probability of having disease, i.e., the risk, given the covariates and the treatment. If the risk models are correctly specified, the optimal rule can be deduced accordingly. While a generalized linear model is a simple and popular option, it may suffer from model misspecification. The proposed method in KJH achieves a measure of robustness to such model misspecification through the use of boosting, combined with iteratively reweighting each subject's potential misclassification based on treatment benefit in the previous iteration.

Although KJH indicate that the purpose of the proposed method is to classify subjects according to the unobserved optimal treatment decision rule, the approach does not utilize a clear objective function for optimization. Risk modeling is required to estimate the optimal rules and to further update the weights at each step. As shown in the simulation results, the performances vary with different working models. An alternative approach is given in Zhao et al. (2012) who propose outcome weighted learning (OWL) which estimates the optimal treatment decision rule through a weighted classification procedure that incorporates outcome information. To be more specific, the optimization target, which directly leads to the optimal treatment decision rule, can be viewed as a weighted classification error where each subject is weighted proportional to his or her clinical outcome. In the next section, we will briefly introduce the idea of OWL and modify it to the binary outcome setup. In Section 3, we present simulation studies comparing OWL with the boosting method proposed by KJH. We conclude with a brief discussion in Section 4.

*yqzhao@biostat.wisc.edu. **kosorok@unc.edu.

2. Outcome Weighted Learning (OWL)

Using the same notation as KJH, we let $D \in \{0, 1\}$ be the binary indicator of an adverse outcome, T indicate treatment ($T = 1$) or not ($T = 0$), and Y be the marker which can be used to identify a subgroup. We assume that the data $\{D_i, T_i, Y_i\}_{i=1}^n$ are from a randomized clinical trial. For arbitrary treatment rule $g: Y \mapsto \{0, 1\}$, the expected benefit under g , i.e., if g were implemented in the whole population, can be written as (Qian and Murphy, 2011)

$$\mathbb{P} \left\{ \frac{1(D=0)I(T=g(Y))}{\pi_c(Y)} \right\}, \quad (2.1)$$

where $\pi_c(Y)$ is the known probability of treatment, and the optimal treatment rule $g^{opt}(Y)$ can be obtained by maximizing the above quantity. Equivalently, by minimizing

$$\mathbb{P} \left\{ \frac{I(D=0)I(T \neq g(Y))}{\pi_c(Y)} \right\},$$

we can obtain that $g^{opt}(Y) = 1\{ (Y) \leq 0 \}$, where $(Y) = P(D = 1|T = 0, Y) - P(D = 1|T = 1, Y)$. Indeed, this can be viewed as a weighted classification error in the setting where we wish to classify T in the responders group using the covariate Y , with $1/\pi_c(Y)$ as the weights. In particular, when $\pi_c(Y) = 0.5$, the problem falls into a regular classification framework if we only consider responders. In this case, we classify subjects according to their assigned treatments only among those who received benefit from the assignment (i.e., the responders).

Provided with the data, we could potentially minimize the empirical analog

$$\min_g \mathbb{P}_n \left\{ \frac{I(D=0)I(T \neq g(Y))}{\pi_c(Y)} \right\} \quad (2.2)$$

to estimate $g^{opt}(Y)$, where $\mathbb{P}_n$ denotes the empirical average. Note that this is similar to the quantity IPWE(η) presented in Zhang et al. (2012). Due to the nonconvexity and discontinuity of the 0–1 loss, it is computationally difficult to minimize (2.2). We address this problem by using a convex surrogate loss function to replace the 0–1 loss, a common practice in the field of machine learning literature (Zhang, 2004; Bartlett et al., 2006). In other words, instead of minimizing (2.2), we minimize

$$\min_{f \in \mathcal{F}} \mathbb{P}_n \left\{ \frac{I(D=0)\phi\{(2T-1)f(Y)\}}{\pi_c(Y)} \right\} + \lambda_n \|f\|^2, \quad (2.3)$$

where $\mathcal{F}$ is the functional space that f resides in, $2T - 1$ is used to rescale T to reside in $\{-1, 1\}$, $\phi(t)$ is a convex surrogate loss function, and $\lambda_n \|f\|^2$ is a regularization term to avoid overfitting, with $\|\cdot\|$ denoting the norm of f in $\mathcal{F}$ and λ_n controlling the amount of penalization. The estimated treatment rule is $\hat{g}(Y) = I(\hat{f}(Y) > 0)$, where $\hat{f}$ is the solution to (2.3). We can specify $\mathcal{F}$ to be a linear functional space if we are only interested in linear decision rules. We can also consider nonlinear functional spaces where treatment effects can potentially be complex and nonlinear. In the simulation section, we will examine the

performances using two popular choices for $\varphi(t)$, including the hinge loss $\varphi(t) = \max(1 - t, 0)$ and the exponential loss $\varphi(t) = \exp(-t)$.

Since being at high risk does not necessarily imply a larger benefit from treatment, we aim to find methods that are optimized for treatment selection (Kang et al., 2014). We point out that OWL directly targets the optimal decision rule, hence the covariate-treatment interaction effects are separated from the main effects. Indeed, it does not involve a modeling step for the risk and $\pi_c(Y)$ as required by KJH. If the functional space $\mathcal{F}$ is correctly specified for the interaction effects, we can consistently estimate $g^{opt}(Y)$ (Zhao et al., 2012). Specifically, using the exponential loss, we generalize Adaboost (Freund and Schapire, 1997; Friedman et al., 2000) to select the optimal treatment. Rather than using the weight function $W\{\pi_c(Y)\}$ to be updated at each step, we use instead $I(D_i = 0) \exp(-T_i f(Y_i))$ as the weight for each observation, where the $f(Y_i)$ are repeatedly updated in each iteration.

As a side note, the OWL method can be naturally generalized to continuous outcomes, which commonly occurs in practice. For example, if we let R denote the continuous outcomes, with larger values being more preferable, we only need to change $I(D = 0)$ to R in (2.1). The subsequent derivation and computation follow accordingly.

3. Simulation studies

We compare the OWL method with logistic regression and with the boosting methods (both linear logistic boosting and classification tree boosting) proposed by KJH. The OWL methods are implemented using the hinge loss and the exponential loss as the convex surrogates. We use the same simulation scenarios as presented in KJH. Since patients are equally randomized to $T = 0$ or 1 , $\pi_c(Y)$ can be dropped from the optimization objective (2.3). Thus, we can apply the standard classification algorithm to the simulated data by only considering the responders with $D = 0$. The adaboost (Freund and Schapire, 1997) or support vector machine (SVM) (Cortes and Vapnik, 1995) can be carried out for this subset of patients, by treating their assignments $T \in \{0, 1\}$ as the class labels and the biomarkers Y as the predictors. The adaboost is implemented by the R function **ada** (R package *ada* (Culp et al., 2006)) using the default settings with exponential loss function. The SVM is implemented by the R function **svm** (R package *e1071* (Dimitriadou et al., 2008)). Both linear and Gaussian kernels are used for comparison, yielding linear and nonlinear decision rules respectively.

For each scenario, 1000 data sets are generated as training data to build the treatment decision rule, $\hat{g}(Y)$. A large independent test data set with $n = 10^5$ observations is generated to evaluate the performance of the obtained $\hat{g}(Y)$ under different methods. Mean and Monte-Carlo standard deviation (SD) of $\theta\{\hat{g}(Y)\}$ and mean $MCR_{TB}\{\hat{g}(Y)\}$ are reported, where θ and MCR_{TB} are defined in KJH. The method is marked in **bold** if it outperforms other methods by producing the highest mean θ . Logistic regression performs the best when the model is correctly specified, which is anticipated and has been noted in KJH. The linear logistic boosting has a similar performance to logistic regression if the models are correct, and can improve on logistic regression to some extent when the models are misspecified. However, if the effects are nonlinear, classification tree boosting may be a better option. An

appropriate working model is important in practice, when the underlying truth is masked. The OWL method has comparable performances with linear treatment effects. When the outcome models are complex with nonlinear treatment interactions (Scenarios 5 & 6), the OWL methods lead to better performances. In general, OWL using hinge loss with Gaussian kernel has a favorable performance which is always close-to-optimal, especially when the sample size is large.

4. Discussion

In summary, the intuition underlying the proposed boosting method by classifying subjects according to their (unobserved) optimal treatment holds much promise. However, in practice, attention must be paid to the selection of the working models and the weight function w since they may impact the results to some extent. The OWL procedure discussed for comparison utilizes a machine learning approach to find the best treatment rule through optimizing a target function which directly reflects the overall benefit of the decision rule. When the outcome is binary, the method proposed in KJH can have a better performance with small sample sizes, given that the OWL essentially only uses information from the responders. Also, if there is prior information on the relationship between outcomes and candidate biomarkers, the KJH method can be preferable. On the other hand, due to its flexibility in handling covariate-treatment interaction effects, OWL can yield better results when the dimension of the covariate space is high or when the true model is fairly complex. For large samples sizes, OWL with hinge loss and Gaussian kernel performs nearly optimally in every scenario. Additionally, while OWL can also readily handle additional data types such as continuous outcomes, this is not yet the case for the KJH method and so it would be worthwhile to investigate the possibility of such a generalization.

References

- Bartlett PL, Jordan MI, McAuliffe JD. Convexity, classification, and risk bounds. *JASA*. 2006; 101:138–156.
- Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995:273–297.
- Culp M, Johnson K, Michailides G. ada: An R package for stochastic boosting. *Journal of Statistical Software*. 2006; 17:9.
- Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A. Misc functions of the department of statistics (e1071), tu wien. R package. 2008:1–5.
- Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*. 1997; 55:119–139.
- Friedman J, Hastie T, Tibshirani R. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*. 2000; 28:337–374.
- Kang C, Janes H, Huang Y. Combining biomarkers to optimize patient treatment recommendations. *Biometrics*. 2014 In press.
- Qian M, Murphy SA. Performance guarantees for individualized treatment rules. *Annals of Statistics*. 2011; 39:1180–1210. [PubMed: 21666835]
- Zhang B, Tsiatis AA, Laber EB, Davidian M. A robust method for estimating optimal treatment regimes. *Biometrics*. 2012; 68:1010–1018. [PubMed: 22550953]
- Zhang T. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*. 2004; 32:56–134.
- Zhao YQ, Zeng D, Rush AJ, Kosorok MR. Estimating individualized treatment rules using outcome weighted learning. *JASA*. 2012; 107:1106–1118. [PubMed: 23630406]

Table 1

Results of the simulation study with sample size $n = 500$. OWL-HL: outcome weighted learning with hinge loss, linear kernel; OWL-HG: outcome weighted learning with hinge loss, Gaussian kernel; OWL-E: outcome weighted learning with exponential loss; L-Boosting: linear logistic boosting; C-boosting: classification tree boosting; Logit: logistic regression.

SN	True θ		OWL-HL	OWL-HG	OWL-E	L-Boosting	C-Boosting	Logit
1	0.1268	Mean	0.1225	0.1057	0.1020	0.1240	0.1149	0.1256
		SD	0.0025	0.0038	0.0034	0.0018	0.0040	0.0019
	MCR_{TB}	Mean	0.0640	0.1363	0.1620	0.0517	0.1128	0.0309
2	0.1243	Mean	0.1224	0.1129	0.0946	0.1248	0.0921	0.1241
		SD	0.0024	0.0052	0.0024	0.0017	0.0037	0.0017
	MCR_{TB}	Mean	0.0664	0.1160	0.1883	0.0481	0.1811	0.0545
3	0.1341	Mean	0.1256	0.1191	0.1010	0.1315	0.1154	0.1315
		SD	0.0026	0.0045	0.0031	0.0015	0.0051	0.0016
	MCR_{TB}	Mean	0.0865	0.1062	0.1461	0.0498	0.1307	0.0499
4	0.0657	Mean	0.0654	0.0061	0.0436	0.0657	0.0356	0.0657
		SD	0.0058	0.0070	0.0031	0.0031	0.0031	0.0040
	MCR_{TB}	Mean	0.0855	0.4638	0.2721	0.0202	0.2853	0.0489
5	0.0950	Mean	0.0798	0.0646	0.0783	0.0594	0.0517	0.0592
		SD	0.0046	0.0034	0.0035	0.0030	0.0037	0.0030
	MCR_{TB}	Mean	0.2049	0.2250	0.1810	0.3110	0.2708	0.3070
6	0.1393	Mean	0.0001	0.0931	0.0903	0.0313	0.0926	-0.0023
		SD	0.0049	0.0035	0.0035	0.0032	0.0034	0.0055
	MCR_{TB}	Mean	0.4010	0.2264	0.2267	0.3469	0.2408	0.4296
7	0.1419	Mean	0.1295	0.1217	0.1140	0.1312	0.1185	0.1313
		SD	0.0028	0.0056	0.0038	0.0015	0.0044	0.0015
	MCR_{TB}	Mean	0.0718	0.1123	0.1455	0.0468	0.1448	0.0468

Table 2

Results of the simulation study with sample size $n = 5000$. OWL-HL: outcome weighted learning with hinge loss, linear kernel; OWL-HG: outcome weighted learning with hinge loss, Gaussian kernel; OWL-E: outcome weighted learning with exponential loss; L-Boosting: linear logistic boosting; C-boosting: classification tree boosting; Logit: logistic regression.

SN	True θ		OWL-HL	OWL-HG	OWL-E	L-Boosting	C-Boosting	Logit
1	0.1268	Mean	0.1228	0.1256	0.1218	0.1265	0.1177	0.1265
		SD	0.0025	0.0038	0.0034	0.0018	0.0040	0.0019
	MCR_{TB}	Mean	0.0776	0.0338	0.0709	0.0124	0.0961	0.0111
2	0.1243	Mean	0.1251	0.1246	0.1233	0.1264	0.1083	0.1264
		SD	0.0024	0.0052	0.0024	0.0017	0.0037	0.0017
	MCR_{TB}	Mean	0.0442	0.0459	0.0598	0.0162	0.1201	0.0181
3	0.1341	Mean	0.1319	0.1316	0.1284	0.1338	0.1204	0.1337
		SD	0.0026	0.0045	0.0031	0.0015	0.0051	0.0016
	MCR_{TB}	Mean	0.0408	0.0433	0.0628	0.0110	0.1095	0.0161
4	0.0657	Mean	0.0653	0.0634	0.0640	0.0652	0.0530	0.0652
		SD	0.0058	0.0070	0.0031	0.0031	0.0031	0.0040
	MCR_{TB}	Mean	0.0357	0.1515	0.1189	0.0090	0.2256	0.0035
5	0.0950	Mean	0.0792	0.0923	0.0893	0.0761	0.0750	0.0732
		SD	0.0046	0.0034	0.0035	0.0030	0.0037	0.0030
	MCR_{TB}	Mean	0.2165	0.0693	0.0878	0.2364	0.2215	0.2522
6	0.1393	Mean	0.0417	0.1315	0.1258	0.0457	0.1116	0.0123
		SD	0.0049	0.0035	0.0035	0.0032	0.0034	0.0055
	MCR_{TB}	Mean	0.3574	0.1081	0.1349	0.3490	0.2585	0.4039
7	0.1419	Mean	0.1294	0.1319	0.1288	0.1327	0.1218	0.1327
		SD	0.0028	0.0056	0.0038	0.0015	0.0044	0.0015
	MCR_{TB}	Mean	0.0928	0.0440	0.0789	0.0329	0.1094	0.0307