nov. 1 pepe 2013 seems to resolve the problem of testing the index aucs for significant difference. they show that $auc(x,y) = P(risk(X,Y|D=0) < risk(X,Y|D=1))$ is equal to $auc(x) = P(risk(X|D=0) < risk(X|D=1)$, where $risk(\cdot) = P(D=1 \mid \cdot)$, if and only if the risks are equal $risk(X,Y) = risk(X)$. suppose it is needed to test $P(\beta^T(x,y)|D=0 < \beta^T(x,y)|D=1) = P(\gamma^T x|D=0 < \gamma^T x|D=1)$, obtained by LDA, logistic regression etc. if there is some monotone link h such that $P(D=1|x,y) = h(\beta^T(x,y)), P(D=1|x) = h(\gamma^T(x))$, then the test is the same as $P(risk(x,y)|D=0 < risk(x,y)|D=1) = P(risk(x)|D=0 < risk(x)|D=1)$ so one may just test $risk(x,y) = risk(x)$. however, this requires knowing the true risk function. one can obtain the indices $beta^T$ and $gamma^T$ and compare the discrimination, and use the indices in practice, without knowing the correct risk function. (i think their result does imply you shouldnt run the two tests, one of association and one of discrimination–those are redundant.)

is it possible to come up with a concrete example, where the postulated risks are equal/unequal but that doesn't hold for the aucs of the indices?

i was confused at first if pepe had settled the issue (pace my issue with correct specification) then why did heller publish the 2017 paper? when i looked at it again, they only justification they offer is for forming CIs under the alternative case, when the aucs are in fact different. they dont give any reason to test the null that there is no difference. that is no different from pepe's paper, who say the auc test should be only for estimation, not testing the null. obfuscation? "To date, methodology to construct accurate confidence intervals for the difference in AUCs from nested models is incomplete. This work fills the gaps in the AUC methodology by developing a proper null asymptotic distribution for the difference in AUCs and an accurate confidence interval for the population difference when the new markers are associated with the binary outcome." the CI they form is based on the assumption that the new markers adds value: " An asymptotic 95% confidence interval, derived directly from Theorem 2...". when introducing the null chi square distribution, they state: "The derived distribution of the difference in the AUC statistic under this condition is useful for deriving a direct test of equality." but, unless worried about misspecification as in the above para, you could test the risk functions.

the small result #1 at the beginning of the pepe paper gives a much shorter proof of the jin lu 2009 result. that result is that if the data satisfy a binary response model with an index $P(D=1|x) = h(\beta_0^t x)$, then the auc $P(\beta^t x \mid D = 0 < \beta^t x \mid D = 1)$ is maximized over all β when $\beta = \beta_0$. this is the pepe result that rules to identify cases of the form $risk(x) > c$, where $c(\alpha)$ is chosen so that the rule has a fixed FPR $P(risk(x) > c(\alpha) \mid D = 0) = \alpha$, maximizes the power $P(risk(x) > c(\alpha) \mid D = 1)$ over any other function of x. Therefore the roc curve built on the marker $risk(x)$ is not less than the roc curve for any other marker based on x, and therefore the auc must be maximal. the jin proof also shows this result a fortiori, by first showing the roc is maximal at every point. but uses a much longer calculus based approach and some inequality from chebychev. and the pepe result is stronger: jin shows the auc is maximize over all other linear combinations, pepe shows it is over all other functions of x. moreover the pepe

result #1 is just the n-p lemma. testing $H_0 : D = 0$ vs $H_1 : D = 1$ using x, d. the most powerful test has the form $f(x, y|D = 1)/f(x, y|D = 0) > c$. but $f(x, y|D = 1)/f(x, y|D = 0) = risk(x, y)P(D = 1)/((1 - risk(x, y))P(D = 0))$ so the MP test is rejecting for large values of $risk(x, y)$.

nov. 28 Another error I was making that caused me time. Given a $\sqrt{n}$ estimator $\hat{\theta}$ and a candidate sd estimator $\hat{sd}(\hat{\theta})$, I was in the habit of looking at the rate of convergence to zero of the difference between the empirical sd of $\hat{\theta}$ and $\hat{sd}(\hat{\theta})$ and making sure that it was faster than $n^{-1/2}$. Since the asymptotic sd is the sd of $\sqrt{n}\hat{\theta}$, so $\sqrt{n}(\hat{sd}(\hat{\theta}) - sd(\hat{\theta})$ needs to be negligible. Especially a problem if $\hat{sd}(\hat{\theta})$ contains parameters that need to be estimated since these will converge at a $1/\sqrt{n}$ rate usually. But here I was actually working with influence functions. $\hat{\theta}$ is an IID sum, and the sd estimate is the empirical estimator. This is itself an estimate of $\sqrt{n}\hat{\theta}$, not $\hat{\theta}$. So there was no need to make sure it is $o(1/\sqrt{n})$, it just needs to be $o(1)$. And esetimated parameters are OK (take taylor expansion) as long as the parameter estimates are consistent.

nov. 30 To compute $E(\Phi(\beta_0 + \beta_1^T x + \beta_2^T y)|x)$ I was integrating with respect to the conditional distribution of $\beta_2^T y$ given $\beta_1^T x$. I should have been computing the conditional distirbution of y given x and from that obtaining the conditional distribution of $\beta_2^t y$. I wasted a lot of time on this issue and even posted to SO a question that never got answered. It made me suspect that probit regression with normal covariates in fact wasn't collapsible. I had just been using the wrong density.

in 12f, infl.probit- (beta.hat-beta) wasn't convering right. realized n was referenced in infl.probit and infl.hat.probit without beign defiend, was using global n. realized the problem because everything worked for the first n in ns.

in 12g. hajek for probit didnt seem to converge at a fast enough rate. nor did auc.probit. noticed the rate of convergence was good for small p, but deteriorated quickly as p increased: " convergence slows down as p increases. at least with these ns, $1/n \cdot 1$ for $p = 6$. ok for $p < 4$." realized i was using

```
obs <- auc.hat(x.0,x.1) - auc
```

instead of

```
obs <- auc.hat(x.0%*%beta,x.1%*%beta) - auc
```

in probit case, why isn't derivative term just 0 by pepe2013 argument? [[update: it is. i was computing the derivative incorrectly. there is a beta in the index, and a beta that is the parameter of the probit likelihood. i shouldve only been differentiating wrt the latter. fixed with updated auc.probit and auc.probit.deriv. but this means delong is sufficient for probit with normal covariates]]

dec. 10

The risk function for lda data (x, g) turns out to be a monotonic function of $x\beta$. so that is a reason delong is sufficient for that data. this explanation is more general than the reason offered by demler et al in 2012 regarding the

mahalonobis distance. also applies to probit with normal data, for exmaple. (also applies to other elliptic pdfs, as they conjecture?)

dec. 20

still trying to find a good exammple to test the proposed procedure. most have 0 derivative in both the full and reduced models so the delong statistic works. the probit and gaussian model is collapsible. tried lda with exponentially distributed x, but it turned out the risk here is also monotonically related to the index. trying logistic with normal data but here too the derivative looks very close to 0 even in the reduced model.

with a single predictor the derivative will often be zero, bc the risk (where the derivative of the auc is 0), $r(x)$, is often a monotonic function of x, and $P(\beta x_0 < \beta x_1) = P(x_0 < x_1) = P(r(x_0) < r(x_1))$ in this case. Demler 2017 uses a single covariate in the reduced model, so this example doesn't say much.

dec. 22

last few days have been thinking the riht way to decompose the problem is into the two aucs (full and reduced) that are being differenced. produce routines to reduce auc(beta.hat) to an iid sum, whatever the data is. can apply this separately to the full and reduced data sets, but also applies more generally to a comparison of any correlated aucs with linear parameters estimated from the data (eg lda and logistic). besides added generality, cleaner way to think about the problem and factor the code. instead of .full and .reduced data floatng around at the same time.