A General Approach to Variable Selection in Nonlinear Models

Carlos Carvalho (UTexas)
Richard Hahn, Rob McCulloch (Arizona State)

1. Model Selection
2. fit-the-fit Variable Selection
3. Cars Data
4. Hockey
5. Friedman
6. Utility Based (fit-the-fit) Variable Selection
7. Conclusion

1. Model Selection

Model selection, and more particularly *variable selection* has been, and continues to be, a major focus of research and practice.

A major selling point of the Lasso is that it gives you variable selection.

A major knock an "machine learning" techniques is that they are "black box".

But if you run (in R):

- deeplearning (in R package h2o)
- rpart (decision trees)
- gbm (boosting)
- randomForest

You can get measures of "variable importance".

Jerome H. Friedman, Multiple additive regression trees with application in epidemiology, STATISTICS IN MEDICINE Statist. Med. 2003;

For a single tree T, Breiman *et al.* [1] proposed a measure of (squared) relevance $I_j^2(T)$ for each predictor variable x_j, based on the number of times that variable was selected for splitting in the tree weighted by the squared improvement to the model as a result of each of those splits. This importance measure is easily generalized to additive tree expansions (3); it is simply averaged over the trees

$$I_j^2 = \frac{1}{M} \sum_{m=1}^{M} I_j^2(T_m) \tag{12}$$

Owing to the stabilizing effect of averaging, this measure (12) turns out to be more reliable than is its counterpart for a single tree. Since these measures are relative, it is customary to assign the largest a value of 100 and then scale the others accordingly.

But, they all have problems.

For example, it seems wrong to say how important a variable is.

Our view is that your focus should be on *subsets of variables* since in interesting cases the variables work together.

In general I like cross-validation to pick subsets and Bayesian posterior model probabilities.

But cross-validation is just a plug in approach and can be tricky in practice.

Witness the 1se rule!!!

Bayes model probabilities are highly sensitive to the prior which can be a real problem.

The popular BIC is a very crude approximation to a Bayesian model probability and how many people have any clue what AIC is?

I want something that:

- Not hard to do.
- Allows me to assess the *practical significance*.
- Allows me to assess the uncertainty.
- someone can use, *without understanding it* and get a reasonable answer.

And makes no assumptions
about the relationship between y and x !!!!!

Paper and R package at http://www.rob-mcculloch.org

In my mind I am following a decision theoretic approach with elements *prior*, *data model*, *utility*

$$p(\theta), \quad f(y \mid \theta), \quad U(y, a)$$

We then choose the action a to minimize our expected loss where the expectation is taken over the *predictive distribution* of Y given paste data.

$$\underset{a}{\text{minimize}} \;\; E(U(Y, a))$$

But, as you will see, we cut lots of conceptual corners.

In our case the action a will be the choice of a function $\gamma_S(x)$ such that

$$\gamma_S(x) \approx E(Y \mid x)$$

where $\gamma_S(x)$ only depends on the subset S of x variables and $E(Y \mid x)$ is the predictive expectation of $Y \mid x$.

Dennis Lindley, David Draper, Steve MacEachern, ...

In particular, our approach follows "Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective." (Hahn and Carvalho, JASA).
but we adapt the ideas to the non-linear case.

2. fit-the-fit Variable Selection

I'm going to use BART (Bayesian Additive Regression Trees) in all the examples.

BART is a pretty good "Machine Learner" and it give Bayesian posterior inference pretty robustly.

See, for example, the R package `BART`, Sparapani and McCulloch (2017) and check out the vignettes ($>$ browseVignettes()). Also see, `rbart` Pratola and McCulloch.

But you can apply the same approach to output from any other flexible fitter. You just may not have the last "uncertainty step".

Note:

Consider linear multiple regression.

Many Bayesian and frequentist approaches build a model around the idea that many of the coefficients are 0 and then attempt an inference.

Hahn and Carvalho build a model without assuming *any* of the coefficients are 0, do an inference, and then seek to approximate that inference with models in which the coefficients are 0.

If many of the coefficents are inferred to be *close to 0, as a practical matter* this will work well!!!!

Note:
If some of the coefficients really are 0, and you just think they may be close to 0 as a practical matter, you will do fine!!

If none of the coefficents are exactly 0, and you see to find the ones that are you will do terrible, and no playing with the prior or reengineering of the p-value will save you.

BART: $Y = f(x) + \epsilon, \;\; \epsilon \sim N(0, \sigma^2)$.

Step 1 (fit):

Run BART using the observed data (X, y).

This will give us MCMC draws (f_d, σ_d), $d = 1, 2, \ldots, D$ from the posterior.

For example: $\hat{f}(x) = \frac{1}{D} \sum_{d=1}^{D} f_d(x)$.

- The BART prior assumes all x matter.
- All inference comes from this step.
  We will just be approximating this inference.

Step 2 (choose future x):

Choose a set $\{x_i^f\}$ of *future* x at which you want to predict Y.

We will let X^f denotes the set of x_i^f.

Of course, given data (y, X), a simple default is $X^f = X$, but

the choice of X^f can and should matter !!!

We have a nice example that hopefully we will get to.

The choice of model should depend on what you want to do with it !!

Step 3 (fit the fit):

Let $|S|$ be the size of the set S (number of variables in our case).

For each $j = 1, 2, \ldots, p-1$:

$$\underset{\gamma_S,\, |S|=j}{\text{minimize}} \; ||\hat{f}(X^f) - \gamma_S(X^f)||^2$$

where (of course),

$$||\hat{f}(X^f) - \gamma_S(X^f)||^2 = \sum (\hat{f}(x_i^f) - \gamma_S(x_i^f))^2$$

For each j, we need a subset S of j variables and an approximating function γ_S using only those variables.

Remember, we don't want to make assumptions about f and hence γ_S.

We can't solve this so, as usual, we approximate our problem with a computationally feasible strategy:

(1):

Use backwards and forwards selection to search for subsets.
As in the linear case, can do all subsets for moderate p.

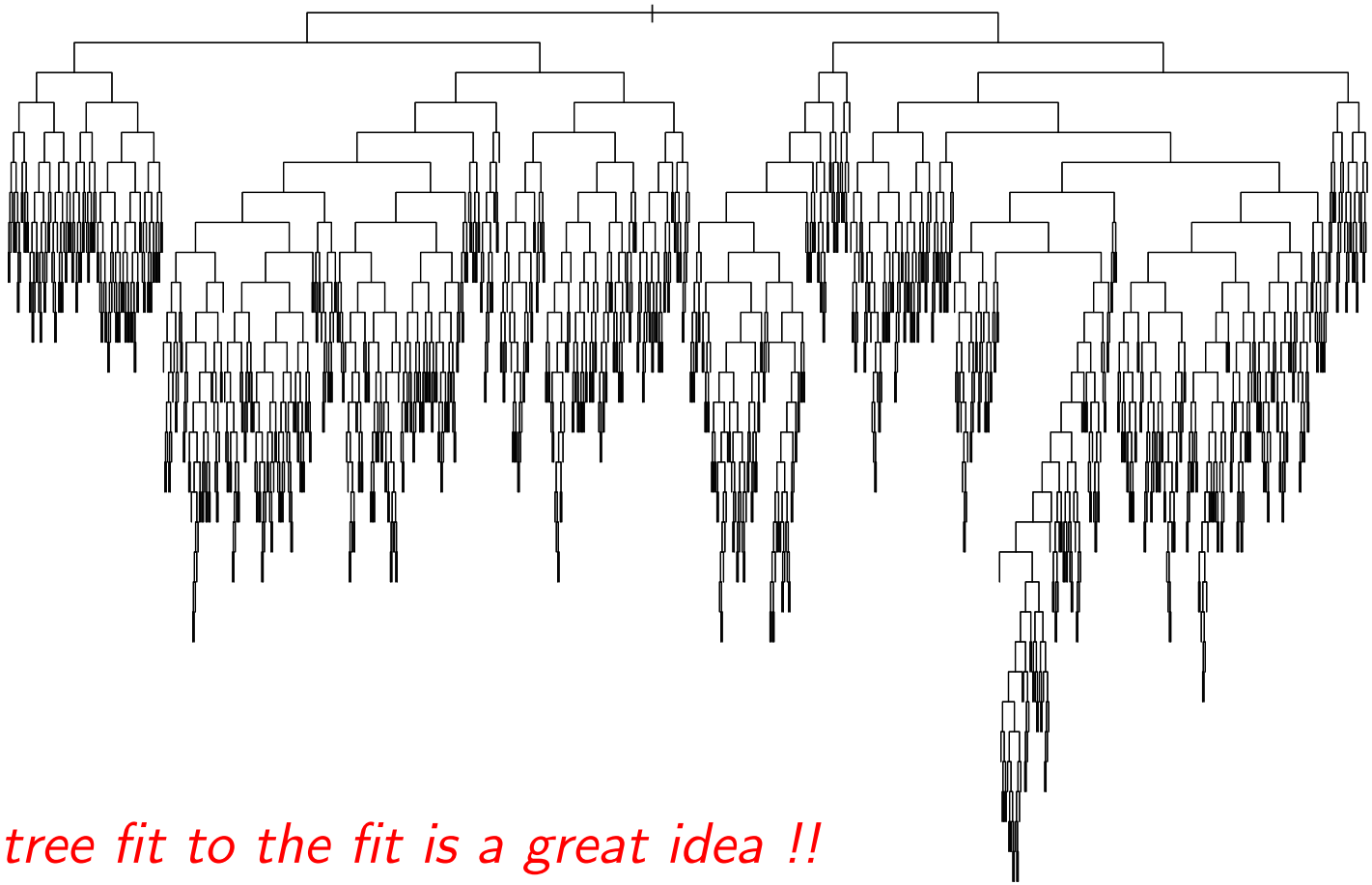
(2):

Rather than run our nonparametric method (BART) using subsets of the x variables to get $\gamma_S(X^f)$, fit a big tree to $\hat{f}(X^f)$ using subsets of the x variables.

note: BART is not engineered to fit perfectly.

(2) is the one simple useful idea in the work.

A big tree fit to the data is a terrible idea (unless you bag).



A big tree fit to the fit is a great idea !!
Forwards selection on the fit is a great idea !!
and *it is pretty fast !!!!*

$\hat{f}(X^f)$ is the posterior mean from step 1.

So, for example, the first step is forwards is to fit a big tree to each data set:

$$(y = \hat{f}(X^f), X = x_j^f), \;\; j = 1, 2, \ldots, p$$

and then pick the x_j that gives you the best fit.

Step 3(a) (Run BART on selected subsets):

Once we have the subsets S_j, $j = 1, 2, \ldots, p-1$,
we can run the $p-1$ BARTS using the data (y, X_{S_j}).

Sometimes, this gives a better approximation than the big tree.

So, we replace the $\gamma_S(X^f)$ we got from the big trees with the BART posterior means from runs on the found subsets.

But we have to use the big trees to search !!

Step 4 (Assess Uncertainty of Approximation Error):

To informally assess a subset choice we look at the posterior distribution of

$$D(f(X^f), \gamma_S(X^f))$$

where f is the random variable, and we consider the posterior distribution of f.

We consider a variety of distance functions D.

That is, we look at the draws:

$$D(f_d(X^f), \gamma_S(X^f))$$

and the differences

$$D(f_d(X^f), \gamma_S(X^f)) - D(f_d(X^f), \hat{f}(X^f))$$

That is, we look at the draws:

$$D(f_d(X^f), \gamma_S(X^f))$$

Given the information in the data, what is the likely size of the approximation error if you use only the variables in S.

and the differences

$$D(f_d(X^f), \gamma_S(X^f)) - D(f_d(X^f), \hat{f}(X^f))$$

Given the information in the data, what is the likely increase in the size of the approximation error if you use only the variables in S.

Choice of D:

For numeric Y the most obvious choice is RMSE:

$$D(f_d(X^f), \gamma_S(X^f)) = \sqrt{\frac{1}{n_f}\sum_{i=1}^{n_f}(f_d(x_i^f) - \gamma_S(x_i^f))^2}$$

But we also have been thinking about distances which incorporate the σ draws, e.g.:

$$D(f_d(X^f), \gamma_S(X^f), \sigma_d) = \sqrt{\frac{1}{n_f}\sum_{i=1}^{n_f}(\frac{f_d(x_i^f) - \gamma_S(x_i^f)}{\sigma_d})^2}$$

you might want to think about the approximation error "relative to" the unvoidable predictive uncertainty represented by σ.

We also consider:

$$D(f_d(X^f), \gamma_S(X^f), \sigma_d) = \sqrt{\frac{1}{n_f}\sum (f_d(x_i^f) - \gamma_x(x_i^f))^2 + \sigma_d^2} - \sigma_d$$

For discrete Y outcomes, we use Kullback-Leibler distances or just the absolute value of the difference in probabilities.

Note that look at these posteriors *does not* fit in with formal Bayesian decision making!!

Summary:

- Step 1: get BART inference using data (X, y).
- Step 2: choose future x's X^f.
- Step 3: find subsets S such that $|S| = j \in \{1, 2, \ldots, p\}$ using "big tree" γ_S.
- Step 3(a): Given the subsets from (3) run BART on (y, X_S) to get alternative γ_S.
- Step 4: Assess uncertainty with draws $D(f_d(X^f), \gamma_S(X^f))$.

In my mind Step 1 gives me the "true" Baysian inference.

After that I am just post-processing/making decisions!!!

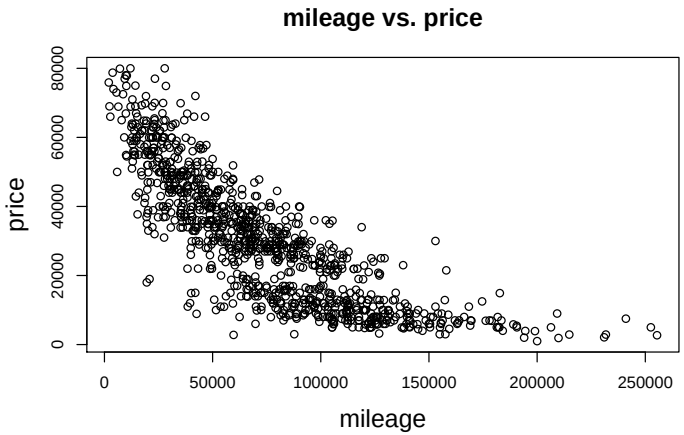
NO "sparsity prior".
NO "multiple comparisons".
NO "model probabilities".
....

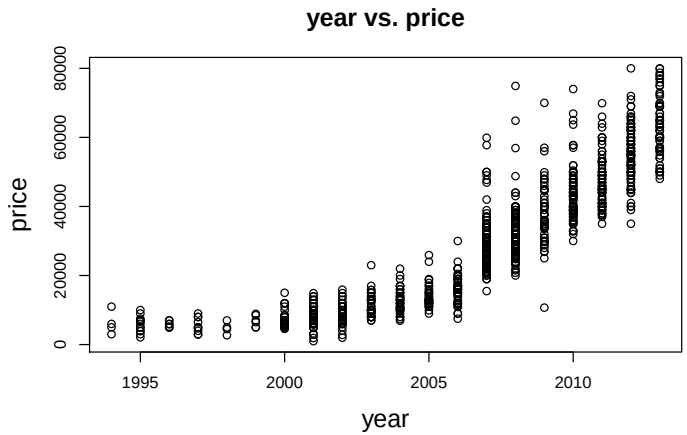
3. Cars Data

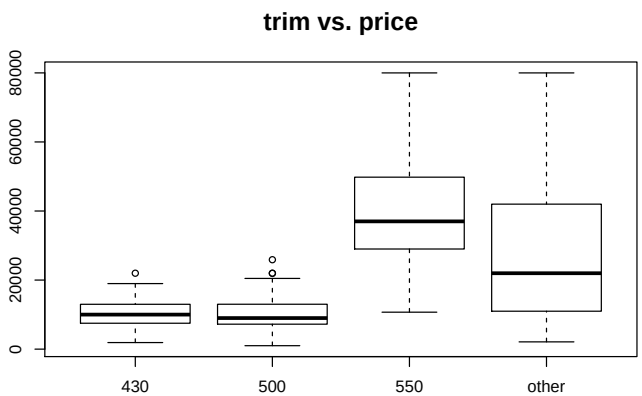
y: price of a used car (Mercedes M class). x: things about the car.
n=1,000.
p = 15 (after making dummies).

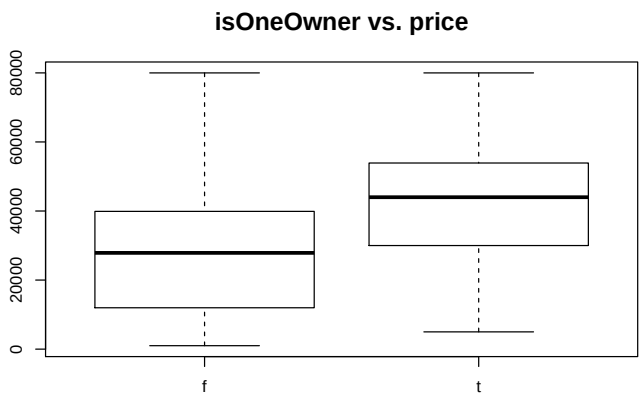
```
n,p:  1000 15
     price           trim     isOneOwner    mileage            year
 Min.   :  995   430  :143   f:841      Min.   :  1997   Min.   :1994
 1st Qu.:12995   500  :127   t:159      1st Qu.: 40133   1st Qu.:2004
 Median :29800   550  :591              Median : 67920   Median :2007
 Mean   :30583   other:139              Mean   : 73652   Mean   :2007
 3rd Qu.:43992                          3rd Qu.:100138   3rd Qu.:2010
 Max.   :79995                          Max.   :255419   Max.   :2013
    color     displacement
 Black :415   4.6  :137
 other :227   5.5  :476
 Silver:213   other:387
 White :145
```

Y vs. four of the x's.









Step 1::

Do a BART inference using all the x's.

This gives us $\hat{f}(X^f)$ for any X^f..

Step 2:

In this example X^f will be the observed x.

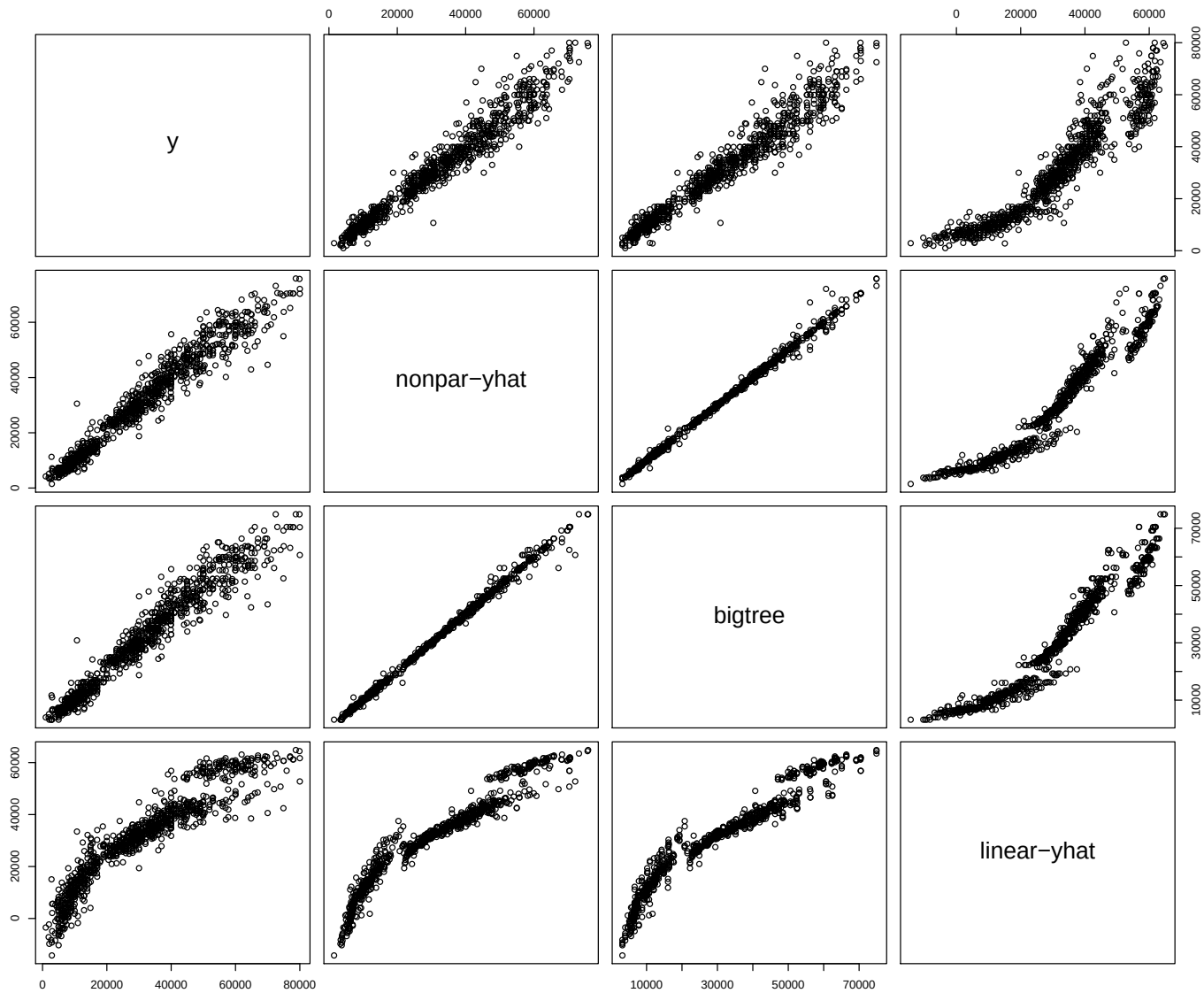
Using all the x's:

y=price.

nonpar-hat:
Bayes nonparametric fit to y: $\hat{f}(X^f)$.

bigtree:
fit big tree to $\hat{f}(X^f)$.

linear-yhat:
linear fit to y.



Step 3:

fit-the-fit:

Let $\gamma_j(x)$ be $\gamma_S(x)$ for $S =$ a subset with j variables.

Find subsets such that $\gamma_j(X^f) \approx \hat{f}(X^f)$

Forward:

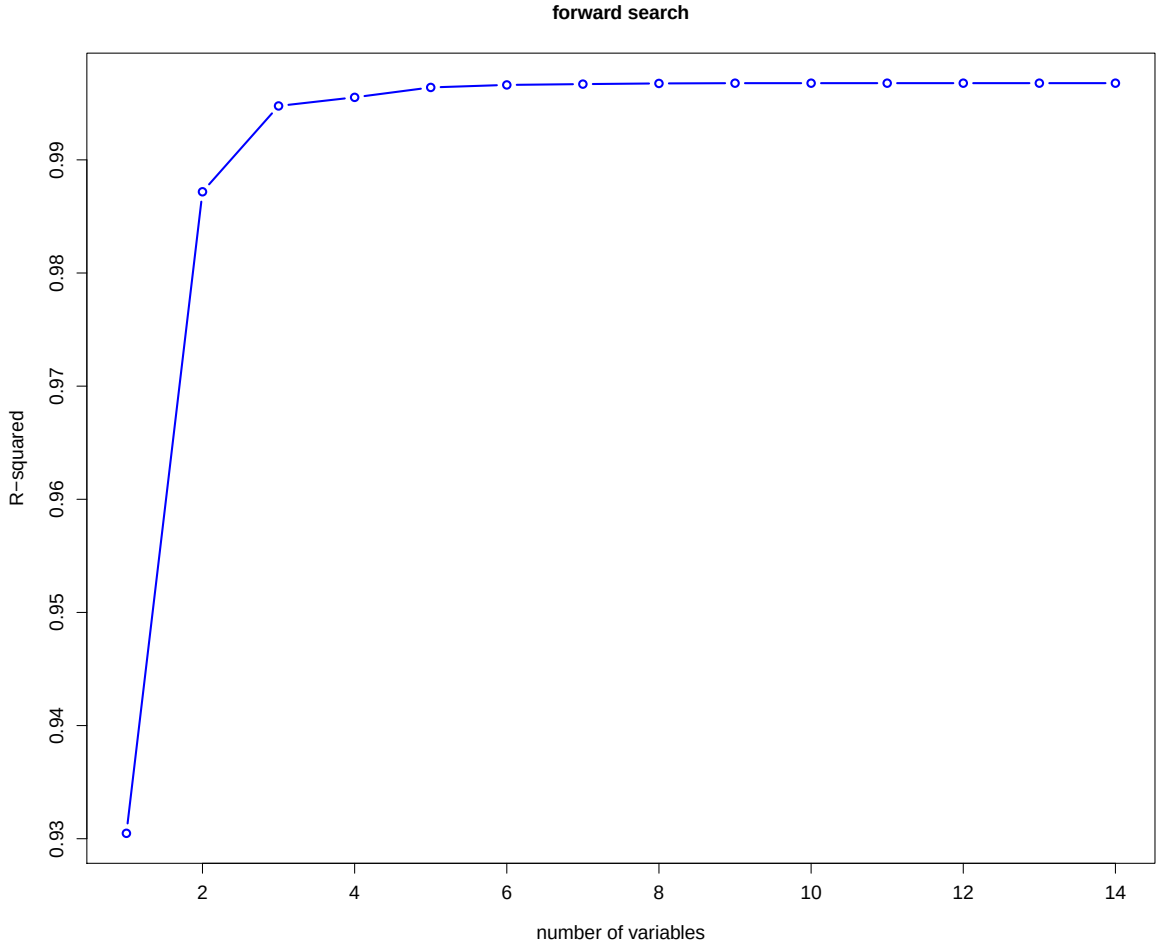
At each step,
add in the x
which makes

$\gamma_j(X^f)$
(fit using big tree)

closest to

$\hat{f}(X^f)$,
the bayes fit.

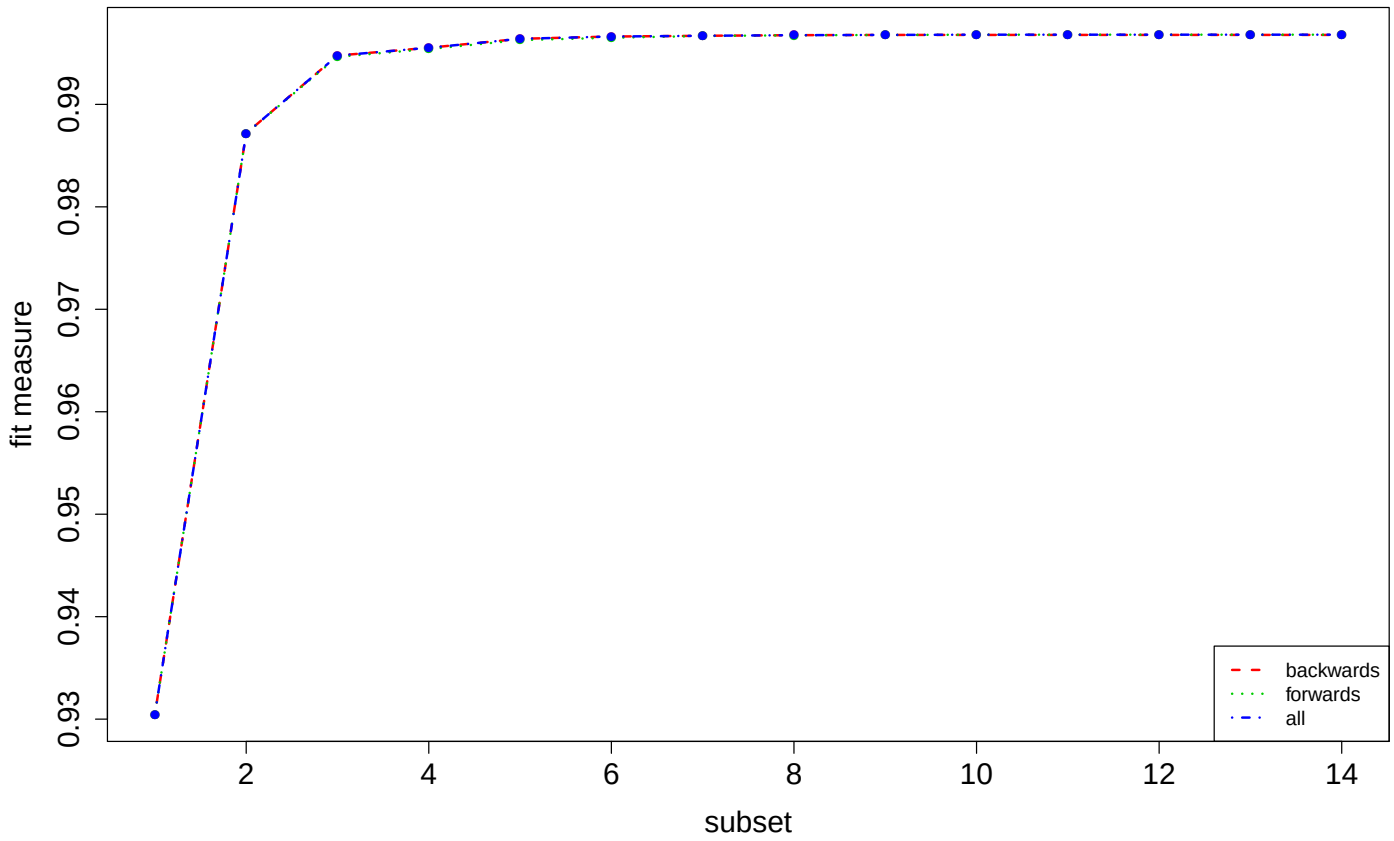
Using "R^2"
type measure.



Variables:

```
year
year mileage
year mileage trim.other
year mileage trim.other displacement.5.5
year mileage trim.other displacement.5.5 color.Black
year mileage trim.other displacement.5.5 color.Black color.White
```

Compare forwards, backwards, and all possible subsets search.



Remember, inference is not an issue.

Top 6, Forwards:

```
year
year mileage
year mileage trim.other
year mileage trim.other displacement.5.5
year mileage trim.other displacement.5.5 color.Black
year mileage trim.other displacement.5.5 color.Black color.White
```

Top 6, Backwards:

```
year
mileage year
mileage year trim.550
mileage year trim.550 displacement.other
mileage year trim.550 color.Black displacement.other
mileage year trim.550 color.Black color.Silver displacement.other
```

Top 6, All:

```
year
mileage year
mileage year trim.other
mileage year trim.other displacement.5.5
mileage year trim.other color.Black displacement.5.5
mileage year trim.other color.Black color.White displacement.5.5
```

Step 3(a) :

Once we have the subsets, we often replace the $\gamma_j(x)$ obtained from the big trees, with the fit from rerunning BART with the data and the subset of x's.

While our basic claim is that the big tree fit is good enough to run forwards (or backwards or all subsets), refitting BART on the subsets with data (y, X_S) can give us a better fit to $\hat{f}(X^f)$.

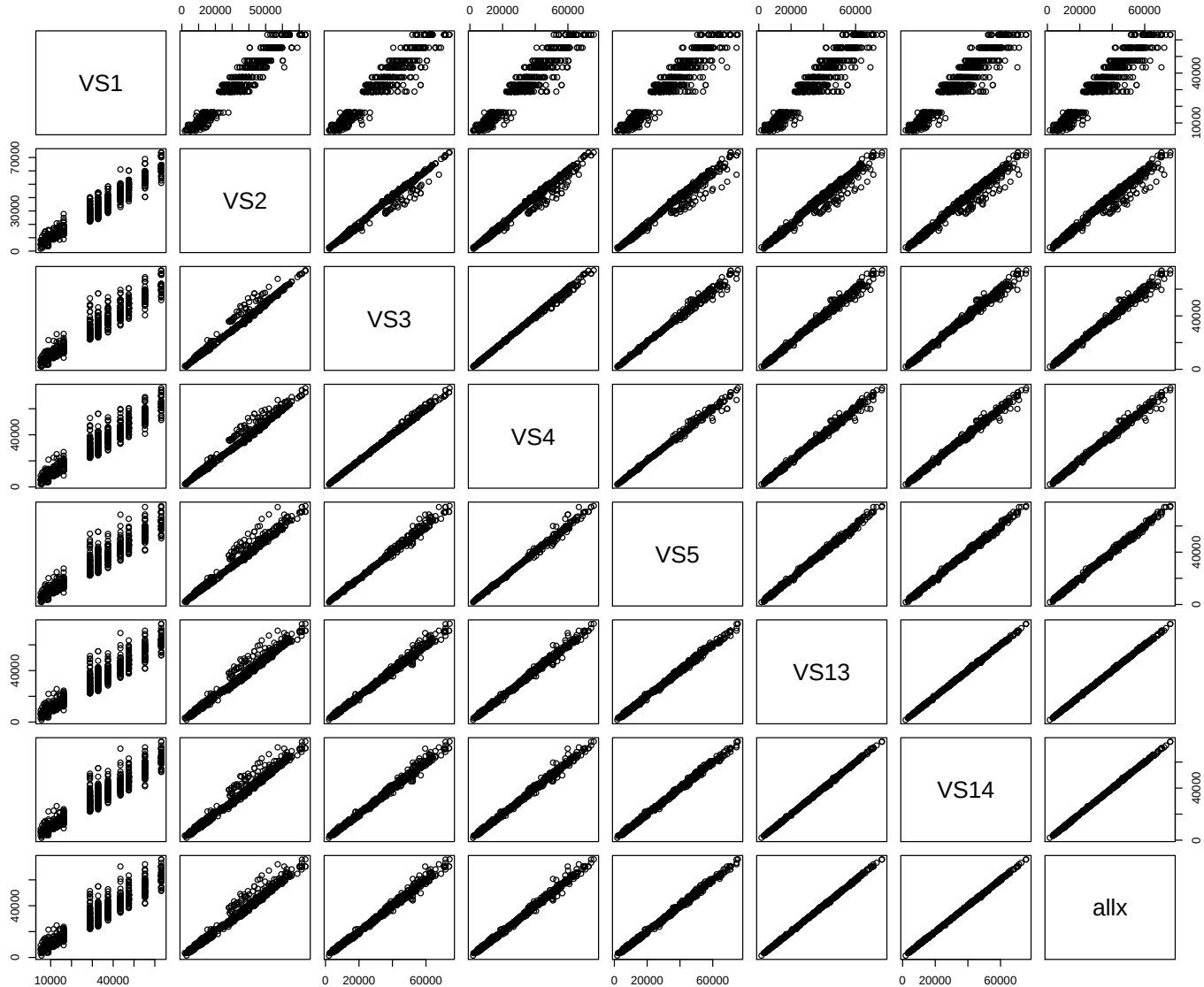
So,

Rerun the nonparametric fit (BART) using the identified subsets.

Let $\hat{f}_j(x)$ be the fit using the subset of variables of size j.

Let $\gamma_j(x) = \hat{f}_j(x)$.

Did it work ?????, VSj $\sim \gamma_j(x)$.



Step 4:

To assess our uncertainty we look at posteriors of $D(f(\mathbf{X}), \gamma_j(\mathbf{X}))$ for various j.

Here, f is the random variable.

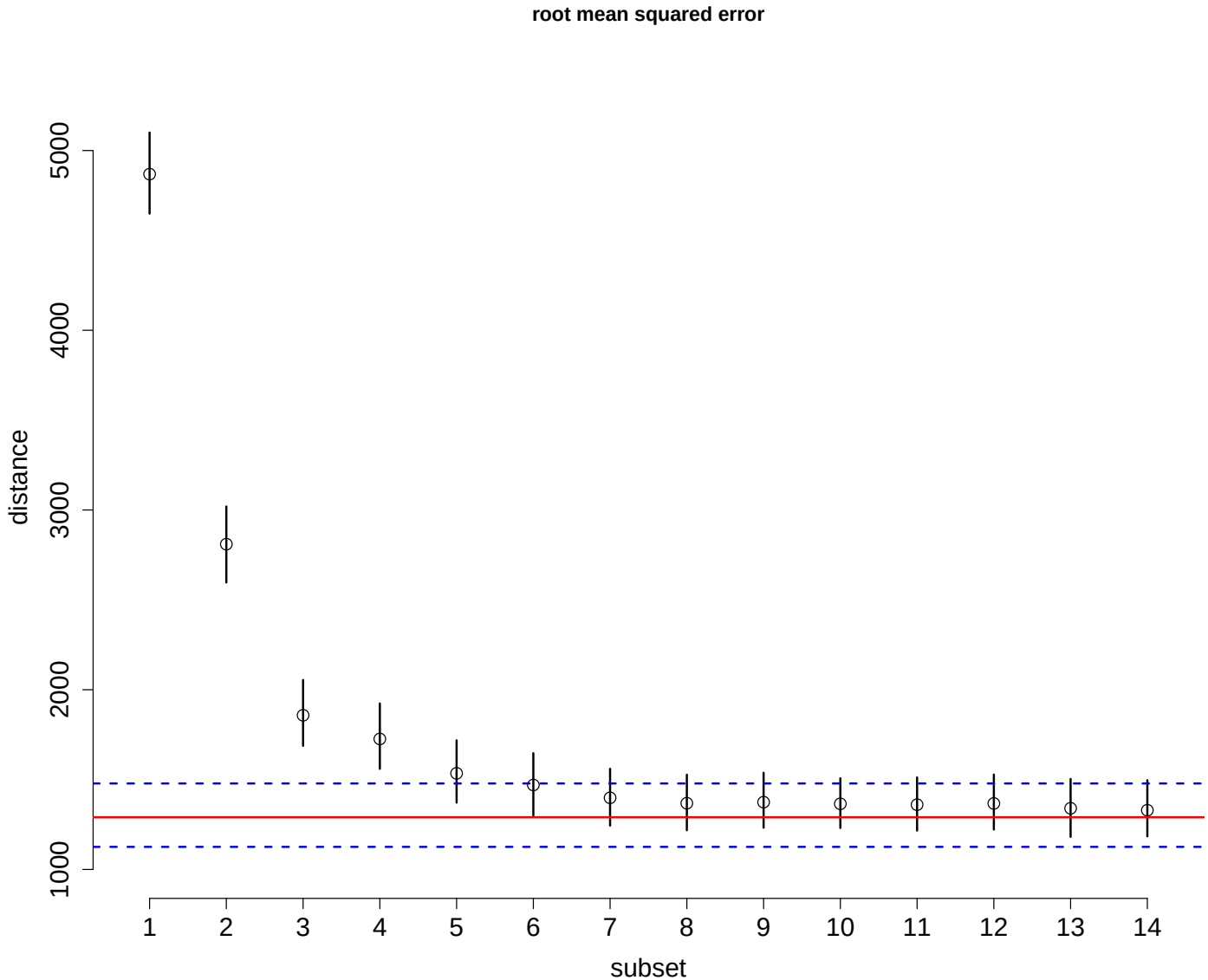
For MCMC draw f_d, we compute $D(f_d(X^f), \gamma_j(X^f))$.

Our first D is RMSE

$$D = \sqrt{\frac{1}{n_f} \sum_{x \in X^f} (f_d(x) - \gamma_j(x))^2}$$

where n_f = number of $x \in X^f$.

Look at draws $D(f_d(X^f), \gamma_j(X^f))$ for various j.
Horizontal blue and red lines are for $j = p$, $\gamma_p(X^f) = \hat{f}(X^f)$.



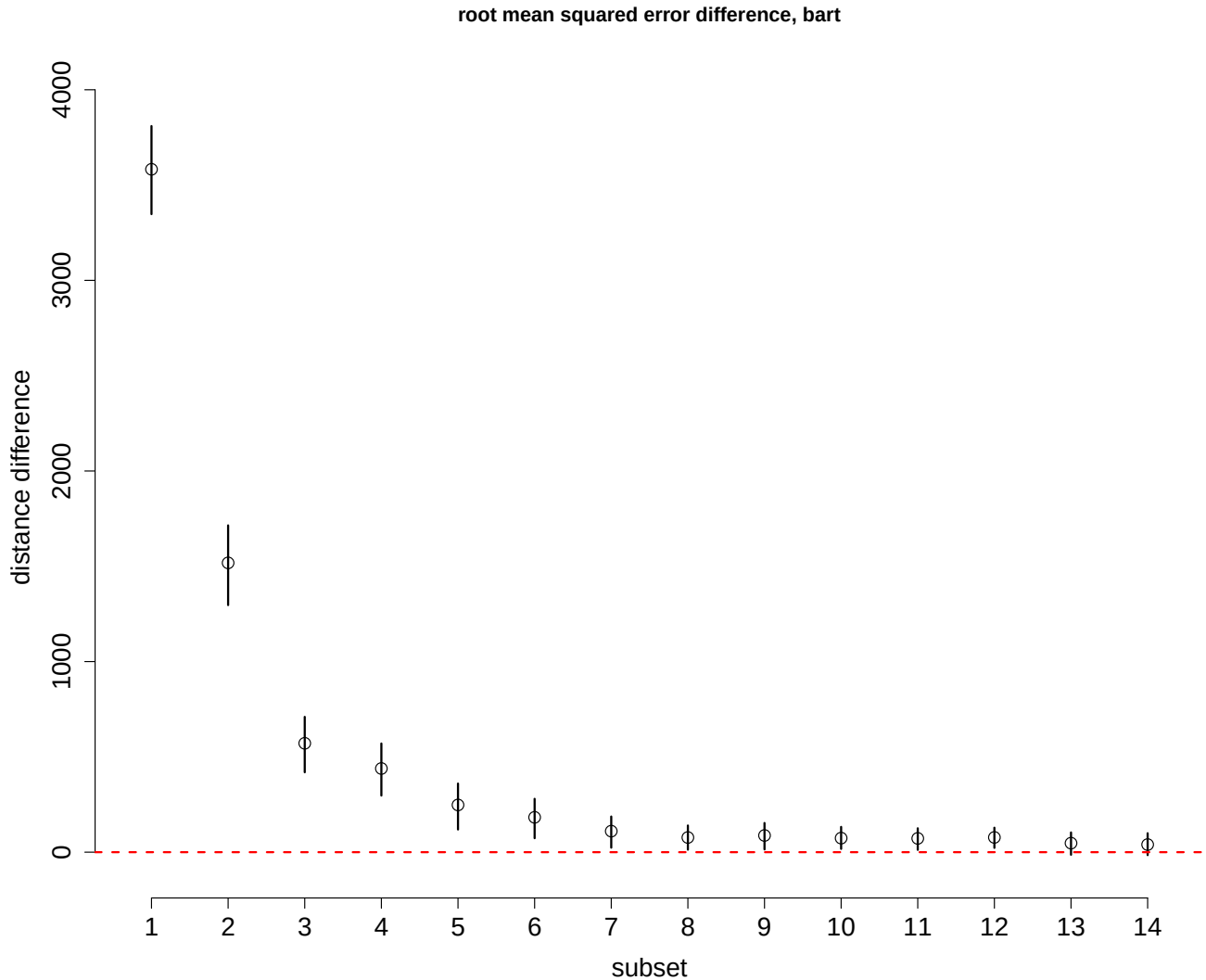
little vertical lines are 95% intervals. Circles at posterior mean.

Now we look at the posterior distribution of the difference in distances

For each draw f_d we compute:

$$D(f_d(X^f), \gamma_j(X^f)) - D(f_d(X^f), \hat{f}(X^f)).$$

rmse distance, difference.



These cars cost $30,000 or more.

As a practical matter, 3 or 5 variables does the trick!!!!!

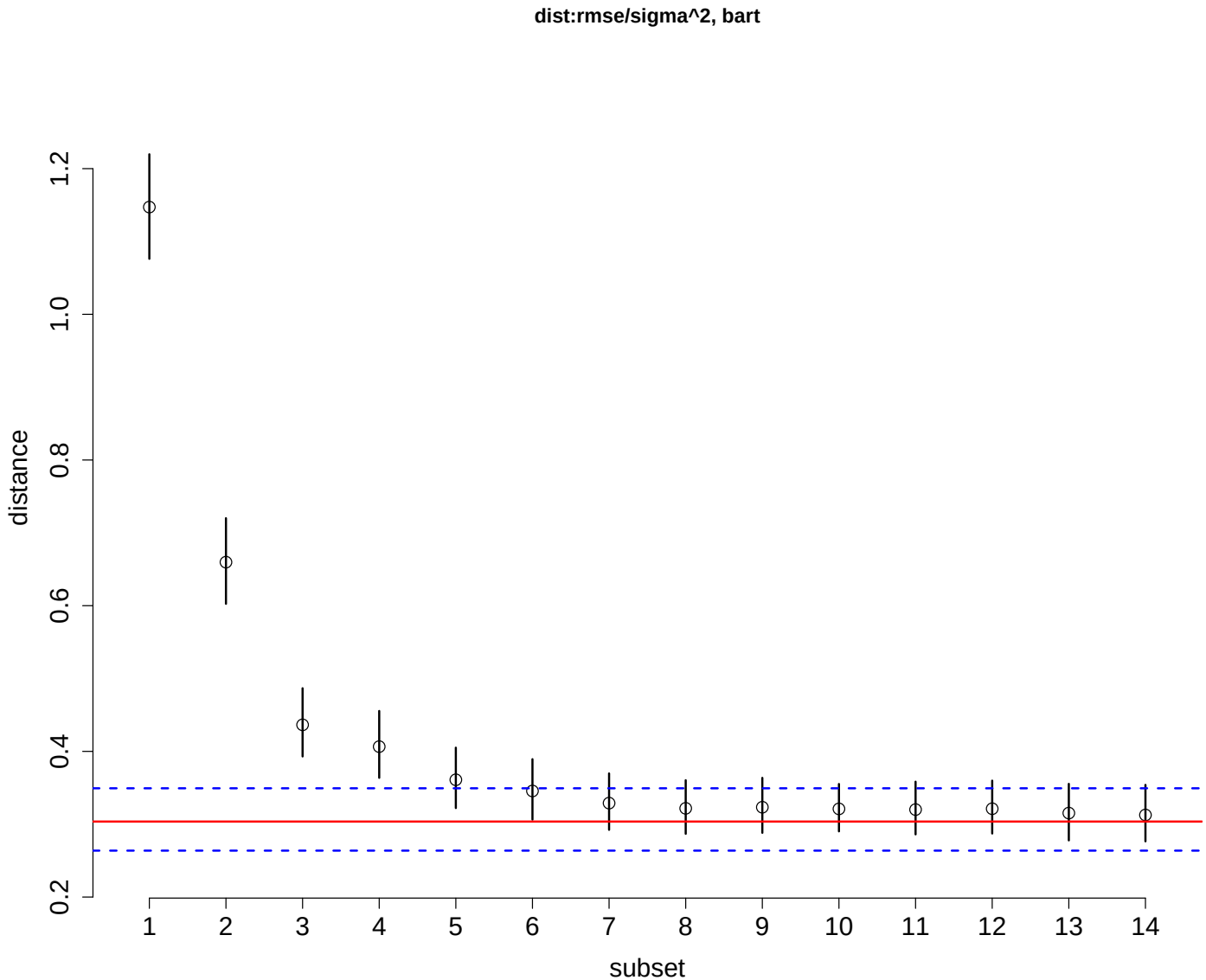
Another distance measure:

$$D = \sqrt{\frac{1}{n_f} \sum_{x \in X^f} \frac{(f(x) - \gamma_j(x))^2}{\sigma^2}}$$

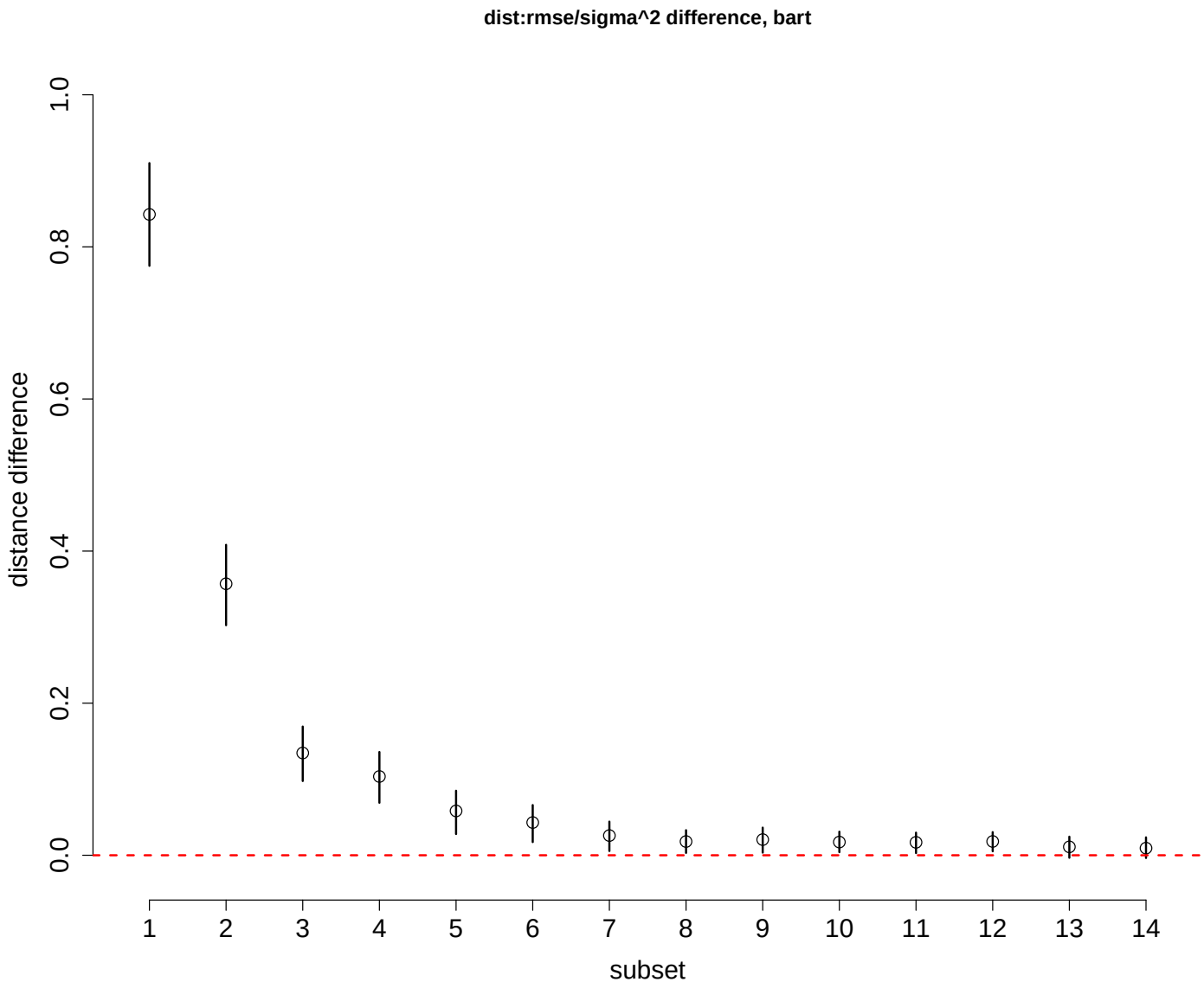
The idea is that if $f(x) - \gamma_j(x)$ is small relative to σ then it does not matter.

This gives us a unitless D.

Small cars example. rmse/sigma distance.



Small cars example. Diff sigsqrt.



4. Hockey

Glen Healey, commenting on an NHL broadcast:

> *Referees are predictable. The flames have had three penalties, I guarantee you the oilers will have three.*

Well, *guarantee* seems a bit strong,
but there is something to it.

How predictable are referees?

"Reversal of fortune: a statistical analysis of penalty calls in the national hockey league", (2014), Journal of Quantitative Analysis in Sports 10 (2), 207-224 (with Jason Abrevaya).

Got data on every penalty in every
(regular season) game for 7 seasons around the time they switched from one referee to two.

For each penalty (after the first one in a game) let

`revcall` =

1 if current penalty and previous penalty are on different teams,

0 otherwise.

You know a penalty has just been called,
which team is it on?
is it a reverse call on the other team???

Mean of `revcall` is .6 !

Table: Variable Descriptions

Variable	Description	Mean	Min	Max
Dependent variable				
`revcall`	1 if current penalty and last penalty are on different teams	0.589	0	1
Indicator-Variable Covariates				
`ppgoal`	1 if last penalty resulted in a power-play goal	0.157	0	1
`home`	1 if last penalty was called on the home team	0.483	0	1
`inrow2`	1 if last two penalties called on the same team	0.354	0	1
`inrow3`	1 if last three penalties called on the same team	0.107	0	1
`inrow4`	1 if last four penalties called on the same team	0.027	0	1
`tworef`	1 if game is officiated by two referees	0.414	0	1
Categorical-variable covariate				
`season`	Season that game is played		1	7
Other covariates				
`timeingame`	Time in the game (in minutes)	31.44	0.43	59.98
`dayofseason`	Number of days since season began	95.95	1	201
`numpen`	Number of penalties called so far (in the game)	5.76	2	21
`timebetpens`	Time (in minutes) since the last penalty call	5.96	0.02	55.13
`goaldiff`	Goals for last penalized team minus goals for opponent	-0.02	-10	10
`gf1`	Goals/game scored by the last team penalized	2.78	1.84	4.40
`ga1`	Goals/game allowed by the last team penalized	2.75	1.98	4.44
`pf1`	Penalties/game committed by the last team penalized	6.01	4.11	8.37
`pa1`	Penalties/game by opponents of the last team penalized	5.97	4.33	8.25
`gf2`	Goals/game scored by other team (not just penalized)	2.78	1.84	4.40
`ga2`	Goals/game allowed by other team	2.78	1.98	4.44
`pf2`	Penalties/game committed by other team	5.96	4.11	8.37
`pa2`	Penalties/game by opponents of other team	5.98	4.33	8.25

$n = 57,883.$

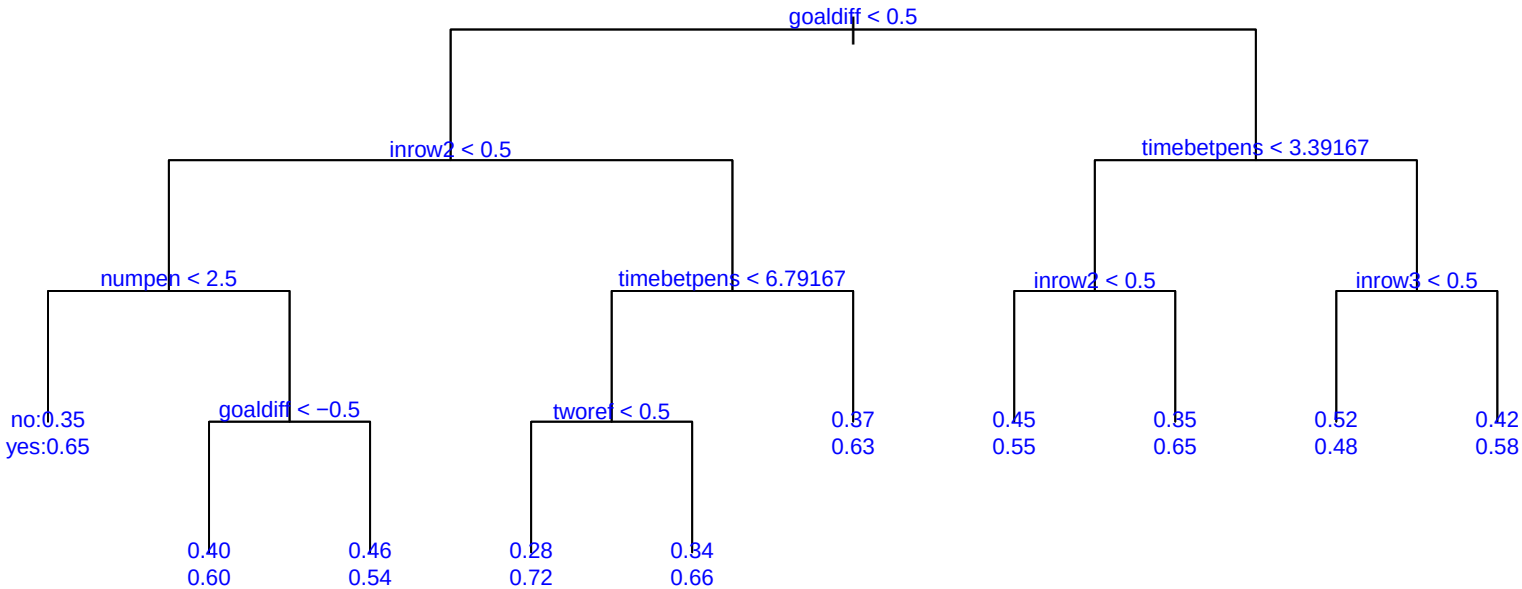
How is `revcall` related to the variables?

	inrow2=0	inrow2=1
revcall=0	0.44	0.36
revcall=1	0.56	0.64

If the last two calls were on the same team (`inrow2=1`), then 64% of the time, the next call will *reverse* and be on the other team.

Otherwise, ((`inrow2=0`), it is only 56%.

A Tree



- Last penalized was not ahead
- Last two penalties on same team
- Not long since last call
- one ref

$\Rightarrow$

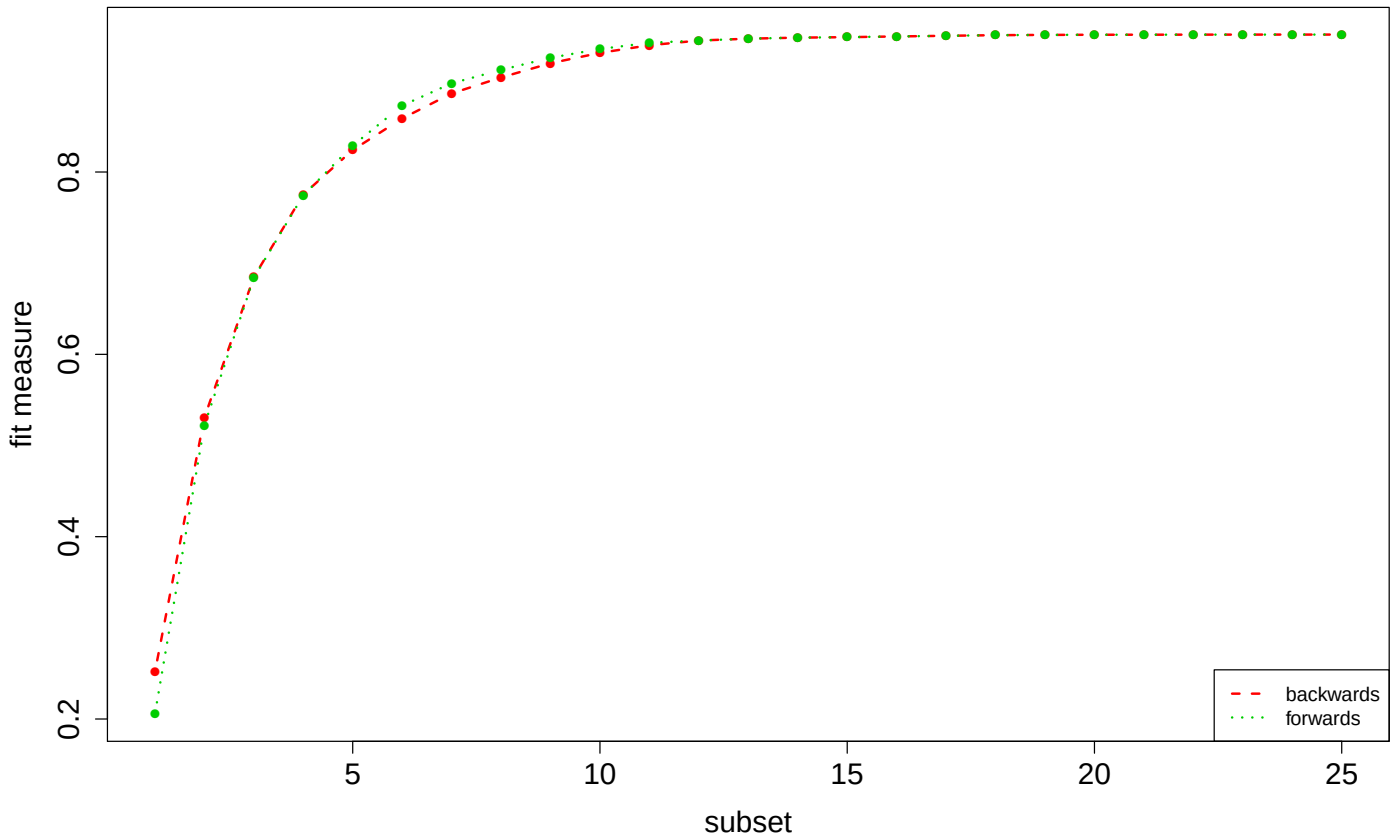
72% revcall.

- Last penalized was ahead
- it has been a while since last penalty
- last three calls not on same team

$\Rightarrow$

48% revcall.

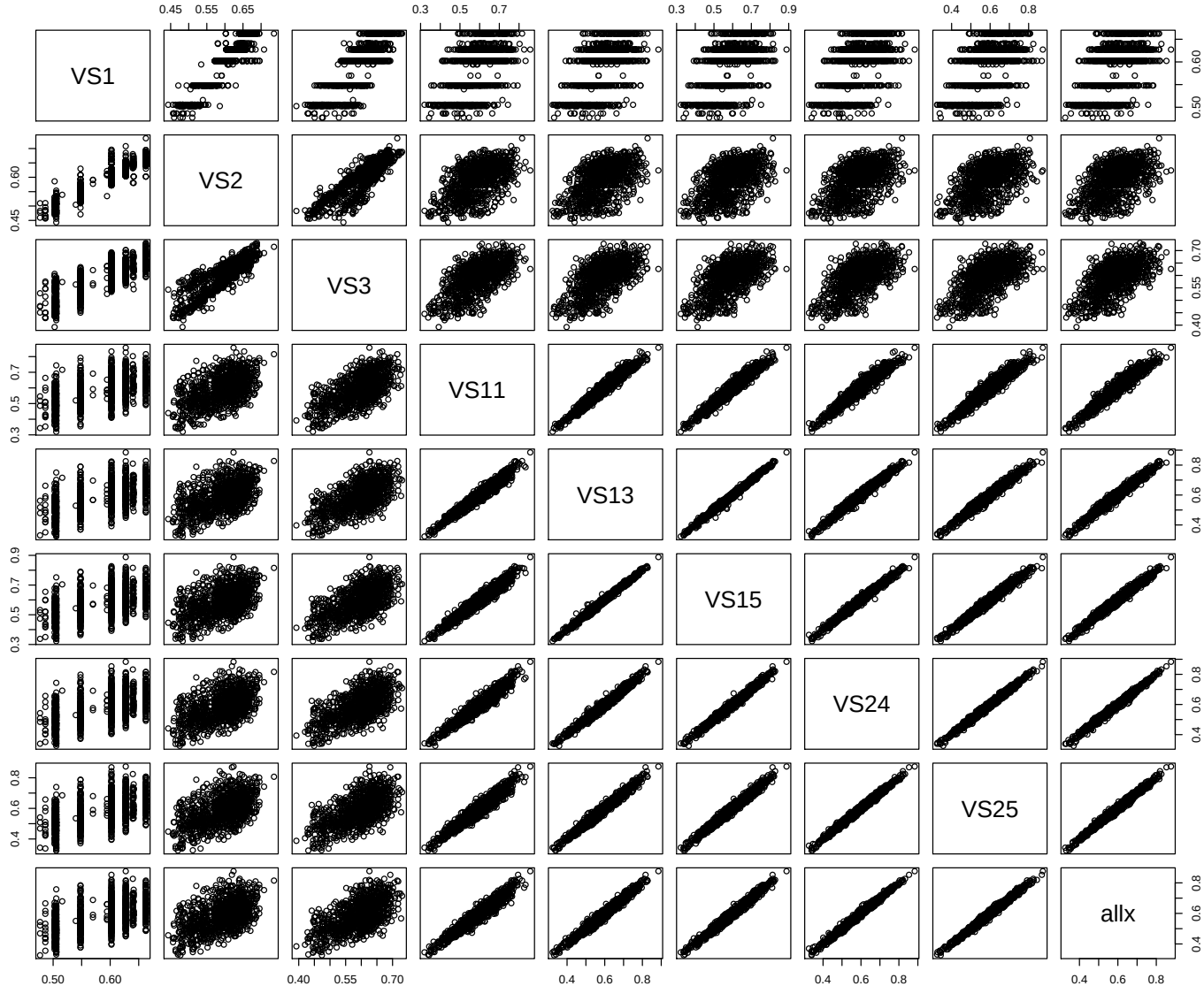
Hockey example. compare forwards and backwards.



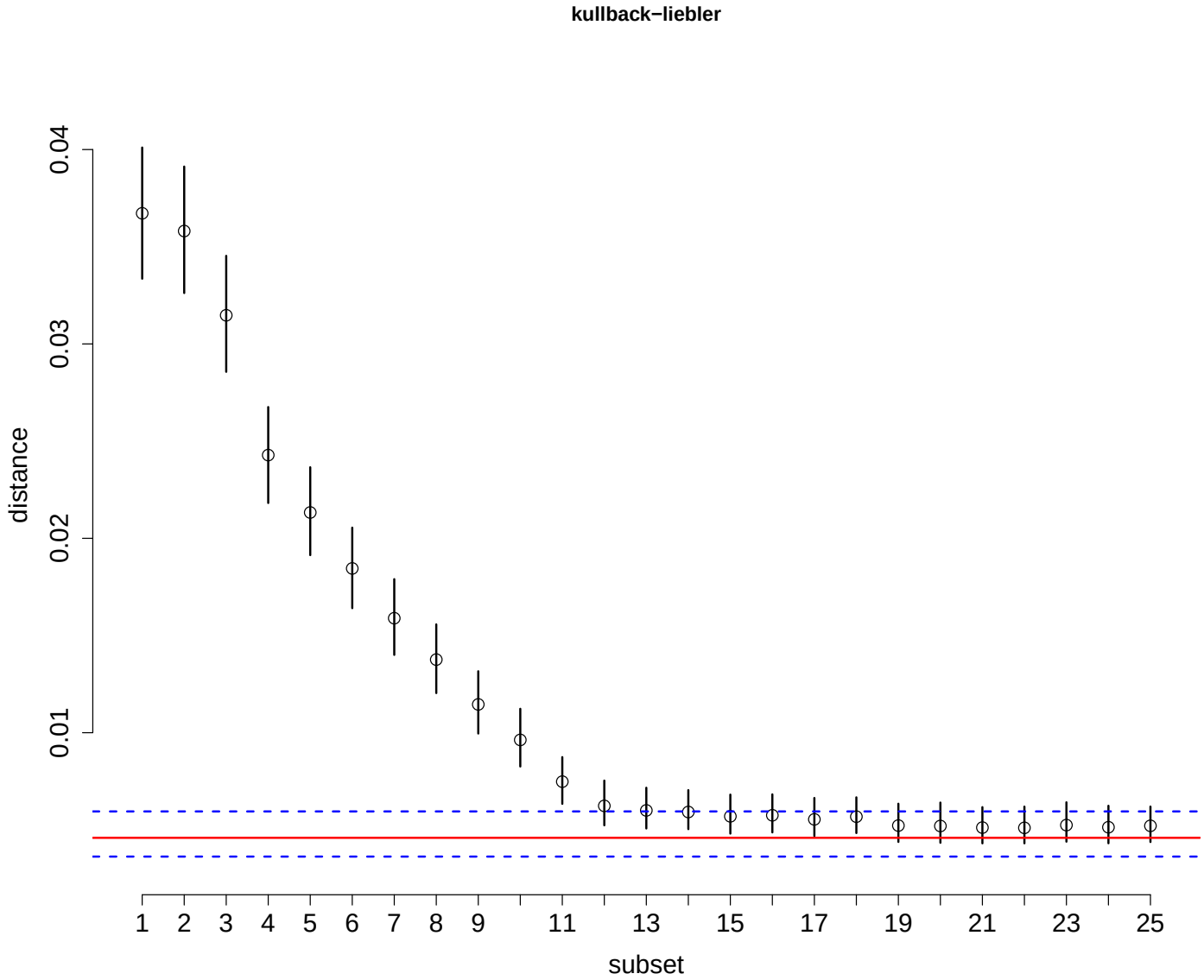
Hockey example. Forwards and backwards subsets.

numvar	forwards	backwards
1	goaldiff	timebetpens
2	timeingame	pf1
3	timebetpens	goaldiff
4	inrow2	inrow2
5	pf1	numpen
6	pf2	pf2
7	numpen	home
8	home	inrow3
9	inrow3	pa1
10	pa1	pa2
11	pa2	tworef
12	tworef	timeingame
13	dayofseason	dayofseason
14	gf2	gf2
15	X2000	X2000
16	ga1	ga1
17	gf1	gf1
18	ga2	ga2
19	ppgoal	ppgoal
20	X1996	X1996
21	inrow4	inrow4
22	X1998	X1998
23	X2001	X2001
24	X1995	X1995
25	X1997	X1997
26	X1999	X1999

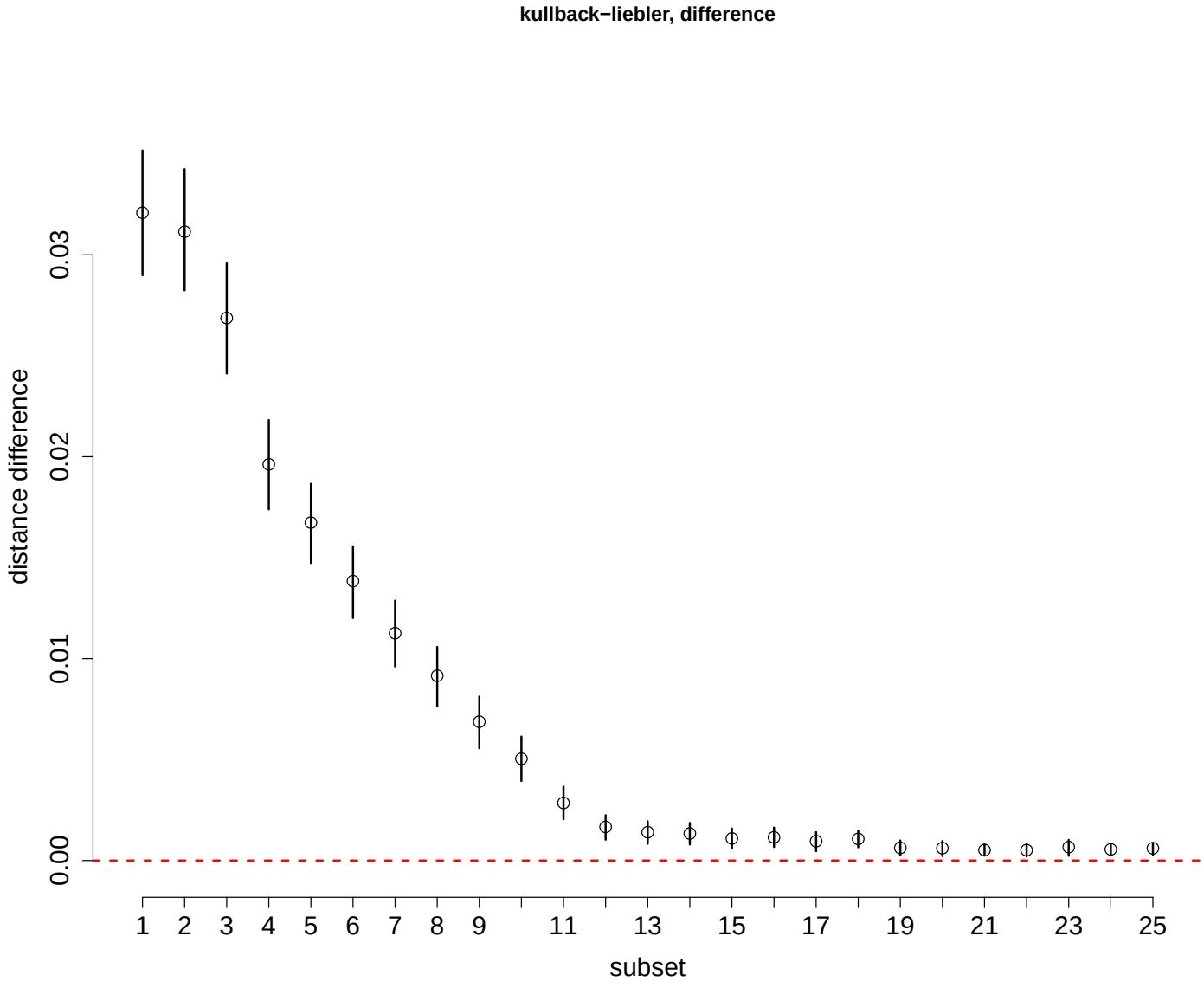
Hockey example. bartsub-pairs. forward vs.



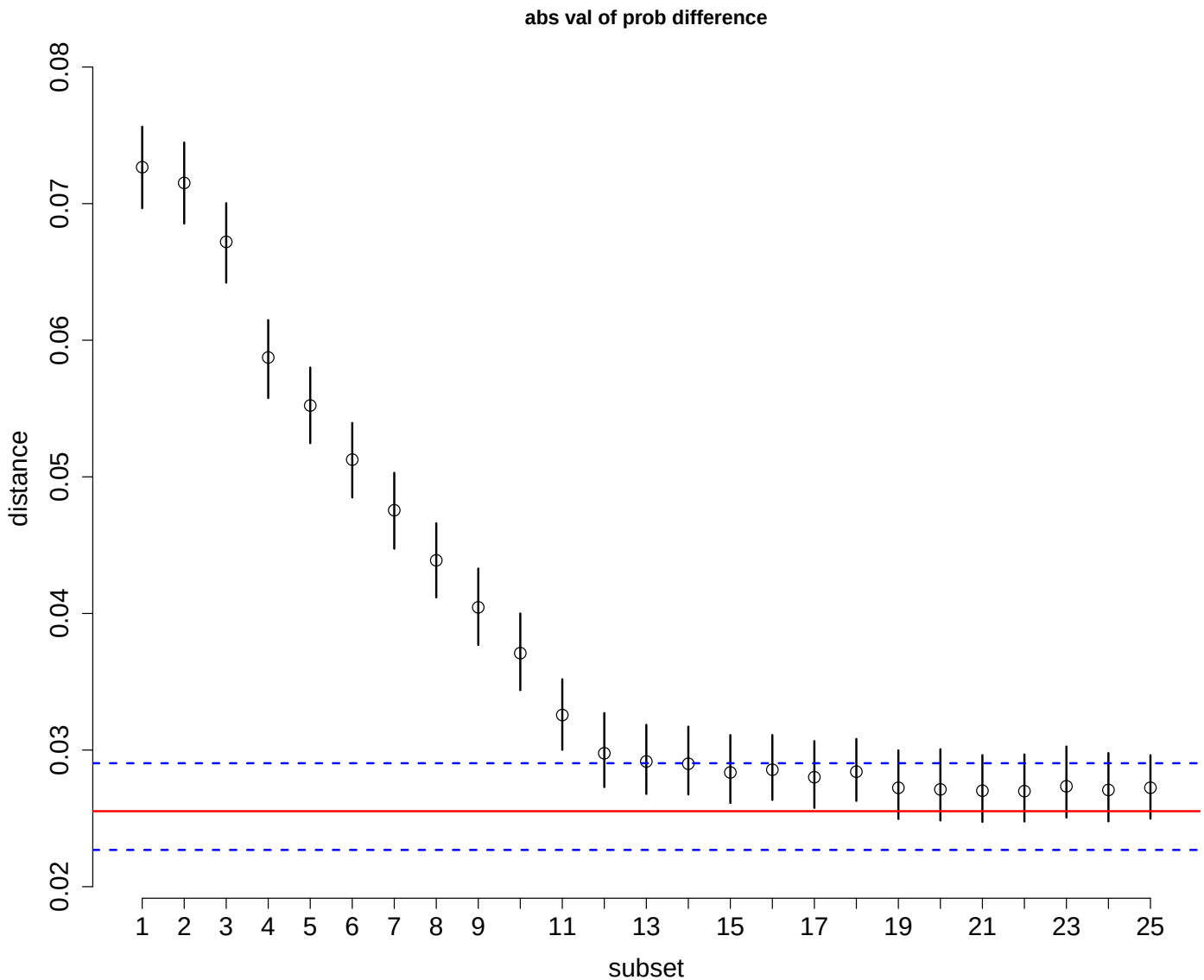
Kullback-Leibler.



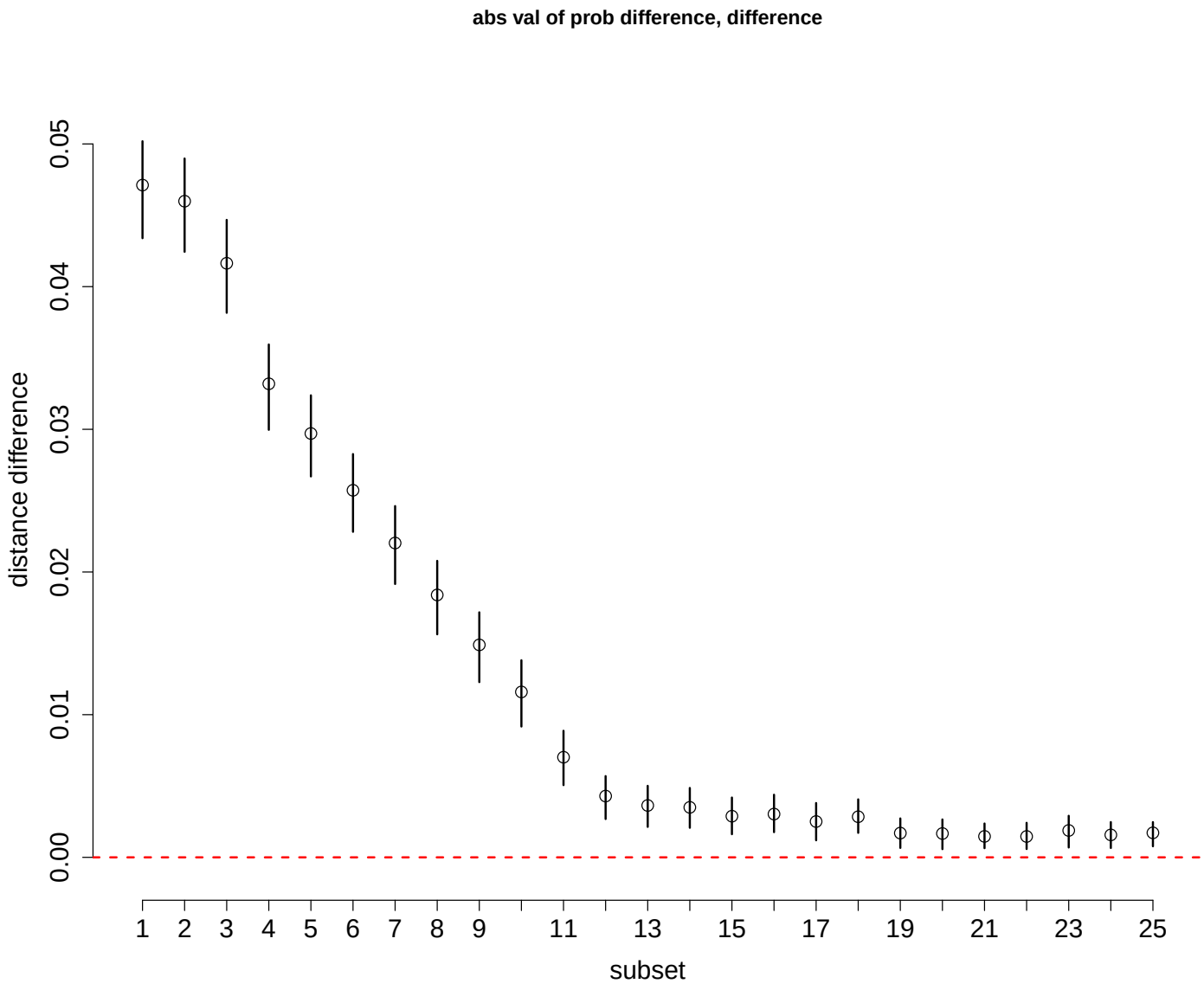
Kullback-Leibler difference.



abs prob difference.



abs prob difference difference.



We actually only used 47,883 of the 57,883 observations in the results so far.

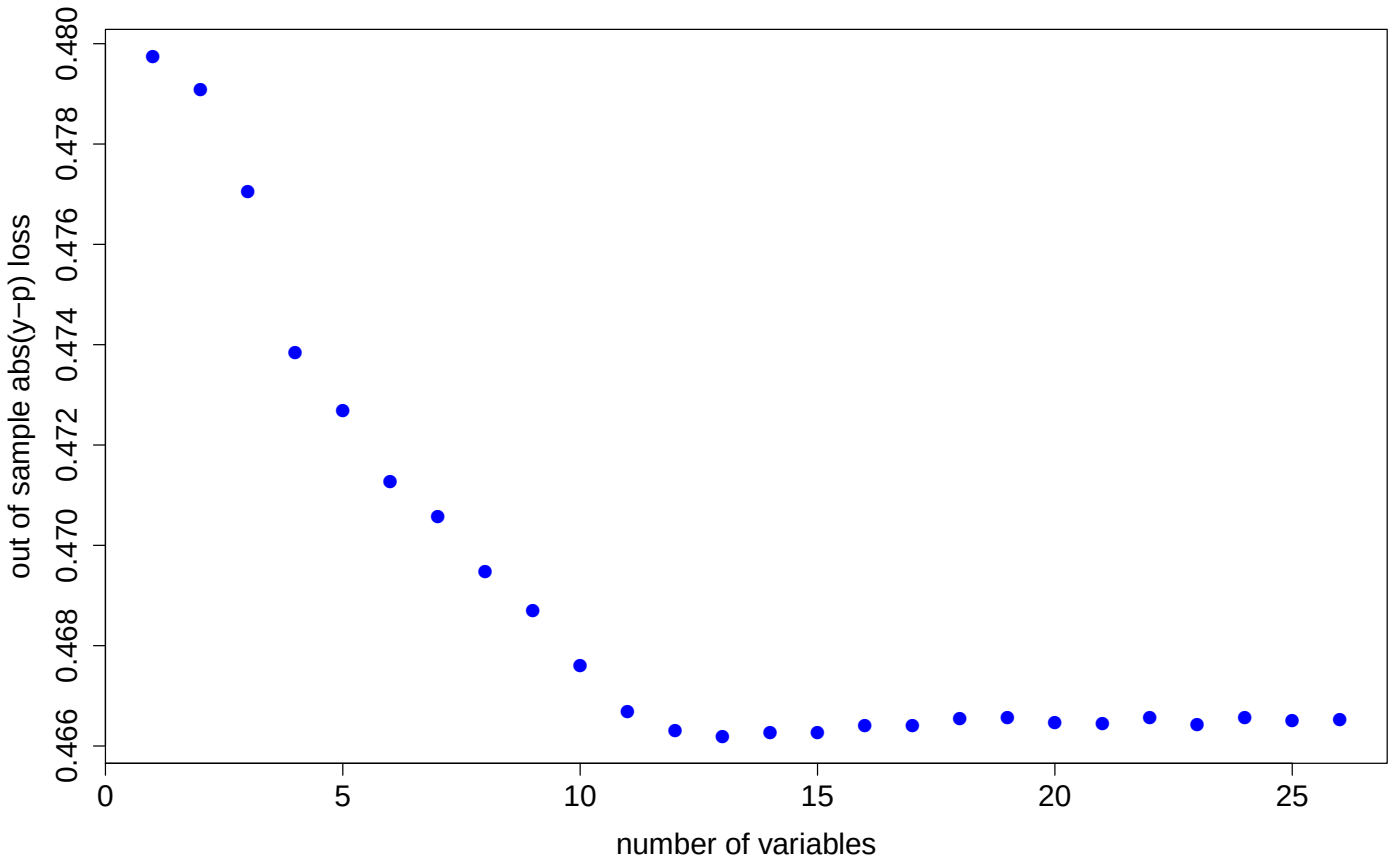
We have a test data set of size 10,000 to asses the out-of-sample preditive performance of models fit using the chosen variable subsets.

Note:

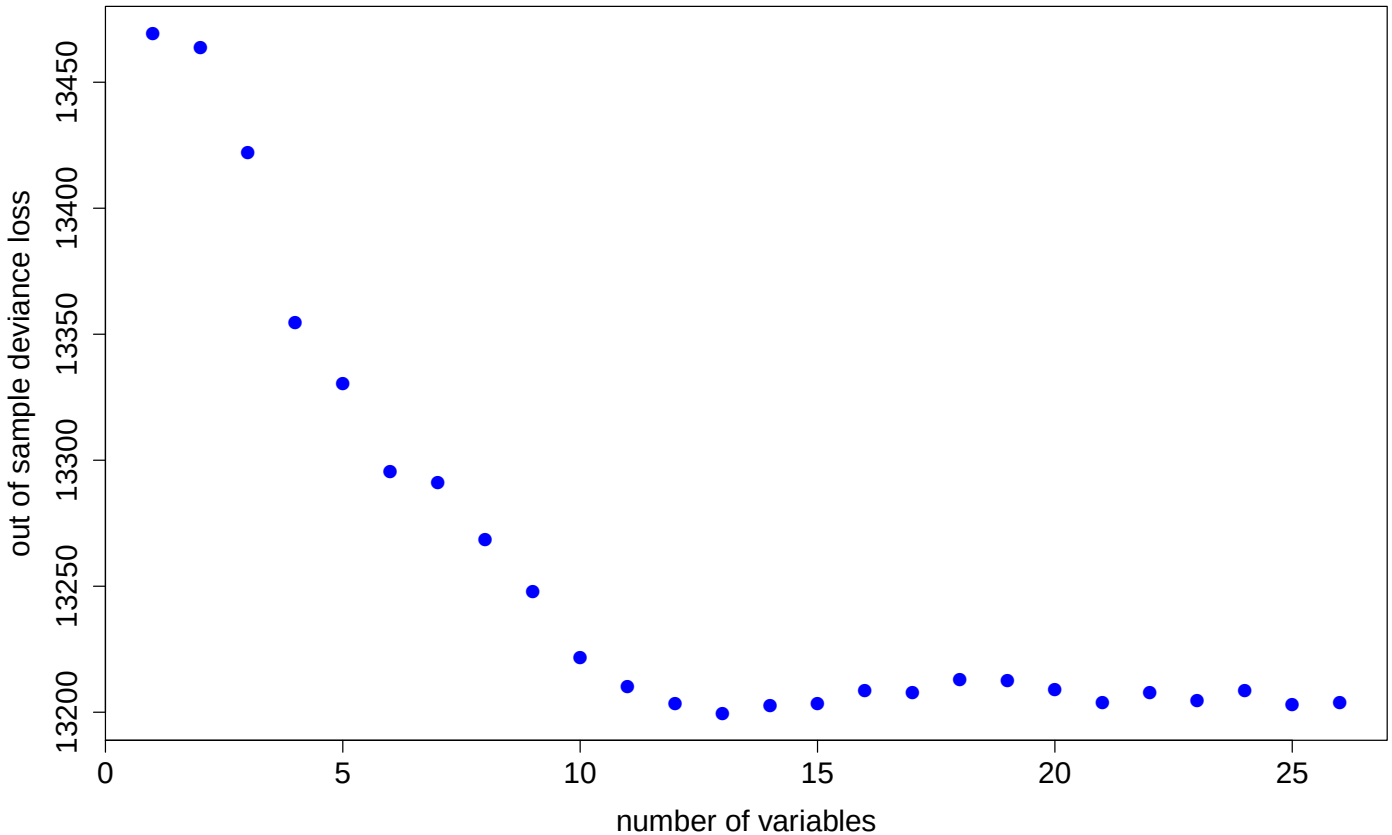
In principle, our approach has nothing to do with "regularization" or *the bias-variance tradeoff*.

But of course, choosing a good simple model helps everything!!!!

oos-abs-bartonsubsets



oos-deviance-bartonsubsets



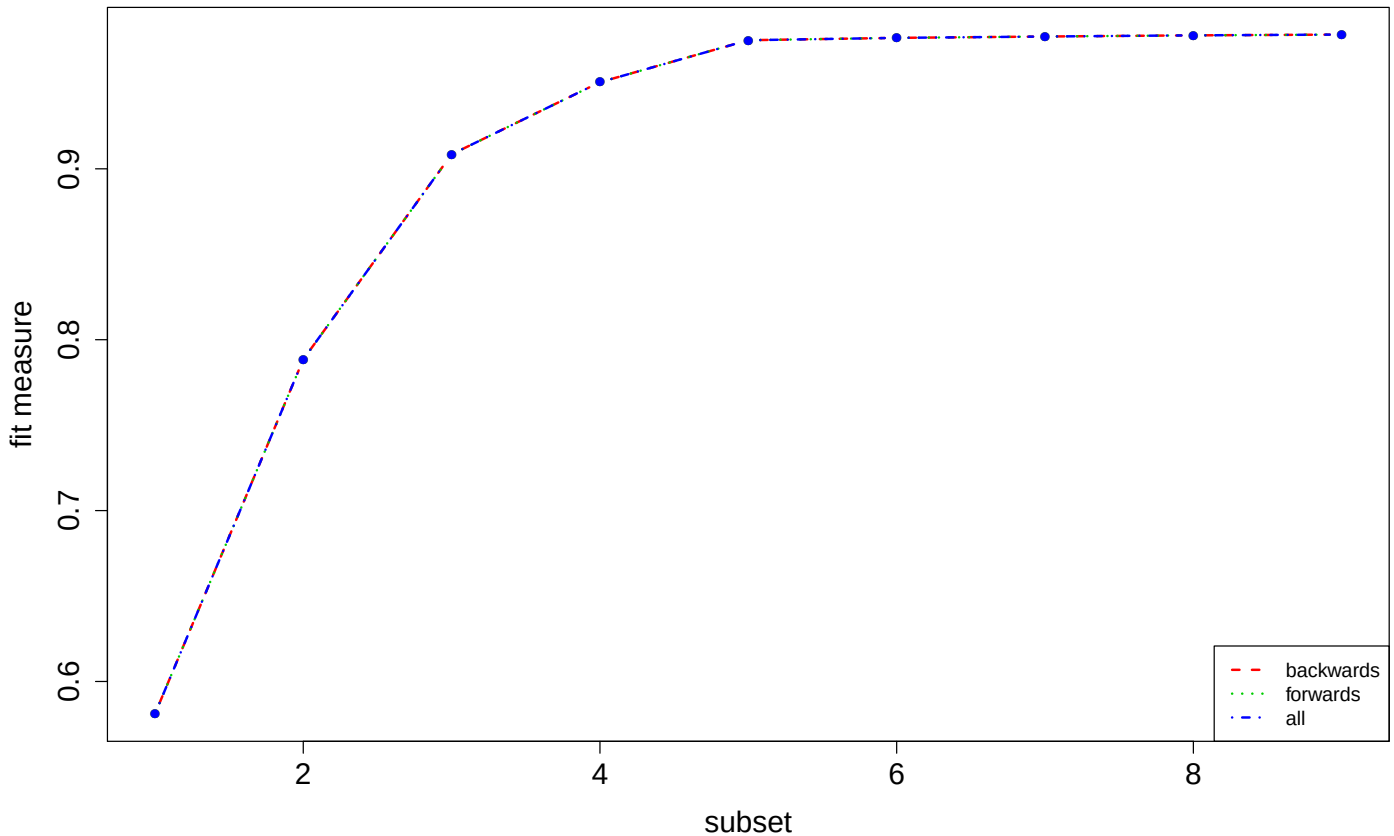
5. Friedman

```
f = function(x){
10*sin(pi*x[,1]*x[,2]) + 20*(x[,3]-.5)^2+10*x[,4]+5*x[,5]
}

sigma = 1.0  #y = f(x) + sigma*z , z~N(0,1)
n = 100      #number of observations
set.seed(99)
X=matrix(runif(n*10),n,10) #10 variables, only first 5 matter
if(0) {
   np=5000
   xp=matrix(runif(np*10),np,10) #10 variables, only first 5 matter
} else {
   np=n
   xp=X
}
Ey = f(X)
y=Ey+sigma*rnorm(n)
```

We will use either 5,000 or 100 x in X^f.

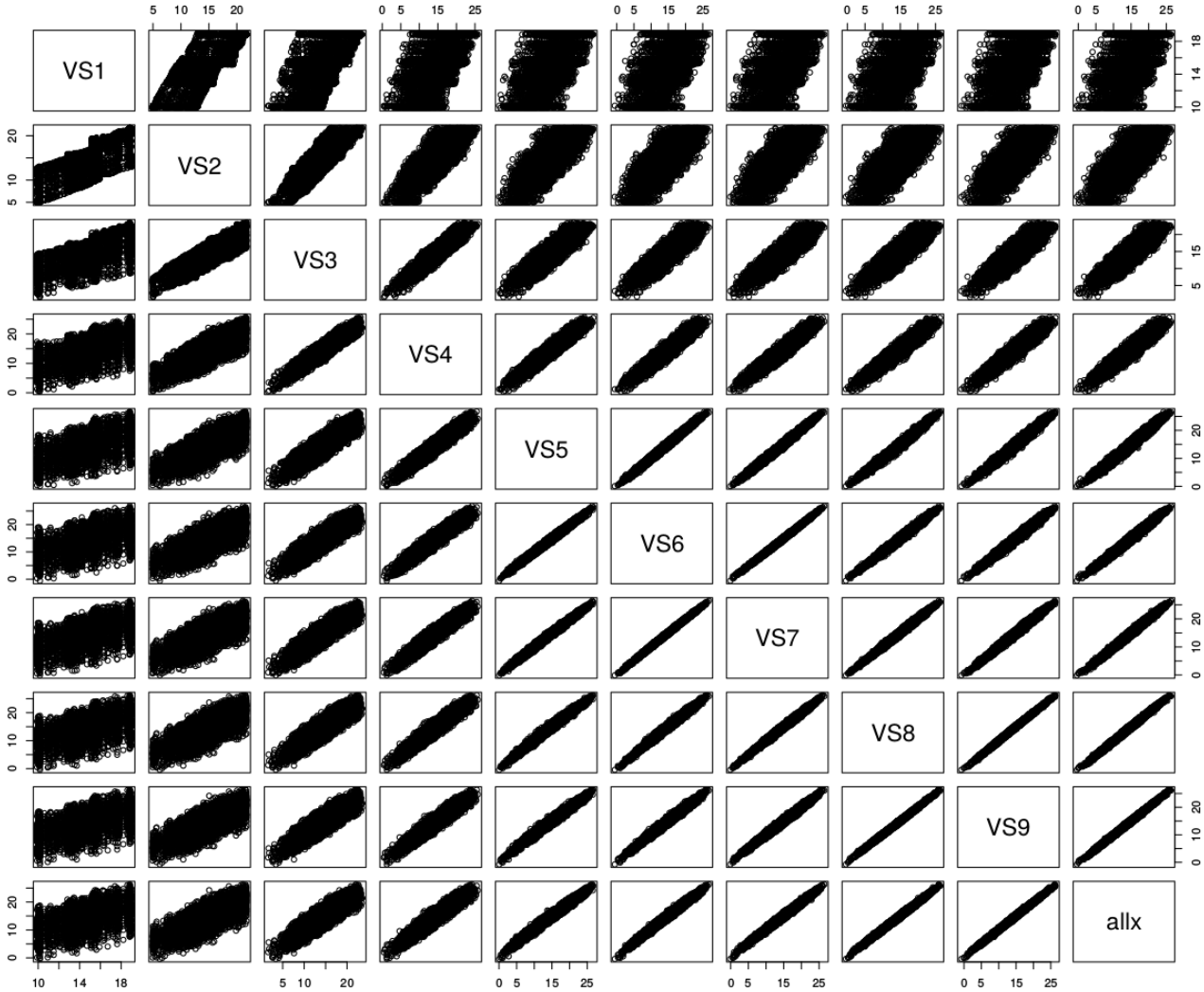
compare all, friedman, big x.



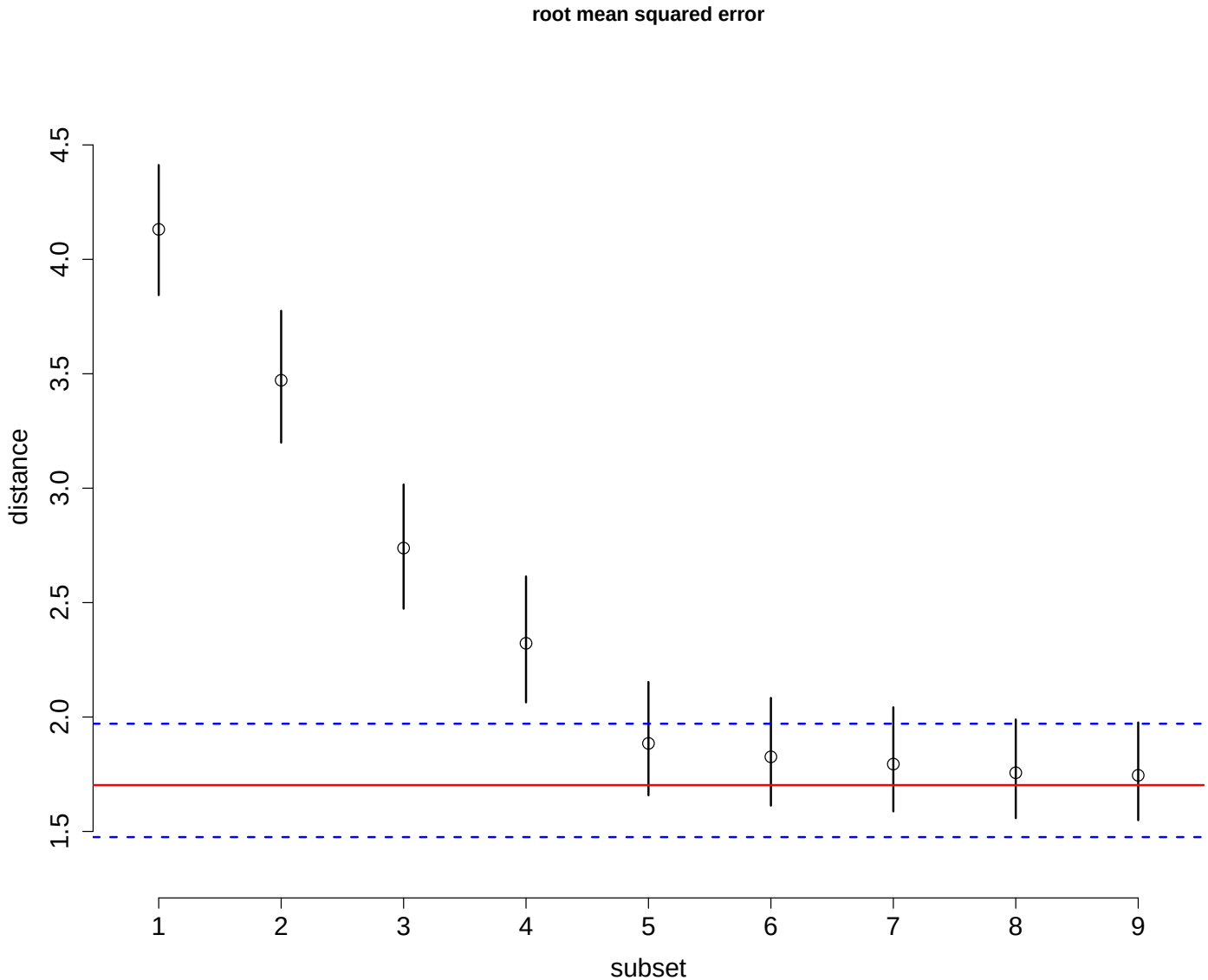
friedman big x example. Forwards and backwards subsets.

numvar	forwards	backwards
1	x4	x4
2	x2	x2
3	x1	x1
4	x5	x5
5	x3	x3
6	x6	x6
7	x8	x8
8	x9	x9
9	x7	x7
10	x10	x10

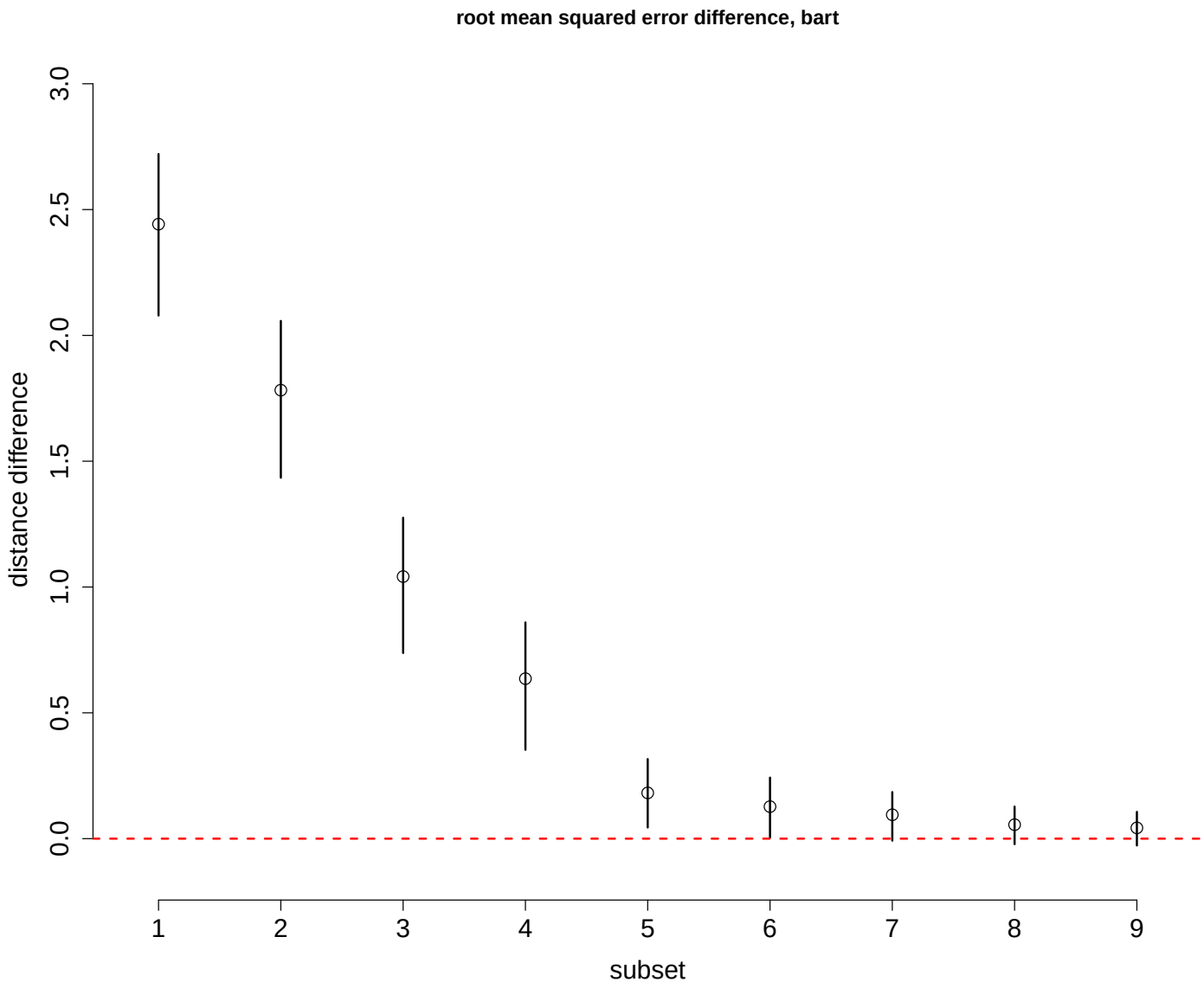
pairs bartsubs, allx, friedman, big x.



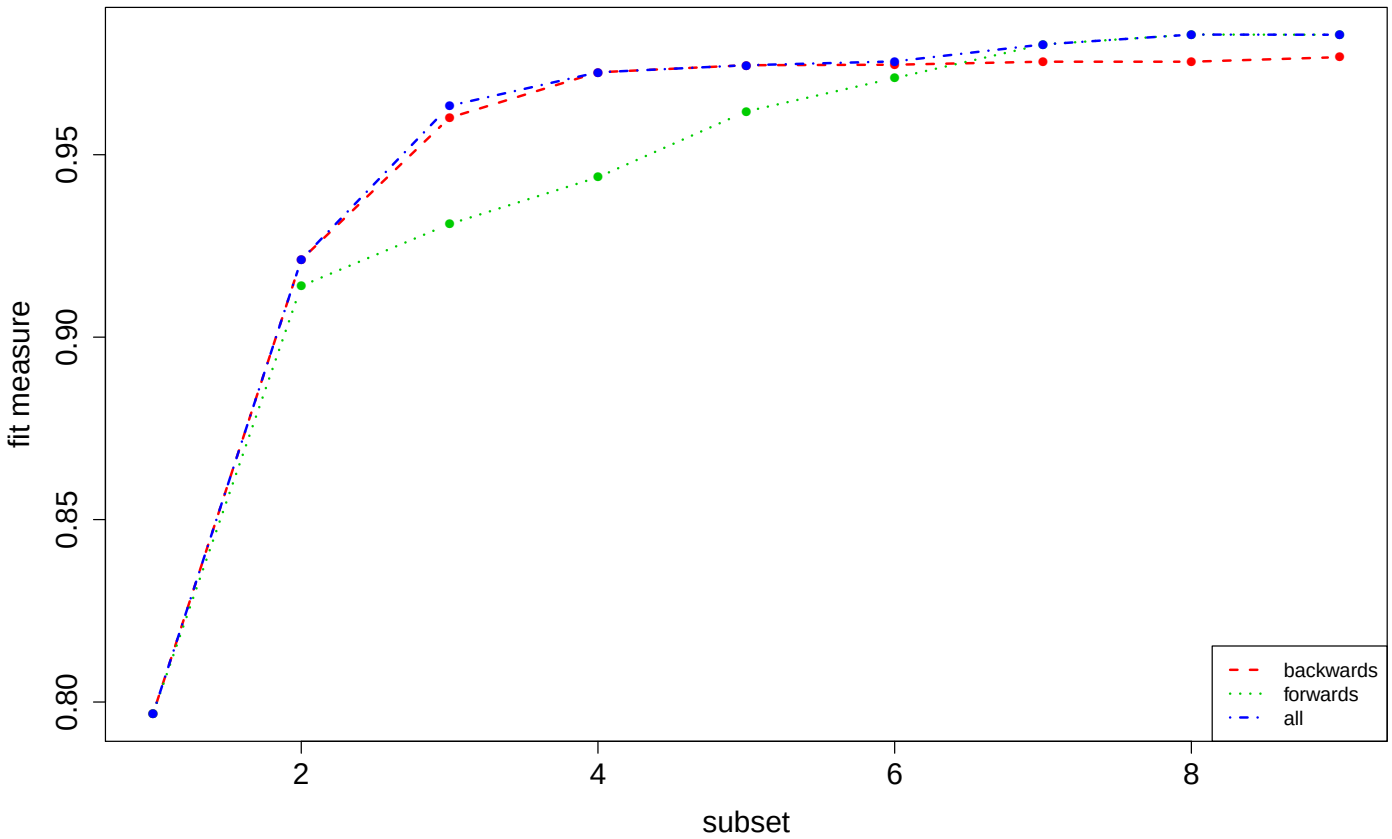
Friedman example. First rmse distance.



Friedman example. rmse distance, difference.



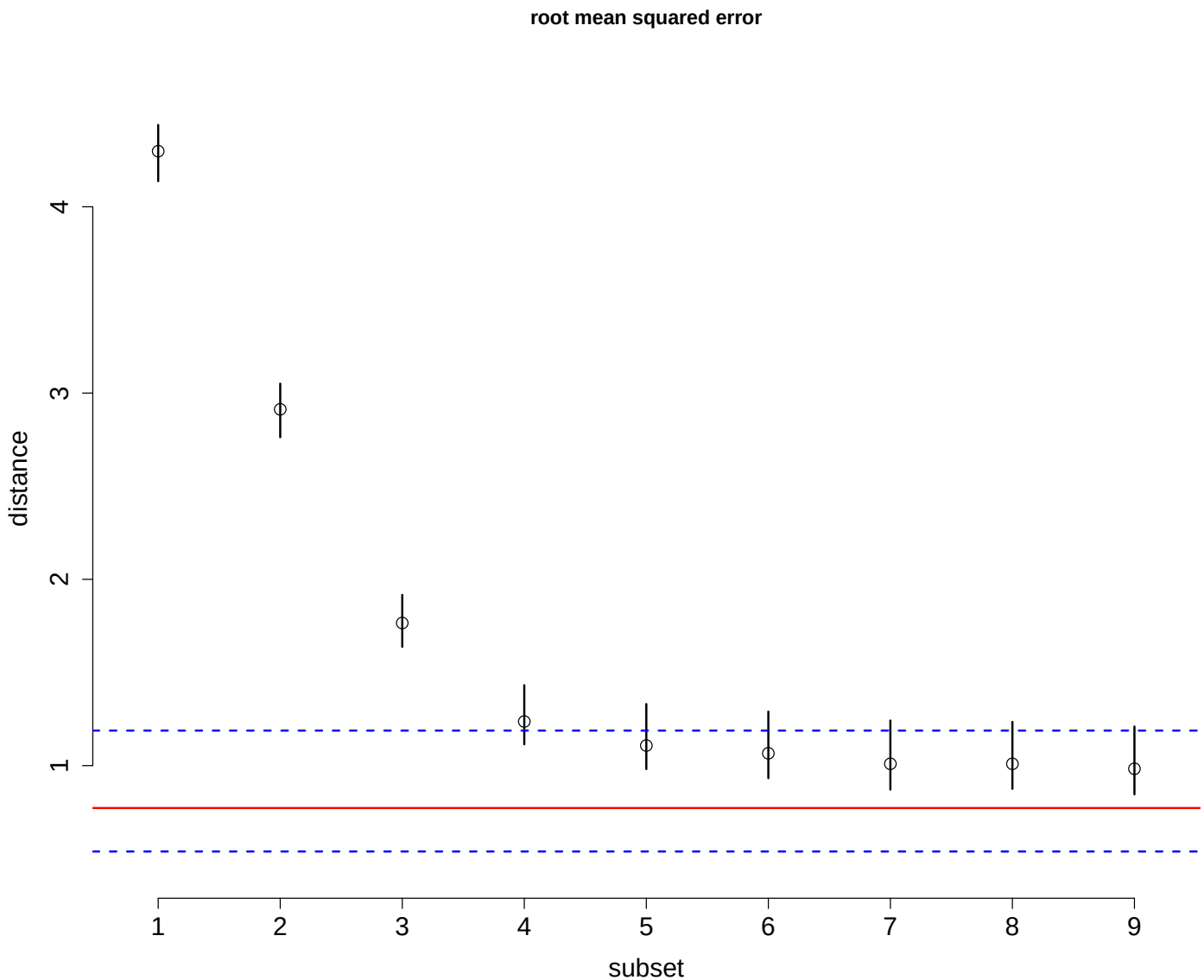
compare all, friedman, small x.



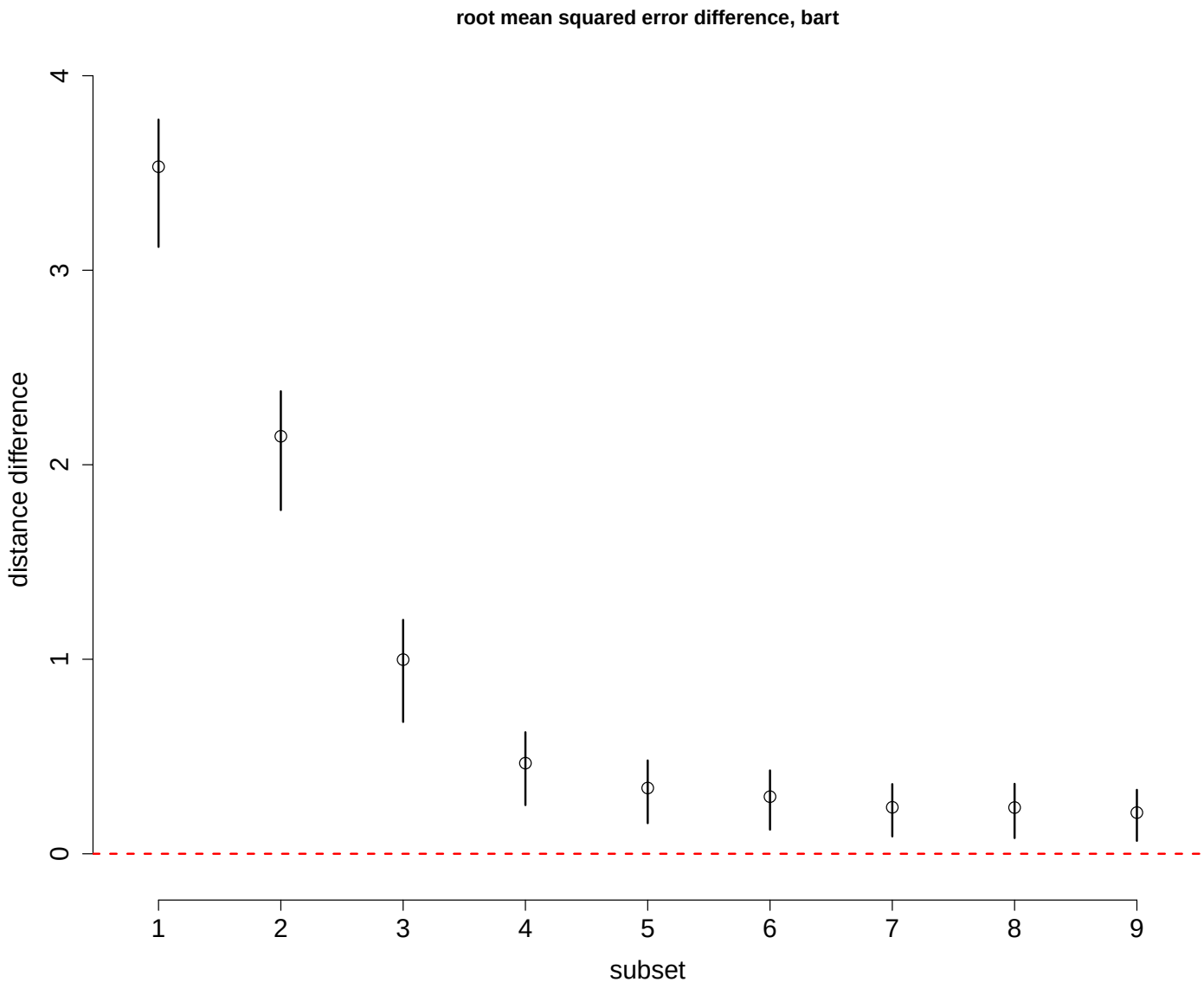
friedman small x example. Forwards and backwards subsets.

numvar	forwards	backwards
1	x2	x2
2	x4	x1
3	x1	x5
4	x3	x7
5	x6	x9
6	x8	x3
7	x10	x6
8	x9	x10
9	x7	x8
10	x5	x4

Friedman example, small x. First rmse distance.



Friedman example difference in rmse, small x. First rmse distance.



6. Utility Based (fit-the-fit) Variable Selection

Claim:

We can make the "fit-the-fit" tie more closely to an underlying utility approach.

We should be maximizing the expected utility or minimizing the expected loss:

For example:

$$L(Y, \gamma_S) = (Y - \gamma_s(x^f))^2 + \lambda|S|$$

We take the expectation using the predictive of $Y|x$ and then average this over our set of future x.

$$L(Y, \gamma_S) = (Y - \gamma_s(x^f))^2 + \lambda|S|$$

But since,

$$E(Y - \gamma_s(x^f))^2 = Var(Y) + (E(Y|x) - \gamma_s(x^f))^2$$

with squared error loss, our expected predictive utility just fits the means as discussed above.

And then minimizing for each $|S|$ is a way to minimize given the $|S|$ complexity penalty.

Key: λ is a utility parameter, not a prior (or estimation penalty) parameter !!!

If you have a sparsity prior, you can put it in Step 1.

7. Conclusion

The simple forward step-wise search seems to work really well. This can be simply parallelized.

Refitting the step 1 type model on the identified subsets can be done in parallel and is relatively easy once you have p possible subsets instead of 2^p.

Fitting a big tree is clearly a basic tool in data analysis (e.g. Random Forests).

Win-Win

The Bayesian decision environment has prior, model, and utility.

Most variable selection work emphasizes prior, and computes the posterior over model choices. This is a nightmare.

Emphasizing utility is the right thing to do *and it is easier !!!*

Redemption

Re George and McCulloch, "Stochastic Search Variable Selection".

Jay Kadane:

"You are confusing your prior with your utility function."

Not any more!!!!