Inspect how priors on $p/se(p)$ affect estimating a variant's effect by simulation

Xulong Wang

January 18, 2016

Background

Logistic model: $logit(p) \sim p * genotype$
Response: case and control in 0/1
Predictor: numerical genotypes in 0/1/2

Model and data

```
rm(list = ls())
setwd("~/GitHub/byglmm")

N = 1e2

set.seed(3)
g = sample(0:2, N, replace = T)
y = sample(0:1, N, replace = T)
cor(g, y)
```

```
## [1] -0.1480029
```

```
mod = stan_model("./Stan/sim1.stan")
dat = list(N = N, y = y, g = g, prior = 0)

mod
```

```
## S4 class stanmodel 'sim1' coded as follows:
## data {
##   int<lower=1> N;  // Sample
##   int<lower=0,upper=1> y[N];  // response
##   vector[N] g;  // genotype
##   real prior;
## }
##
## parameters {
##   real p;  // variant effect
##   real t;
##   real<lower=0.01> sigma;
## }
##
## transformed parameters {
##   real mu;
##   mu <- t * sigma;
## }
##
## model {
##   p ~ normal(mu, sigma);
##   t ~ normal(prior, 1);
##   sigma ~ inv_gamma(2, 1);
##
##
##   for (n in 1:N)
##     y[n] ~ bernoulli_logit(p * g[n]);
## }
##
```

```
dat
```

```
## $N
## [1] 100
##
## $y
##   [1] 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 0 0 0 0 0 1
##  [36] 1 0 0 1 1 0 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1
##  [71] 0 0 1 1 0 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0
##
## $g
##   [1] 0 2 1 0 1 1 0 0 1 1 1 1 1 1 2 2 0 2 2 0 0 0 0 0 0 2 1 2 1 2 1 1 0 1 0
##  [36] 1 2 0 1 0 0 2 0 1 1 0 0 0 0 2 0 0 2 2 2 2 1 0 0 0 2 0 2 0 2 0 2 1 1 0
##  [71] 2 2 2 2 1 1 0 0 2 2 2 1 1 0 2 2 0 1 1 2 0 0 1 2 2 0 0 0 0 2
##
## $prior
## [1] 0
```

Fitting model with a flat prior: $p/se(p) = 0$

```
fit = optimizing(mod, hessian = T, algorithm = "LBFGS", data = dat)
```

```
## STAN OPTIMIZATION COMMAND (LBFGS)
## init = random
## save_iterations = 1
## init_alpha = 0.001
## tol_obj = 1e-12
## tol_grad = 1e-08
## tol_param = 1e-08
## tol_rel_obj = 10000
## tol_rel_grad = 1e+07
## history_size = 5
## seed = 1570880141
## initial log joint probability = -114.229
## Optimization terminated normally:
##   Convergence detected: relative gradient magnitude is below tolerance
```

```
se = sqrt(diag(solve(-fit$hessian)))["p"]
(x = c(fit$par, se = se))
```

```
##           p           t       sigma          mu        se.p 
## -0.19767735 -0.36851588  0.26822980 -0.09884694  0.15268371
```

Fitting model with testing priors

```
t1 = seq(-15, 15, 0.1)
```

Ignoring codes for optimizing with t1.

```
(test = as.data.frame(t(test))) %>% head
```

```
##            p         t      sigma         mu      se.p
## 1 -0.7688617 -14.00694 0.05908185 -0.8275560 0.1197532
## 2 -0.7665302 -13.90520 0.05937309 -0.8255947 0.1197358
## 3 -0.7641984 -13.80372 0.05966815 -0.8236427 0.1197184
## 4 -0.7618421 -13.70203 0.05996692 -0.8216684 0.1197014
## 5 -0.7594879 -13.60073 0.06026978 -0.8197133 0.1196825
## 6 -0.7570975 -13.49920 0.06057624 -0.8177304 0.1196607
```

Ouputs

Absolutes of $p/se(p)$ were used for significance quantity.

Abbreviations of statistics:

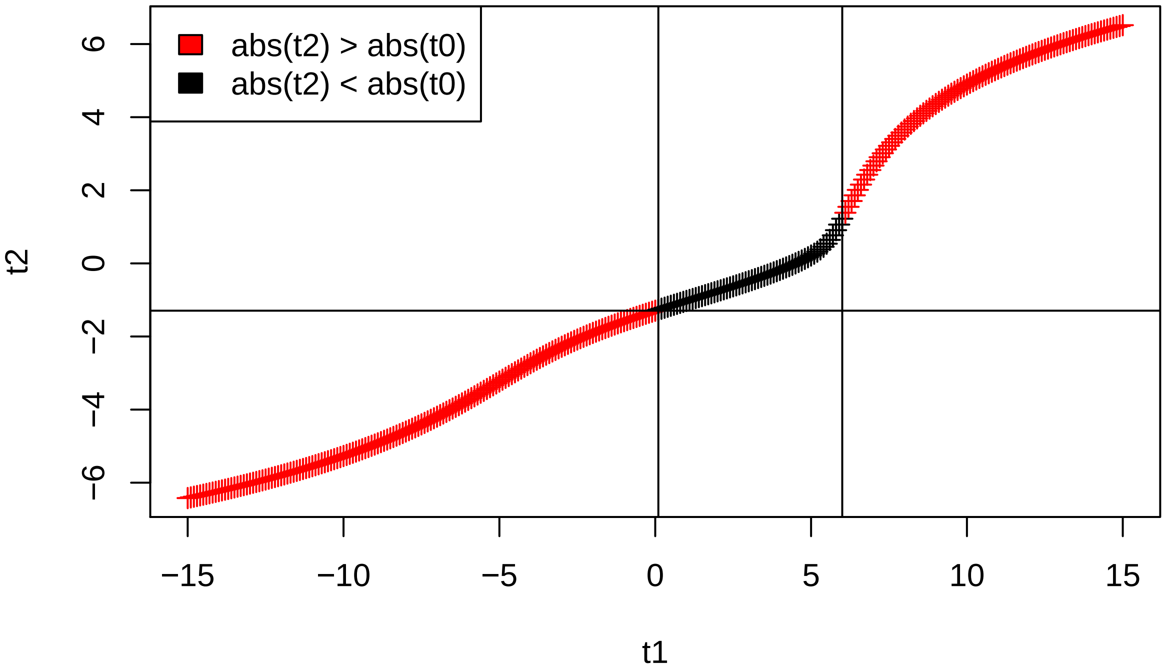
1. $t0$ is the estiamted $p/se(p)$ with a flat prior

2. $t1$ is the prior $p/se(p)$
3. $t2$ is the estimated $p/se(p)$ with $t1$ priors

```
t0 = x[1] / x[5]
t2 = test$p / test$se.p
```

Estimates with flat and $t1$ priors: t0 vs t2

```
plot(t1, t2, pch = 3, col = as.numeric(abs(t2) > abs(t0)) + 1)
abline(v = max(t1[abs(t2) < abs(t0)]))
abline(v = min(t1[abs(t2) < abs(t0)]))
legend("topleft", c("abs(t2) > abs(t0)", "abs(t2) < abs(t0)"), fill = c("red", "black"))
abline(h = t0)
```

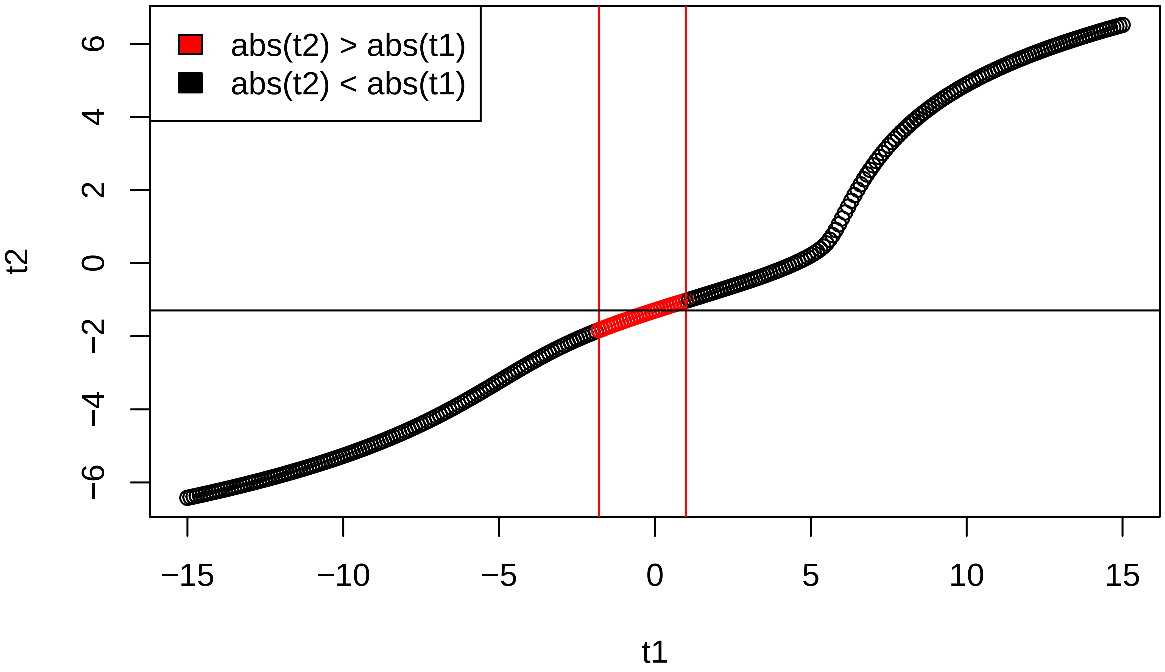


1. Posterior p-values improved when prior $t1$ was at the same direction of $t0$
2. Posterior p-values deteoriated when prior $t1$ was contradict with $t0$
3. Extreme prior $t1$ can dominate the data

Prior and post: t1 vs t2

Red: abs(t2) > abs(t1) Blk: abs(t2) < abs(t2)

```
plot(t1, t2, col = as.numeric(abs(t2) > abs(t1)) + 1)
abline(v = max(t1[abs(t2) > abs(t1)]), col = "red")
abline(v = min(t1[abs(t2) > abs(t1)]), col = "red")
legend("topleft", c("abs(t2) > abs(t1)", "abs(t2) < abs(t1)"), fill = c("red", "black"))
abline(h = t0)
```



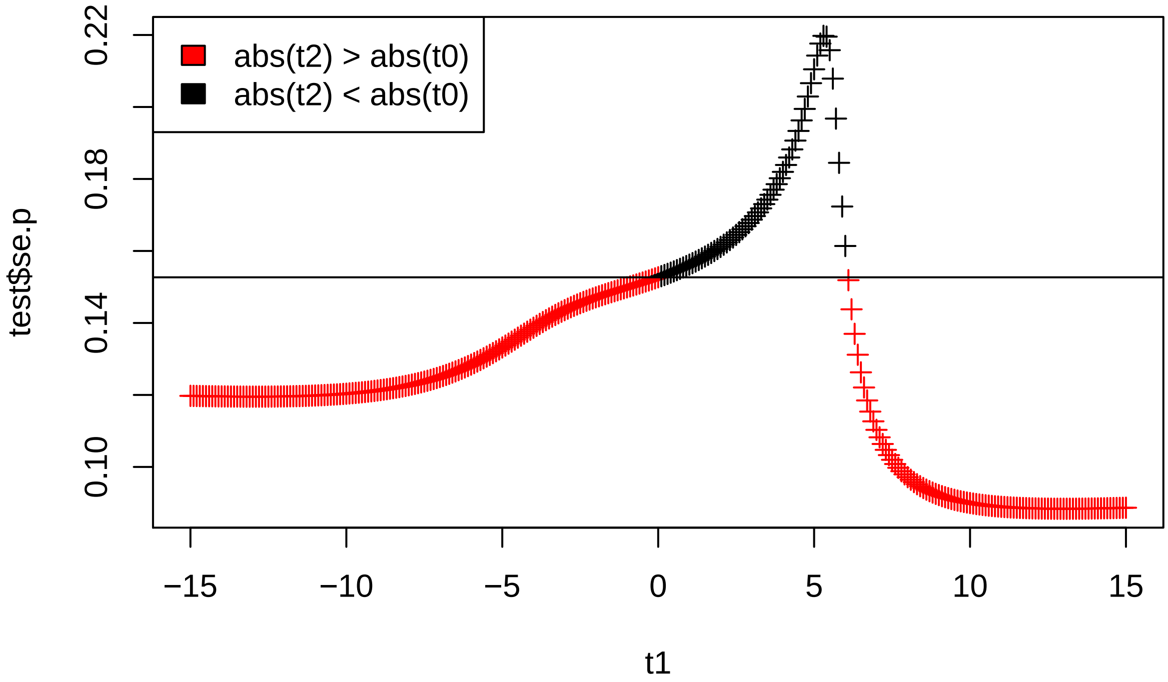
```
# t1[abs(t2) > abs(t1)]
```

$abs(t2) > abs(t1)$ happened with weaker priors, where data dominated the estimations.

Took the last two graphs together, we saw situations where $abs(t2)$ was smaller than both $abs(t0)$ and $abs(t1)$, and it happend when the $t1$ was modestly contradict with $t0$.

Graph below gave the estimated $se(p)$ versus prior $t1$. Red was abs(t2) $>$ abs(t0). Blk was abs(t2) $<$ abs(t0). Standard error was high when evidences from prior and data were comparable and contradicting.

```
plot(t1, test$se.p, pch = 3, col = as.numeric(abs(t2) > abs(t0)) + 1)
legend("topleft", c("abs(t2) > abs(t0)", "abs(t2) < abs(t0)"), fill = c("red", "black"))
abline(h = x["se.p"])
```



abs(t2) > abs(t0)
abs(t2) < abs(t0)
test$se.p
0.10
0.14
0.18
0.22
−15
−10
−5
0
5
10
15
t1