Convergence rate examples and theory

Simen Gaure

Abstract. If you use **lfe** for various tasks, you will notice that some estimations converge fast, whereas others converge slowly. Convergence rate of the methods used by **lfe** is not a walk in the park. Here are some examples.

1. Introduction

The method employed by **lfe** is *the method of alternating projections* ([**7**]). The link to this method comes from viewing demeaning of a vector as a *projection* ([**4**]), i.e. a linear operator P with the property $P^2 = P = P^t$. Also, the Kaczmarz-method used by **lfe** to solve the sparse resulting system is a variant of alternating projections, though P is then an affine projection with $P^2 = P$, not a linear operator. That is, if the projection demeaning factor i is P_i, their intersection which demeans all factors is $\lim_{n\to\infty}(P_1 P_2 \cdots P_e)^n$. We can therefore iterate the operator $T = P_1 P_2 \cdots P_e$ on a vector x: $T^n x$, to demean x.

The convergence rate has been analysed in [**3**] and references cited therein; there are also newer elaborations, e.g. in [**1**]. The convergence rate theory is in terms of generalized angles between subspaces, i.e. the ranges of the projections. The angle concept may not be immediately intuitive to practitioners of linear regression methods, so let's have a look at some examples.

2. Examples

Our first example has two factors, they are independent of each other, and of quite high cardinality

```
library(lfe)
set.seed(42)
x <- rnorm(100000)
f1 <- sample(10000, length(x), replace = TRUE)
f2 <- sample(10000, length(x), replace = TRUE)
y <- x + cos(f1) + log(f2 + 1) + rnorm(length(x), sd = 0.5)
```

We time the first step:

```
system.time(est <- felm(y ~ x | f1 + f2))
##    user  system elapsed
##   0.080   0.089   0.080
```

and the second step

```
system.time(alpha <- getfe(est))
##    user  system elapsed
##   0.054   0.000   0.053
```

We see that there's nothing to complain about in terms of speed. After all, there are 20,000 dummies in this model.

Now we let `f2` have fewer levels:

```
f2 <- sample(300, length(x), replace = TRUE)
y <- x + cos(f1) + log(f2 + 1) + rnorm(length(x), sd = 0.5)
system.time(est <- felm(y ~ x | f1 + f2))
##    user  system elapsed
##   0.069   0.075   0.061
system.time(alpha <- getfe(est))
##    user  system elapsed
##   0.036   0.000   0.036
```

Not much happens, the second step is apparently faster, whereas the first is about the same. Note that much of the time in the first step is spent in mundane bookkeeping such as creating a model matrix, not in the demeaning as such.

Now we make the fixed effects dependent. We do this by ensuring that for each value of `f1` there are at most 10 different values of `f2`. We keep the number of levels in `f2` at 300, as in the previous example. The size of this problem is exactly the same as in the previous example, but the factor structure is very different.

```
f2 <- (f1 + sample(10, length(x), replace = TRUE)) %% 300
y <- x + cos(f1) + log(f2 + 1) + rnorm(length(x), sd = 0.5)
system.time(est <- felm(y ~ x | f1 + f2))
##    user  system elapsed
##   1.161   0.076   1.158
system.time(alpha <- getfe(est))
##    user  system elapsed
##   0.298   0.000   0.300
```

The estimation now takes a lot more time. Indeed, in this case, it is worthwhile to consider coding `f2` as 300 dummies, i.e. as an ordinary factor, even though there now are ≈ 300 vectors to centre, as opposed to two in the previous examples.

```
system.time(est <- felm(y ~ x + factor(f2) | f1))
##    user  system elapsed
##   1.705   1.785   1.106
system.time(alpha <- getfe(est))
##    user  system elapsed
##   0.025   0.000   0.027
```

We could be tempted to believe that this is the whole story, but it's not. We may create another example, where each `f1` only has 10 different `f2`'s as above, but where the overlap is different. In the above example, an observation with `f1=1` will have `f2` drawn from $\{2,3,4,\ldots,11\}$, whereas an observation with `f1=2` will have `f2` in {3,4,5,. . . ,12} and so on, i.e. a considerable overlap. We reduce this overlap by creating `f2` with a little twist, by introducing some unevenly sized “holes” by cubing the samples of `1:10`.

```
f2 <- (f1 + sample(10, length(x), replace = TRUE)^3) %% 300
y <- x + cos(f1) + log(f2 + 1) + rnorm(length(x), sd = 0.5)
system.time(est <- felm(y ~ x | f1 + f2))
##    user  system elapsed
##   0.127   0.082   0.121
system.time(alpha <- getfe(est))
##    user  system elapsed
##   0.037   0.000   0.038
nlevels(est[["cfactor"]])
## [1] 1
```

We are now back at approximately the same fast convergence as before. That is, convergence rate is not a simple function of data size, counts of levels and so on, even when we have only two factors.

2.1. Why?

In the examples above we have played with the structure of the bipartite graph mentioned in [**4**], indicating that convergence rate is not a function solely of the average degree and number of connected components of the graph. Intuitively (to the author), the graph determines the convergence rate, at least in the operator norm topology, but it is unclear to me whether the rate can be described in simple graph theoretic terms, let alone in terms of some intuitive relations between dummy variables.

One could speculate that convergence is related to how tightly connected the graph is, one such measure is the diameter of the graph, or the set of lengths of shortest paths.

```
library(igraph)
##
## Attaching package:  'igraph'
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
## The following object is masked from 'package:base':
##
##     union
mkgraph <- function(f1, f2) {
  graph.edgelist(cbind(paste("f1", f1), paste("f2", f2)), directed = FALSE)
}

appxdiam <- function(g) max(shortest.paths(g, sample(V(g), 10), sample(V(g), 10)))
f2 <- sample(10000, length(x), replace = TRUE)
appxdiam(mkgraph(f1, f2))
## Warning:  ‘shortest.paths()‘ was deprecated in igraph 2.0.0.
## i Please use ‘distances()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning
was
## generated.
## Warning:  ‘graph.edgelist()‘ was deprecated in igraph 2.0.0.
## i Please use ‘graph_from_edgelist()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning
was
## generated.

f2 <- (f1 + sample(5, length(x), replace = TRUE)^3) %% 300
appxdiam(mkgraph(f1, f2))

f2 <- (f1 + sample(5, length(x), replace = TRUE)) %% 300
appxdiam(mkgraph(f1, f2))
```

Loosely speaking, data with “six degrees of separation” seems to converge faster than less well connected networks. Informal trials, using `demeanlist` instead of `felm` to avoid overhead, show that convergence time grows roughly as the square of the diameter, also in more varied graphs than the above. However, the author has not had the time to attempt to prove such an assertion.

A special case of this is when the second factor is an interaction of several factors in such a way that there is multicollinearity by design. This does not introduce any bias problems, but the speed can often be dramatically improved by removing the known multicollinearities in advance by collapsing appropriate levels to one. This will diminish the diameter of the graph.

Note that convergence rate in practice is also a function of the vector to be centred, this can be seen from the trivial example when the vector is already centred and convergence is attained after a single iteration.

It is a sad fact of life, shown in [**3**], that when there are more than two factors, the convergence rate can not be determined solely by considering the angles of *pairs* of subspaces. In [**1**] a generalized angle between a finite set of subspaces is introduced for the purpose of analysis. It is left to the reader to imagine some simple examples with more than two factors. A real example with three factors can be found in [**10**], and real examples with up to five factors are treated in the appendix of [**4**] as well as in [**8**].

3. Acceleration schemes

One point to note is that with more than two factors, even their *order* may be important for the convergence rate, this has been utilized to speed up convergence in some cases, picking a random order of the projections for each iteration, or sampling randomly from the projections. One such scheme is used in [**9**], and randomization is also treated in [**1**] and the papers they cite. Random shuffling of the equations is now used in the Kaczmarz-step of **lfe**, i.e. in `getfe`, as suggested in [**4**].

Another acceleration scheme is found in [**5**], where a line search method is used in each iteration. I.e. when a single iteration transforms our vector $\eta \mapsto \eta'$, it is fairly easy to find the exact point on the line $\{\eta + t(\eta' - \eta)\ :\ t \in \mathbb{R}\}$ which is closest to the solution. **lfe** uses this line search method in `felm` (or rather, in `demeanlist`). However, it is noted in [**2**] that this method does not always accelerate convergence.

After publishing [**4**], the author was made aware that exactly the same method of alternating projections was arrived at in [**6**, p. 637], though as a technical simplification of the Gauss-Seidel method of iterated regressions. It is possible that acceleration schemes for the Gauss-Seidel method may be applicable, though the author has not yet looked into it.

References

[1] C. Badea, S. Grivaux, and V. Müller, *The rate of convergence in the method of alternating projections*, St. Petersburg Mathematical Journal **23** (2012), 413–434, preprint in arXiv:1006.2047.

[2] Heinz H. Bauschke, Frank Deutsch, Hein Hundal, and Sung-Ho Park, *Accelerating the convergence of the method of alternating projections*, Trans. Amer. Math. Soc. **355** (2003), no. 9, 3433–3461.

[3] F. Deutsch and H. Hundal, *The Rate of Convergence for the Method of Alternating Projections, II*, J. Math. Anal. App. **205** (1997), no. 2, 381–405.

[4] S. Gaure, *OLS with Multiple High Dimensional Category Variables*, Computational Statistics and Data Analysis **66** (2013), 8–18.

[5] W.B. Gearhart and M. Koshy, *Acceleration schemes for the method of alternating projections*, Journal of Computational and Applied Mathematics **26** (1989), no. 3, 235–249.

[6] P. Guimarães and P. Portugal, *A simple feasible procedure to fit models with high-dimensional fixed effects*, Stata Journal **10** (2010), no. 4, 628–649(22).

[7] I. Halperin, *The Product of Projection Operators*, Acta Sci. Math. (Szeged) **23** (1962), 96–99.

[8] S. Markussen and K. Røed, *Social insurance networks*, IZA Discussion Papers 6446, Institute for the Study of Labor (IZA), March 2012.

[9] T. Strohmer and R. Vershynin, *A Randomized Kaczmarz Algorithm with Exponential Convergence*, Journal of Fourier Analysis and Applications **15** (2009), 262–278.

[10] S.M. Torres, P. Portugal, J.T. Addison, and P. Guimarães, *The Sources of Wage Variation: A Three-Way High-Dimensional Fixed Effects Regression Model.*, IZA Discussion Paper 7276, Institute for the Study of Labor (IZA), March 2013.

Ragnar Frisch Centre for Economic Research, Oslo, Norway