Finding maximum correlations

Nathanael Fillmore
Computer Sciences Department
University of Wisconsin-Madison
nathanae@cs.wisc.edu

June 6, 2009—DRAFT

Abstract

An efficient algorithm is presented to find the most highly correlated pair of variables in a dataset. The expected number of correlations that the algorithm needs to compute is shown to be linear in the number of variables rather than quadratic.

1 Introduction

For several tasks in machine learning it is useful to know which two (or few[1]) variables in a dataset are most highly correlated. Example #1: The treelet algorithm [1] repeatedly merges the two most highly correlated variables in a dataset using a Jacobi rotation; the result is an analog of the wavelet transform, but for unordered data. Example #2: A robot (or other learner) may have a large number of tasks it wants to learn to perform and a large number of possible variables it can observe. However, observing each variable has some cost, so it is desirable not to observe superfluous variables in the field. One approach to determining which variables to select is to choose those most highly correlated with the targets (e.g., [2], §2, and references therein).

A straightforward method for finding the most highly correlated pair of variables in a dataset is to compute the correlation between each pair of variables and keep track of which pair's correlation is largest. For n observations of p variables, this approach requires $O(np^2)$ time, since computing the correlation between two variables requires $O(n)$ time, and there are $p(p-1)/2$ pairs.

For small datasets, with a small number of variables, this approach is adequate. However, for many modern datasets, $O(np^2)$ time is a burden, especially since finding the most highly correlated pair is typically only a subroutine used to help solve another problem. As an example of the scale that may be required, we mention the Web 1T 5-Gram corpus [3], which is generated from $p = 13,588,391$ word types and $n = 95,119,665,584$ sentences. Even in more moderate cases, the complexity of the straightforward method may be onerous.

In this paper, we present an new algorithm to find the most highly correlated pair of variables in a dataset (§4). The algorithm relies on a geometrically motivated bound to avoid computing correlations for many variable pairs that will not be most highly correlated. To our knowledge, this

[1] In the remainder of the paper, we talk only about the most highly correlated pair of variables, but all algorithms discused can be used to find the top k pairs by keeping track of the k most highly correlated pairs of variables instead of the single most highly correlated pair.

is the first such algorithm in the literature, although similar geometrical arguments have been used in other domains (e.g., [4], ch. 24). We show that under appropriate conditions the algorithm can be expected to require computation of the correlation between only $\Theta(p)$ pairs of variables (§5). First, though, we briefly review notation (§2) and consider an instructive special case (§3).

2 Problem formulation and notation

We consider p input variables $X_1, X_2, \ldots, X_p$. We assume for convenience that all variables are quantitative and are centered on zero, but neither assumption is necessary.

We are given n observations of each variable, and we array the n observations of each variable X_i in an n-dimensional vector $x_i := [x_{1i}\ x_{2i}\ \cdots\ x_{ni}]^T$.

Both p and n may be very large.

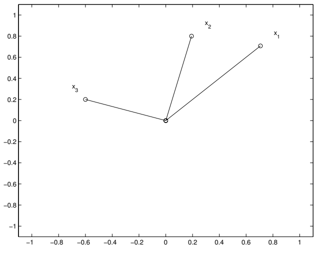
For the zero-centered variables X_i and X_j, the sample correlation is defined as

$$\rho_{ij} := \frac{x_i^T x_j}{\sqrt{x_i^T x_i}\sqrt{x_j^T x_j}}.$$

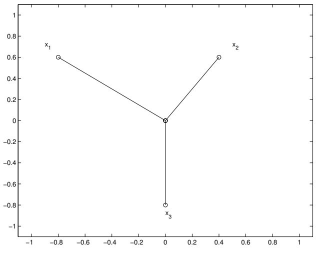
The task is to find the pair of distinct variables (X_i, X_j) with greatest sample correlation ρ_{ij}.

3 Warm-up

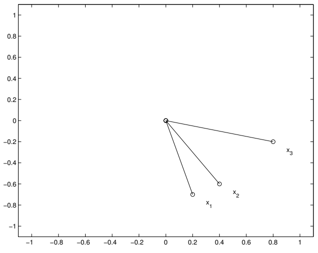
We suppose, for this section only, that our dataset has just $n = 2$ observations of the p variables. In this case, the vectors $x_1, x_2, \ldots, x_p$ lie in the plane $\mathbb{R}^2$. Some examples, with $p = 3$, are plotted below:



or



or



(a) (b) (c)

Let θ_{ij} denote the angle between x_i and x_j, measured counterclockwise from x_i. Notice, first, that in two dimensions,

$$\theta_{jk} = \theta_{ik} - \theta_{ij}$$

for any variable indices i, j, and k. Second, observe that

$$\cos(\theta_{ij}) = \frac{x_i^T x_j}{\|x_i\| \|x_j\|} = \rho_{ij},$$

where $\| \cdot \|$ is the ℓ_2-norm. This suggests that, for $n = 2$ observations of an arbitrary number p of variables, it is possible do the following:

Algorithm 1.

(a) Choose one arbitrary "bridge" variable X_b. Compute the angle θ_{ib} between X_b and each other variable X_i.

(b) For all $i < j$, compute
$\theta_{ij} = \theta_{ib} - \theta_{jb}$, and
$r_{ij} = \arccos(\theta_{ij})$, keeping track of which r_{ij} is largest.

(c) Designate the (X_i, X_j) pair whose r_{ij} is largest as the pair of most highly correlated variables.

This algorithm requires only p passes through the n observations, in order to compute the angle θ_{ib} in step (a) between the "bridge" and other variables. In contrast, the straightforward algorithm mentioned in the introduction requires p^2 passes through the n observations.

Algorithm 1 also requires, in step (b), to iterate over the $p(p-1)/2$ variable pairs. However, the computation required at each iteration is small: only algebra is performed.

Although we do not prove it rigorously, it seems that Algorithm 1 is optimal, or nearly so, in the class of algorithms that compute the most highly correlated pair of variables exactly, without making approximations. Any exact algorithm must (a) iterate over the pn observation-variable combinations at least once—any unexamined combination could invalidate the result—and (b) iterate over the $p(p-1)/2$ variable pairs at least once—any unexamined pair could be the best. This is precisely, and only, what Algorithm 1 does.

Unfortunately, Algorithm 1 is not practical, because it only works for $n \leq 2$ observations. The reason for this limitation is that Algorithm 1 works by moving back and forth between angles and cosines (that is, correlations), and this is not well-defined when $n > 2$.

Nevertheless, Algorithm 1 suggests a way to construct a general-purpose method that, while not as efficient as Algorithm 1, is better than the straightforward approach considered in the introduction.

4 The main algorithm

The following upper bound on sample correlation is given by Langford, Schwertman, and Owens [5].

Proposition 1. *For variables X_i, X_j, X_k with sample correlations $\rho_{ik}, \rho_{ij}, \rho_{jk}$, the following inequality holds:*

$$\rho_{ik} \leq \rho_{ij}\rho_{jk} + \sqrt{(1-\rho_{ij}^2)(1-\rho_{jk}^2)}.$$

This result suggests an efficient algorithm for computing maximum correlations. If the upper bound on the correlation between a pair of variables (X_i, X_k) is less than the largest correlation seen so far, then (X_i, X_k) cannot be the most highly correlated pair, and we do not need to compute the correlation between X_i and X_k to determine this. Formally:

Algorithm 2.

Initially, set $\rho^* = -\infty$.

- (1) Choose one arbitrary "bridge" variable X_b. Compute the sample correlation ρ_{ib} between X_b and each other variable X_i.
- (2) For $i < j$,
  - (a) Set $r_{ij} = \rho_{ib}\rho_{jb} + \sqrt{(1-\rho_{ib}^2)(1-\rho_{jb}^2)}$, an upper bound on the sample correlation between the variables X_i and X_j.
  - (b) If $r_{ij} > \rho^*$,
    - (i) Compute ρ_{ij}, the sample correlation between X_i and X_j, from the observations x_i and x_j, using the definition.
    - (ii) If $\rho_{ij} > \rho^*$, set $\rho^* = \rho_{ij}$.
- (3) Designate the (X_i, X_j) corresponding to the last ρ^* as the most highly correlated variables.

As already mentioned, Algorithm 2 is in some respects similar to Algorithm 1. In both cases, we aim to avoid computing the sample correlation from the definition, and in both cases, we achieve this by comparing each variable in each pair individually to a "bridge" variable.

On the other hand, a key difference from Algorithm 1 is that in Algorithm 2, when the upper bound r_{ij} is larger than the currently maximum correlation r^*, we do not thereby know the value of ρ_{ij}. Hence, in that case we need to compute the correlation from the definition, based on the observations, in order to check whether it is the largest correlation so far.

This makes analysis of Algorithm 2 more difficult, as its complexity depends strongly on how many times we need to compute the correlations. If we reach step (2bi) every iteration or nearly so, then Algorithm 2 will be no better than the straightforward algorithm presented in the introduction. On the other hand, if we never reach step (2bi) more than once per execution, then Algorithm 2 is almost as good as Algorithm 1.

It is clear that both of these are attainable: in the best case, we will only have to compute the sample correlation between one non-"bridge" (X_i, X_j) pair, while in the worst case, we will need to compute the sample correlation between every (X_i, X_j) pair. In the next section, we will carry out a more precise analysis.

5 Analysis of the main algorithm

In this section, we seek a precise understanding as to how many times the sample correlation will need to be computed from the definition in Algorithm 2. Our main result is as follows.

Theorem 1. *If the correlations ρ_{ij}, $1 \leq i < j \leq p$, are iid uniformly between -1 and 1, then the expected number of times $r_{ij} > \rho^*$ in Algorithm 2, i.e., the expected number of times Algorithm 2 will need to compute a correlation from scratch, is $\Theta(p)$.*

The remainder of this section is devoted to a proof.

First, we formalize the problem.

Definition 1. (a) Let $\mathcal{I}$ be a one-to-one map $t \mapsto (i, j)$, for $t = 1, \ldots, p(p-1)/2$, so that

$$\rho_t = \rho_{ij} \quad \text{and} \quad r_t = r_{ij} = \rho_{ib}\rho_{jb} + \sqrt{(1-\rho_{ib}^2)(1-\rho_{jb}^2)}.$$

(b) Let κ_t indicate whether the upper bound r_t is greater than all previous correlations:

$$\kappa_t = \begin{cases} 1, & \text{if } r_t > \max\{\rho_1, \rho_2, \ldots, \rho_{t-1}\}, \\ 0, & \text{otherwise.} \end{cases}$$

(c) Let v_t denote the number of times κ_t has been on so far:

$$v_t = \sum_{s=1}^{t} \kappa_s.$$

We always assume correlations are drawn iid from uniform$[-1, 1]$.

In this notation, Theorem 1 claims that $E[v_{p(p-1)/2}] = \Theta(p)$. We will establish the claim by (a) finding the distribution of r_t (Lemma 1), (b) using Watson's lemma to approximate the expected value of κ_t (Lemma 2), and (c) using Euler-Maclaurin summation to approximate the expected value of v_t (Lemma 3).

We begin with the distribution of r_t.

Lemma 1. *For fixed* $-1 \le a \le 1$*, the cdf*

$$\Pr[r_t \le a] = \frac{1}{2}(a+1) + \frac{1}{4}\sqrt{1-a^2}(\arccos(a) - \pi).$$

Proof. Note that r_t's distribution does not depend on t, so for convenience of notation we write $r(x,y) = xy + \sqrt{1-x^2}\sqrt{1-y^2}$, with x, y drawn uniformly from $[-1,1]$. We seek the area of the sublevel set

$$\begin{aligned}\{(x,y) : r(x,y) \le a\} = &\{(x,y) : -a \le x \le 1\,,\ y \le ax - \sqrt{1-a^2}\sqrt{1-x^2}\} \cup \\ &\{(x,y) : -1 \le x \le a\,,\ y \ge ax + \sqrt{1-a^2}\sqrt{1-x^2}\}.\end{aligned}$$

The area is given by

$$2\int_{-a}^{1} \left(ax - \sqrt{1-a^2}\sqrt{1-x^2} - (-1)\right) dx = 2(a+1) + \sqrt{1-a^2}(\arccos(a) - \pi).$$

Dividing by the area of $[-1,1] \times [-1,1] = 4$, the result follows. □

Next, we find an asymptotic approximation for $E[\kappa_t]$, the expected number of times the upper bound r_t is greater than the current maximum correlation:

Lemma 2. *As* $t \to \infty$*,*

$$E[\kappa_t] \sim \frac{\pi^{3/2}}{4\sqrt{t - 3/2}}.$$

Proof. First, note that $E[\kappa_t] = \int_{-1}^{1} E[\kappa_t|r] f(r)\, dr$, where

$$E[\kappa_t|r] = \Pr[r > \max\{\rho_1, \rho_2, \ldots, \rho_{t-1}\}|r] = (\Pr[\rho \le r])^{t-1} = \left(\frac{1+r}{2}\right)^{t-1},$$

since the ρ are uniform iid, and

$$f(r) = \frac{d}{dr}\left[\frac{1}{2}(r+1) + \frac{1}{4}\sqrt{1-r^2}(\arccos(r) - \pi)\right] = \frac{1}{4} + \frac{r(\pi - \arccos(r))}{4\sqrt{1-r^2}}.$$

Substituting and simplifying, we have

$$E[\kappa_t] = \frac{1}{2^{t+1}} \int_{-1}^{1} (1+r)^{t-1} \left(\frac{r(\pi - \arccos(r))}{\sqrt{1-r^2}}\right) dr + \frac{1}{2t} := I + \frac{1}{2t}.$$

We find an asymptotic approximation to I as follows. By substituting $\tau = t - 3/2$, $x = -\log(2) + \log(1+r)$, and reflecting the integrand across $x = 0$, we obtain

$$I = \frac{1}{2^{5/2}} \int_{0}^{\infty} e^{-x\tau} f(x), dx,$$

where

$$f(x) := \frac{2e^{-x}\left(-1+2e^{-x}\right)\left(\pi - \arccos(-1+2e^{-x})\right)}{\sqrt{2-2e^{-x}}}$$
$$= \frac{\sqrt{2}\pi}{\sqrt{x}} - 2\sqrt{2} - \frac{11\pi\sqrt{x}}{2\sqrt{2}} + \frac{17\sqrt{2}x}{3} + \frac{265\pi x^{3/2}}{48\sqrt{2}} + O(x^2).$$

By Watson's lemma (e.g., [6]), taking the first two terms of the series, we have

$$I \sim \frac{\pi}{2^{5/2}} \frac{\sqrt{2}(-1/2)!}{\tau^{-1/2+1}} - \frac{1}{2^{5/2}} \frac{2\sqrt{2}\,0!}{\tau^{0+1}} = \frac{\pi^{3/2}}{4\sqrt{t-3/2}} - \frac{1}{t-3/2},$$

where we have substituted $\tau = t - 3/2$ and $(-1/2)! = \sqrt{\pi}$ and simplified. We conclude that

$$E[\kappa_t] \sim \frac{\pi^{3/2}}{4\sqrt{t-3/2}} - \frac{1}{t-3/2} + \frac{1}{2t} \sim \frac{\pi^{3/2}}{4\sqrt{t-3/2}}.$$

□

Lemma 3. *As $t \to \infty$,*

$$E[v_t] \sim \frac{\pi^{3/2}\sqrt{t-3/2}}{2} = \Theta(\sqrt{t}).$$

Proof. We have

$$E[v_t] = \sum_{s=1}^{t} E[\kappa_s] \sim \sum_{s=1}^{t} \frac{\pi^{3/2}}{4\sqrt{s-3/2}} \sim \frac{\pi^{3/2}}{4} \sum_{s=a}^{t} \frac{1}{\sqrt{s-3/2}} := \frac{\pi^{3/2}}{4} \sum_{s=a}^{t} f(s),$$

where we have used Lemma 2, and we choose $a > 3/2$ to avoid the singularity. By Euler-Maclaurin summation (e.g., [7], §4.5), we have

$$E[v_t] = \frac{\pi^{3/2}}{4}\left[\int_a^t f(s)ds + \frac{1}{2}f(s)\Big|_a^t + R\right],$$

where $f(s)|_a^t := f(t) - f(a)$, and the remainder

$$R := \int_a^t \left(s - \lfloor s \rfloor - \frac{1}{2}\right) f'(s)ds \le \frac{1}{2}\int_a^t f'(s)ds = \frac{1}{2}\left(\frac{1}{\sqrt{s-3/2}}\right)\Big|_a^t.$$

Thus

$$E[v_t] = \frac{\pi^{3/2}}{4}\left[2\left(\sqrt{s-3/2}\right)\Big|_a^t + \frac{1}{2}\left(\frac{1}{\sqrt{s-3/2}}\right)\Big|_a^t + R\right] \sim \frac{\pi^{3/2}\sqrt{t-3/2}}{2},$$

since the first term dominates. □

Theorem 1 now follows by substituting $t = p(p-1)/2$:

$$E[v_{p(p-1)/2}] \sim \frac{\pi^{3/2}\sqrt{p(p-1)/2-3/2}}{2} \sim \frac{\pi^{3/2}}{2\sqrt{2}}p = \Theta(p).$$

6 Discussion and future work

We have presented a new algorithm for efficiently computing the pair of variables most highly correlated in a dataset. There are several directions of particular interest for future investigation. First, we might model the correlations ρ_t as being drawn from a more complex distribution than uniform$[-1, 1]$. In particular, it is of interest to explore how dependencies among correlations and among variables affects the number of computations needed. Second, we intend to investigate how existing learning algorithms can be improved using our algorithm, and how new algorithms can be devised. Finally, it is of interest to carry out an extensive empirical validation; preliminary work in this direction indicates that the algorithm is indeed quite effective on real datasets.

Acknowlegements

Thanks to Eric Bach for help with the analysis.

References

[1] Ann B. Lee, Boaz Nadler, and Larry Wasserman. Treelets—an adaptive multi-scale basis for sparse unordered data. *The Annals of Applied Statistics*, 2(2):435–471, 2008.

[2] Isabelle Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[3] Thorsten Brants and Alex Franz. Web 1T 5-gram version 1. *Linguistic Data Consortium, Philadelphia*, 2006.

[4] Robert Sedgewick. *Algorithms*. Addison Wesley Longman, 1983.

[5] E. Langford, N. Schwertman, and M. Owens. Is the property of being positively correlated transitive? *The American Statistician*, 55:322–325, 2001.

[6] L. Sirovich. *Techniques of Asymptotic Analysis*. Springer, 1971.

[7] Paul Walton Purdom, Jr. and Cynthia A. Brown. *The Analysis of Algorithms*. Holt, Rinehart, & Winston, 1985.