A Distributional Measure of Correlation

Dror G. Feitelson
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Abstract

Common metrics of correlation are based on the degree to which the value of one random variable can be used to predict the value of another random variable. However, there are forms of correlation that do not lead to a high ability to predict actual values. We identify such a correlation, and suggest it can be measured by a certain monotonicity property among distributions of the dependent variable, for different ranges of the independent variable.

Keywords: correlation coefficient, data analysis, distribution prediction

1 Introduction

A basic form of describing empirical multivariate data is by measuring the degree of correlation between the different variables. This can be done graphically using scatter plots, or analytically using various formulas. The most commonly used is Pearson's *correlation coefficient*

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{S(X)S(Y)}$$

where $\bar{X}$ and $\bar{Y}$ are the means of X and Y, and $S(X)$ and $S(Y)$ are their standard deviations, respectively. In effect, this measures the degree to which X and Y are linearly related. A correlation coefficient of 1 indicates a linear relationship. A coefficient of -1 indicates an inverse relationship. More generally, correlation coefficients with an absolute value near 1 indicate that when X grows, so does Y. Small correlation coefficients indicate that no such relationship holds.

The limitation of only measuring a linear relationship is reduced by using Spearman's *rank correlation coefficient*. This means that instead of using the data values directly, we first sort them, and use their ranks instead. For example, consider an item for which $X = 13.2$ and $Y = 25.7$. If the rank of this X value is 19th out of all the Xs of the different items, and the rank of this Y value is the 15th of all Ys, then the item will be represented by the pair $(19, 15)$ instead of the pair $(13.2, 25.7)$. Calculating the correlation coefficient on the ranks then gives a measure of the degree to which X can be used to predict Y. This works for all cases where the relationship between them is monotonic.

system	*CC*	*rank CC*	*dist CC*
CTC SP2	0.057	0.244	0.892
KTH SP2	0.038	0.250	0.876
NASA iPSC	0.157	0.242	0.884
SDSC Paragon	0.280	0.486	0.990
SDSC SP2	0.146	0.360	0.962
SDSC Blue	0.121	0.411	0.993
LANL CM5	0.178	0.293	0.986
LANL O2K	−0.096	−0.214	−0.872
OSC Cluster	0.029	0.158	0.889

Table 1: *Correlation coefficients calculated for the job sizes and runtimes of parallel jobs in workloads from several parallel supercomputers.*

But what about weaker forms of correlation, where the value of one variable cannot be used effectively to predict the other? While it may be impossible to make individual predictions, it may still be possible to characterize *aggregate* behavior. This is done by linking the distributions of X and Y. Specifically, we are interested in the case where knowing X may indicate the *distribution* of Y, but this is insufficient to make precise predictions regarding the exact *value* of Y, because the distributions for different Xs have a large overlap. The next section gives an example of real data for which this form of correlation occurs. Section 3 then suggest a metric for measuring such correlations, and discusses its properties.

2 Motivation and Intuition

Parallel jobs executed on parallel supercomputers have two main attributes: the number of processors they use, and how long they run. When characterizing the workloads on parallel supercomputers the correlation between these two workload attributes is of interest.

Scatter plots showing data from several large-scale installations in production use are shown in Fig. 1 (the data comes from the Parallel Workloads Archive at www.cs.huji.ac.il/labs/parallel/workload/, using the “cleaned” versions of the logs). Quite obviously, there is no linear or monotonic relationship between these two variables. In all cases, sizes emphasize small sizes and powers of two (two systems allow only powers of two), and runtimes span the range from seconds to many hours with no obvious patterns.

Calculating the correlation coefficient or the rank correlation coefficient leads to values that are typically rather low, indicating a possible weak positive correlation (Table 1). Only one system has a negative correlation. In all cases, the rank correlation coefficient leads to higher values than the correlation coefficient of the original values. However, there seems to be little connection between these calculations and the scatter plots, except, perhaps, for the SDSC Paragon, which indeed has slightly higher coefficients than the other systems. These calculations and subsequent ones focus on the parallel jobs only, excluding serial ones, as it is known that serial jobs on large parallel supercomputers tend to have unique statistics.

Why do we care whether a correlation exists? Scheduling parallel jobs on a massively parallel

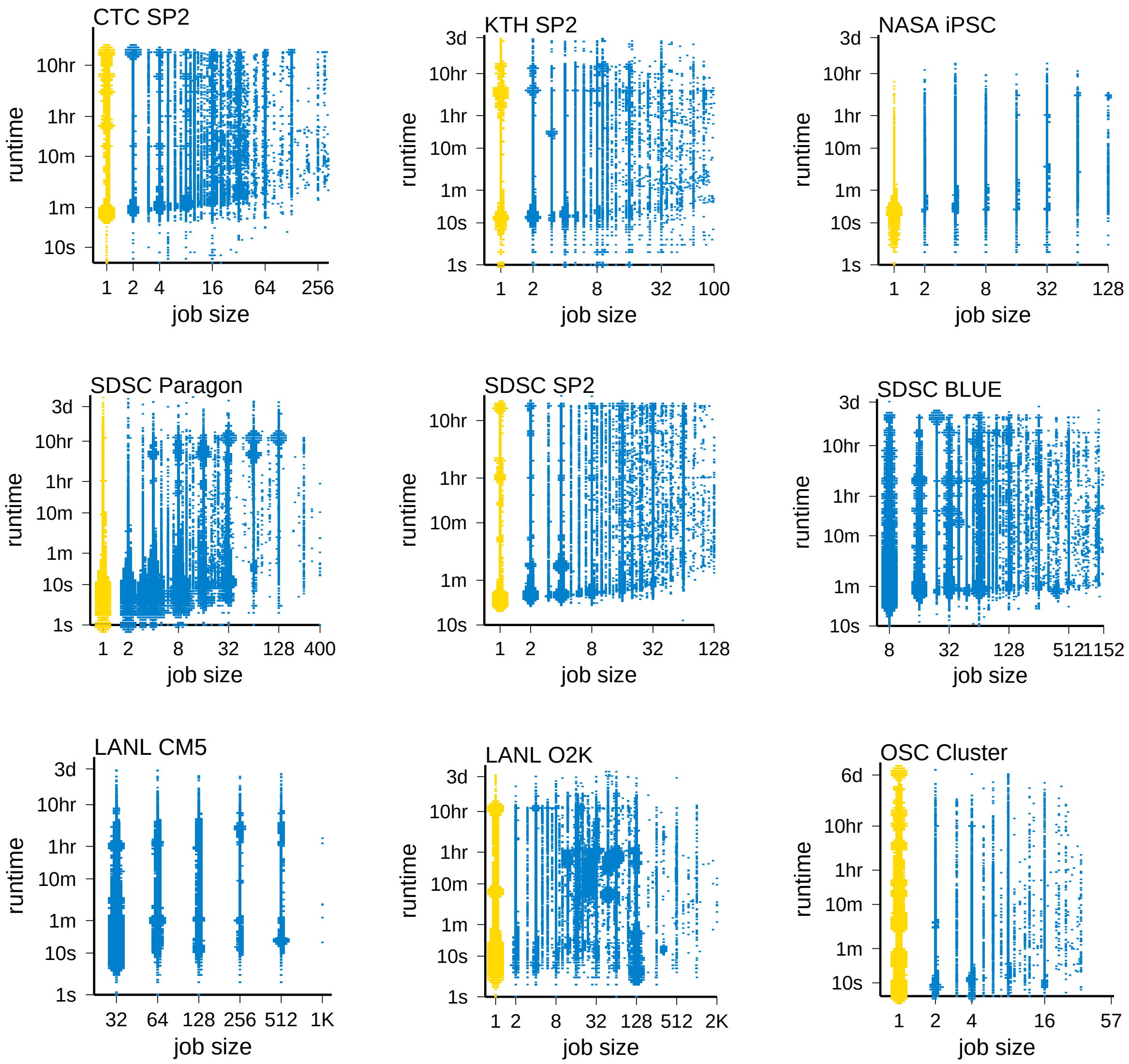

Figure 1: *Scatter plots of job sizes and runtimes for workloads from several parallel supercomputers. Note that logarithmic scales are used.*

machine is akin to 2D bin packing: each job is represented by a rectangle in processors×time space, and these rectangles have to be packed as tightly as possible. Assuming that when each job is submitted we know how many processors it needs, but do not know for how long it will run, it is natural to do the packing according to size. Specifically, packing the bigger jobs first may be expected to lead to better performance [1]. But what if there is a correlation between size and running time? If this is an inverse correlation, we find a win-win situation: the larger jobs are also shorter, so packing them first is statistically similar to using SJF (shortest job first), which is known to lead to the minimal average runtime [4]. But if size and runtime are positively correlated, and large jobs run longer, scheduling them first may cause significant delays for subsequent smaller jobs, leading to dismal average performance results [5].

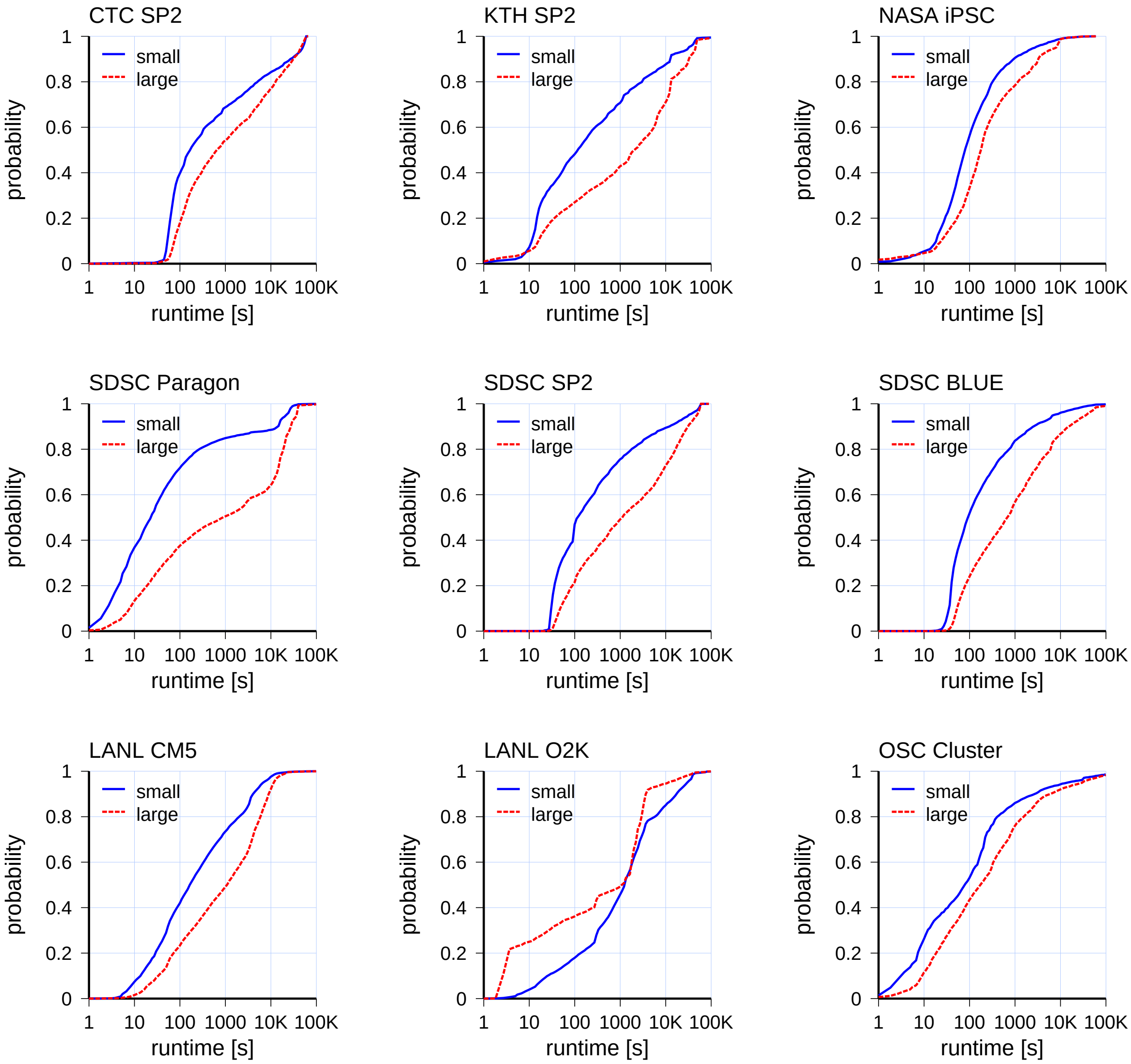

Figure 2: *Distributional correlation among job sizes and runtimes of parallel jobs in different workloads. Compare with Fig. 1. "Small" and "large" are defined relative to the median size for each system, given in Table 2.*

In order to assess whether some form of correlation exists, we suggest the following procedure. First, partition each workload log into two equal parts: the half with the smaller jobs, and the half with the larger jobs (where "small" or "large" relates to the number of processors used). Now plot the cumulative distribution functions of runtimes for the jobs in the two sets. If one CDF is consistently above the other, we say that a distributional correlation exists. If the two repeatedly cross each other, there is no such correlation.

The results of using this procedure are demonstrated in Fig. 2, for the same systems shown in Fig. 1. Obviously, there is a strong distributional correlation in all systems. In all but one this is a positive correlation: the CDF of the smaller jobs is above that of the large jobs, indicating that their runtimes are typically shorter. For the LANL O2K machine, an inverse relationship seems to hold.

This answers the question of how parallel job size and runtime are related, and can be incorporated in workload models. It is formalized and quantified in the following section.

3 Definition and Properties

Our measure of distributional correlation is that one CDF be above the other. This can be quantified as

$$distCC = \frac{1}{|P|} \sum_{y \in P'} \text{sgn}\Big(F^{\perp}(y) - F^{\top}(y)\Big)$$

where $F^{\perp}(\cdot)$ is the empirical distribution function of Y for samples in which X is in the bottom half of values, $F^{\top}(\cdot)$ is the distribution function of Y for samples in which X is in the top half, and sgn is the sign function

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$$

This definition of a distributional correlation coefficient leads to values in the range $[-1, 1]$, due to dividing by the size of P — the set of all sample points. 1 indicates the strongest correlation, -1 indicates an inverse correlation, and values near 0 indicate little correlation — the same as for other coefficients.

The set of points P' over which the sum is taken can be set to all points at which the CDFs differ:

$$P' = \Big\{ y \Big| \; |F^{\perp}(y) - F^{\top}(y)| > 0 \Big\}$$

This fits naturally with the decision to partition the dataset at the median of the independent variable, as this ensures that both CDFs are defined by the same number of samples. Note that all samples are counted: if a certain values appears multiple times, its contribution to the metric reflects this multiplicity.

However, one must also consider the discriminatory power of the results. In particular, we want to ensure that high values cannot occur by chance. Regrettably, with P' as defined above, this is not the case. The leftmost graph in Fig. 3 shows that results with high absolute values are in fact quite common. The reason is that a random fluctuation might cause a small discrepancy between the two CDFs, which is then maintained. We therefore need some way to discriminate against small differences that could occur by chance.

One way to look at the two CDFs when samples are randomly assigned to the small and large groups (thus simulating the case where the CDFs are actually the same distribution) is as follows. We start from the smallest values and move upward. Each new sample has a probability of 0.5 to increase the CDF of the small group, and the same probability of 0.5 to increase the CDF of the large group. Thus the difference between the CDFs behaves like a random walk with equal probabilities to take a step in either direction. It is well-known that the expected distance that such a random walk covers after n steps is $\sqrt{n}$. Therefore we may expect the difference between the CDFs to be proportional to the square root of the number of samples we have seen so far. This reasoning leads to the definition

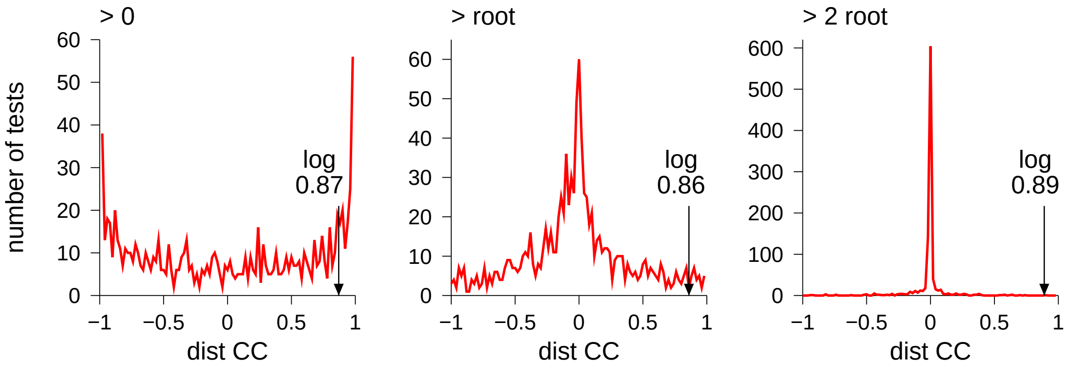

Figure 3: *Distribution of distCC for 1000 random partitionings of the jobs in the CTC SP2 log into two groups, rather than partitioning according to job size. The three graphs correspond to three ways to select the set P'.*

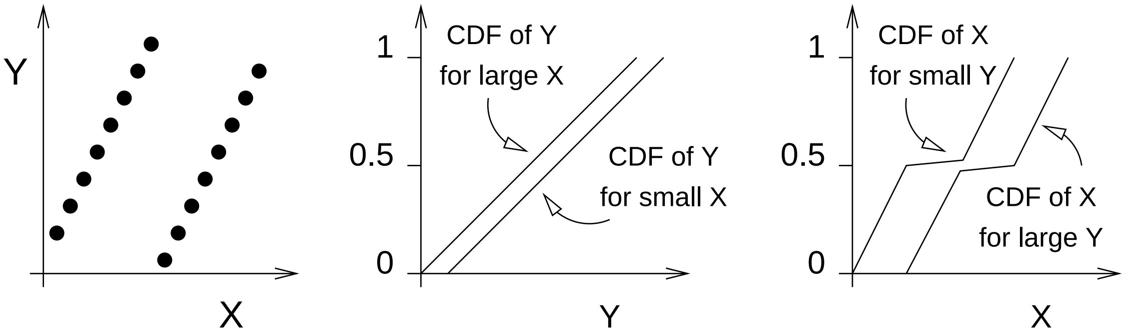

Figure 4: *Demonstration of asymmetry of the distributional correlation coefficient. Given the hypothetical scatter plot on the left, calculating $distCC$ after dividing the dataset using X leads to a value of -1 (middle), whereas calculating it after dividing using Y leads to a value of 1 (right).*

$$P' = \left\{ y \,\middle|\, |F^{\perp}(y) - F^{\top}(y)| > \sqrt{|\{y' \mid y' \leq y\}|} \right\}$$

As shown in the middle of Fig. 3 this is much better, and the distribution of random results peaks prominently at 0. However, the probability of high values is still uncomfortably high. But if we require that the difference between the CDFs be twice as big, as in

$$P' = \left\{ y \,\middle|\, |F^{\perp}(y) - F^{\top}(y)| > 2\sqrt{|\{y' \mid y' \leq y\}|} \right\}$$

we eliminate practically all possibility of getting a high value by chance, as shown in the right of Fig. 3. We therefore adopt this definition of P'.

The results of computing the distCC metric with this P' are shown in Table 1, and compared with the conventional correlation coefficients. Obviously the distributional coefficients are rather close to 1 (or -1), as one would expect based on the distributions shown in Fig. 2.

Note that the distributional correlation coefficient has the drawback of not being symmetric, as opposed to conventional correlation coefficients that are symmetric (in this it is similar, however,

	div by size		*div by time*	
system	*median*	*dist CC*	*median*	*dist CC*
CTC SP2	8	0.892	352	0.976
KTH SP2	6	0.876	503	0.905
NASA iPSC	16	0.884	135	0.878
SDSC Paragon	8	0.990	71	0.974
SDSC SP2	8	0.962	342	0.927
SDSC Blue	8	0.993	229	0.987
LANL CM5	32	0.986	414	0.931
LANL O2K	48	−0.872	1338	−0.094
OSC Cluster	4	0.889	124	0.251

Table 2: *Results of calculating $distCC$ both ways for the parallel jobs data.*

to regression analysis, which is also asymmetrical). The lack of symmetry is the result of choosing one variable to split the observations into two, and then plotting the distributions of the other variable. It is possible that different results would be obtained if we choose the variables the other way around. In fact, it is even possible to find a high positive correlation when using one variable, and an inverse correlation when using the other, as demonstrated in Fig. 4.

Does this indeed occur in practice? As a rudimentary check, we calculate $distCC$ for the parallel jobs data after dividing it by runtimes, rather than by sizes, and counting for all possible size values. The results are shown in Table 2. For most systems the result is similar albeit sometimes a bit lower than it was when dividing by sizes. The only exceptions are the OSC cluster, for which the result is significantly lower, and the LANL O2K system, that went from a strong negative correlation to a very weak (but still negative) correlation. Based on these results, it seems advisable to check both directions when analyzing unknown data.

4 Conclusions

We have identified distributional correlation as a new property of data sets that is not reflected in the conventionally used correlation coefficients. While less useful for accurate predictions, this property is nevertheless potentially important for correct modeling of the data.

The proposed distributional correlation coefficient provides a means to quantify this effect in data, and to verify that it is retained to a similar degree in a model based on this data. In fact, it also suggests a modeling technique: start with a statistical model of one attribute, and generate a family of models for the other attribute conditioned on the value of the first attribute. It thereby provides a justification for such models that have been proposed in the past [2, 3].

Acknowledgments

This research was supported in part by the Israel Science Foundation (grant no. 167/03). Many thanks are due to all those who deposited their workload logs in the Parallel Workloads Archive.

References

[1] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson, “*Approximation algorithms for bin-packing — an updated survey*”. In *Algorithm Design for Computer Systems Design*, G. Ausiello, M. Lucertini, and P. Serafini (eds.), pp. 49–106, Springer-Verlag, 1984.

[2] D. G. Feitelson, “*Packing schemes for gang scheduling*”. In *Job Scheduling Strategies for Parallel Processing*, pp. 89–110, Springer-Verlag, 1996. Lect. Notes Comput. Sci. vol. 1162.

[3] J. Jann, P. Pattnaik, H. Franke, F. Wang, J. Skovira, and J. Riodan, “*Modeling of workload in MPPs*”. In *Job Scheduling Strategies for Parallel Processing*, pp. 95–116, Springer Verlag, 1997. Lect. Notes Comput. Sci. vol. 1291.

[4] P. Krueger, T-H. Lai, and V. A. Dixit-Radiya, “*Job scheduling is more important than processor allocation for hypercube computers*”. *IEEE Trans. Parallel & Distributed Syst.* **5(5)**, pp. 488–497, May 1994.

[5] V. Lo, J. Mache, and K. Windisch, “*A comparative study of real workload traces and synthetic workload models for parallel job scheduling*”. In *Job Scheduling Strategies for Parallel Processing*, pp. 25–46, Springer Verlag, 1998. Lect. Notes Comput. Sci. vol. 1459.