Computing the standard deviation efficiently

Mark Hoemmen <mhoemmen AT cs DOT berkeley DOT edu>

25 August 2007

1 Statistics review

1.1 Arithmetic mean

Let's say we have a set of n numbers x_1, x_2, $\ldots$, x_n. These might be heights of a collection of people, the scores of the individual students in a class on a particular exam, or the number of points that a particular basketball player scores in each game in a season. The average or (more accurately) *arithmetic mean* of these numbers is

$$\frac{1}{n} \sum_{k=1}^{n} x_i. \tag{1}$$

Some authors write μ (pronounced "mu") for the arithmetic mean, and some write $\bar{x}$. I like Greek letters, so I'll write μ.

1.2 Variance and deviation

The mean tells us the "most likely number," but we also might like to know something about how close to the mean the numbers tend to be. For example, just knowing that everybody in a class scored 70% on an exam doesn't tell you a whole lot. Maybe everybody got 70% (which means everyone understood the material pretty well), or maybe 70 people got 100% and 30 got 0% (which means a lot of people didn't understand the test at all!). Statisticians use the *variance* to measure "average distance from the mean." There are (at least) two different kinds of variance: *population variance* σ^2, and *sample variance* s^2. The sample variance s^2 of n numbers $x_1, \ldots, x_n$ is defined as

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu), \tag{2}$$

and the population variance σ^2 is defined as

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu). \tag{3}$$

The only difference is the divisor in front, and for large data sets, the difference is very small indeed.

Why do we define the variance as some number *squared*? This is because the actual "distance of the data set from the mean" isn't the variance, but the square root of the variance. The quantity σ, which is the square root of the population variance, is called the *standard deviation.* This term might be more familiar to you. The number s is called the *sample deviation.* Remember the Pythagorean theorem? Imagine that there are two data points (i.e., $n = 2$). Then we have:

$$s_n = \sqrt{(x_1 - \mu)^2 + (x_2 - \mu)^2}$$

which is just the distance between the points (x_1, x_2) and (μ, μ) in the two-dimensional plane. The $1/\sqrt{n-1}$ factor just averages out that distance over all the points.

2 Computing the variance

2.1 Two-pass formula

Equation (2) gives you a formula for computing the variance. This is called a "two-pass" formula because it requires two passes through the data, once to compute the mean and once to compute the variance. You might do it like this:

```
$\mu := 0$
for $k = 1$ to $n$ do
    $\mu := \mu + \frac{x_i}{n}$
end for
$v := 0$
for $k = 1$ to $n$ do
    $v := v + (x_k - \mu)^2$
end for
if we want the standard variance then
    Return $v/n$
else                                        ▷ we want the sample variance
    Return $v/(n-1)$
end if
```

Passing through the data twice is annoying if you have lots of data. Most of the cost of computing the mean and variance on modern computers is just running through the array of data: this is a bandwidth cost. Reading the array twice pretty much means doubling the runtime. Also, let's say you don't actually have an array of data; instead, each datum is generated on the fly, like this:

```
for $k = 1$ to $n$ do
    Create some number $x_k$ somehow
    Do something with $x_k$ and then throw it away
end for
```

The two-pass variance algorithm means that you can't throw away each x_k in the loop. You have to save it in an array, wait until you've gone through all the numbers and computed the mean, and then go through the whole array and compute the variance. This wastes both space and time. We'll see, however, that making a one-pass formula work isn't as easy as you might think.

2.2 One-pass formulas

2.2.1 The wrong formula

Some statistics books give an alternate formula for the sample variance, which only requires one pass through the data:

$$s_n^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right). \qquad (4)$$

You can show that this is algebraically the same as Equation (2); just take that equation and rearrange the terms. Here's how you might code up this one-pass formula:

```
$sum := 0$
$sumsq := 0$
for $k = 1$ to $n$ do
    $sum := sum + x_k$
    $sumsq := sumsq + x_k^2$
end for
$v := \left(sumsq - \frac{1}{n} sum^2\right)$
if we want the standard variance then
    Return $v/n$
else                                        ▷ we want the sample variance
    Return $v/(n-1)$
end if
```

Although this algorithm is mathematically correct, it often gives you the wrong answer, because of rounding error. The way that it's arranged makes it very susceptible to rounding error, but the two-pass formula doesn't have this problem.[1]

You may think this is something that would only matter for really strange inputs, but it's actually easy to come up with normal-looking inputs that break the algorithm. For example, let's say you're working with single-precision floating-point numbers (the C **float** datatype), and you have $x_1 = 10000$, $x_2 = 10001$, and $x_3 = 10002$. Then the two-pass formula gives you $s_n^2 = 1.0$ (which is exactly right), but the one-pass formula above gives you 0.0 (which is 100% wrong!).

[1] The problem with this one-pass formula is that it takes the difference of two positive numbers that might be very close together. If the positive numbers themselves aren't exact but have been rounded off somehow, then taking their difference can magnify this rounding error. This is called *cancellation*. In fact, sometimes Equation (4) gives a negative answer, which is impossible according to the definition of variance. In contrast, the two-pass formula adds up a bunch of nonnegative numbers. It can never be negative, and is much less susceptible to cancellation. See [2] for details.

You can see from this that rounding error is a big deal! It's something you should learn about before you graduate. If you ever plan to use **float** or **double**, you need to know about rounding error!

2.2.2 A better formula

Fortunately, there are one-pass formulas with much better accuracy than Equation (4). Let's define two quantities, M_k and Q_k:

$$\begin{aligned} M_k &= \begin{cases} x_1, & k = 1, \\ M_{k-1} + \frac{x_k - M_{k-1}}{k}, & k = 2, \ldots, n, \end{cases} \\ Q_k &= \begin{cases} 0 & k = 1, \\ Q_{k-1} + \frac{(k-1)(x_k - M_{k-1})^2}{k}, & k = 2, \ldots, n, \end{cases} \end{aligned} \tag{5}$$

Once we get up to Q_n, then $s_n^2 = Q_n/(n-1)$ is the sample variance, and $\sigma^2 = Q_n/n$ is the standard variance. This is true because

$$M_k = \frac{1}{k} \sum_{i=1}^{k} x_i$$

and

$$Q_k = \sum_{i=1}^{k} (x_i - M_k)^2 = \sum_{i=1}^{k} x_i^2 - \frac{1}{k} \left(\sum_{i=1}^{k} x_i \right)^2 .$$

Proving *why* this method is more accurate would be a nice exercise for Math 128 or 221, but I won't make you do it here. You may want to check the algebra, though. Remember: you have to use the first set of formulas (Equation (5)) for computing M_k and Q_k. Changing the algebra changes the roundoff properties.

3 Summary

- *Variance* measures average distance of a data set from its mean.
- The usual variance formula makes you keep all the data and pass through it twice.
- One-pass formulas let you throw away each datum after processing it. This saves space and time.
- The obvious one-pass variance formula is inaccurate, but there is an accurate formula.

References

[1] T. F. Chan, G. H. Golub, and R. J. LeVeque, *Algorithms for computing the sample variance: Analysis and recommendations*, The American Statistician, 37 (1983), pp. 242–247.

[2] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, second ed., 2002.