[Thesis text....] ...As one can easily notice, we have got exactly the same curve as Kanerva. Both his and our model expect that, after reading, say, from 550 bits of distance from a written bitstring, we should obtain the $n/2$ equator distance. We have not, however. This question has intrigued us, and here we look for a more analytic explanation than merely interference from the other written attractors. Let us turn back to mathematics to study this anomaly.

1 A deviation from the equator distance?

Dr. Kanerva writes[1]:

> You have done an incredibly thorough analysis of SDM. I like the puzzle in your message and believe that your simulations are correct and to be learned from. So what to make of the difference compared to my Figure 7.3 (and your Figure **??**)? I think the difference comes from my not having accounted fully for the effect of the other 9,999 vectors that are stored in the memory. You say in it
>
> "Our results show that the theoretical prediction is not accurate. There are interaction effects from one or more of the attractors created by the 10,000 writes, and these attractors seem to raise the distance beyond 500 bits (Figure **??**)."
>
> I think that is correct. It also brings to mind a comment Louis Jaeckel made when we worked at NASA Ames. He pointed out that autoassociative storage (each vector is stored with itself as the address) introduces autocorrelation that my formula for Figure 7.2 did not take into account. When we read from memory, each stored vector exerts a pull toward itself, which also means that each bit of a retrieved vector is slightly biased toward the same bit of the read address, regardless of the read address. We never worked out the math.

[1] Email thread 'SDM: A puzzling issue and an invitation', started March 16th 2018, in which we discussed the aforementioned discrepancy.

This is an important observation. A hard location is activated because it shares many dimensions with the items read from or written onto it. Imagine the 'counter's eye view': each individual counter 'likes' to write on its own corresponding bit-address value more than it likes the opposite; as each hard-location has a say in its own area — and nowhere else.

Let x and y be random bitstrings and n be the number of dimensions in the memory; let x_i and y_i be the i-th bit of x and y, respectively; and $d(x, y)$ be the Hamming distance. Whilst the probability of a shared bit-value between same dimension-bits in two random addresses is 1/2, an address only activates hard-locations close to it. Let us call these shared bitvalues a *bitmatch in dimension i*.

So, what is the probability of bitmatches given that we know the access radius r between the address and a hard-location?

Theorem 1.1. Each dimension has a small pull bias, which can be measured by $P(x_i = y_i | d(x, y) \leq r) = \frac{\sum_{k=0}^{r} \binom{n-1}{k}}{\sum_{k=0}^{r} \binom{n}{k}}$.

Proof. The left-hand expression $P(x_i = y_i | d(x, y) \leq r)$ computes the probability of a bitmatch in i, given that we know that x and y are in the access radius defined by r, i.e., $d(x, y) \leq r$.

Applying the law of total probability to the left-hand expression we obtain

$$\sum_{k=0}^{r} P(x_i = y_i | d(x, y) = k \leq r) P(d(x, y) = k | d(x, y) \leq r) \tag{1}$$

We also know that

$$P(x_i = y_i | d(x, y) = k) = \frac{n-k}{n} \tag{2}$$

$$P(d(x, y) = k | d(x, y) \leq r) = \frac{\binom{n}{k}}{\sum_{j=0}^{r} \binom{n}{j}} \tag{3}$$

Hence,

$$P(x_i = y_i | d(x,y) \leq r) = \frac{\sum_{k=0}^{r} \frac{n-k}{n} \binom{n}{k}}{\sum_{j=0}^{r} \binom{n}{j}} \tag{4}$$

Finally, the combinatorial identity

$$\frac{n-k}{n} \binom{n}{k} = \frac{(n-k)}{n} \frac{n!}{(n-k)!k!} = \frac{(n-1)!}{k!(n-1-k)!} = \binom{n-1}{k} \tag{5}$$

closes the theorem.

□

Theorem 1.1 is valid for both "x at x" (autoassociative memory) and "random at x" (heteroassociative memory). When $n = 1,000$ and $r = 451$, $P(x_i = y_i | d(x,y) \leq r) = p = 0.552905498137$. Each bit of a hard location does indeed have a small pull bias. What is meant by this is that each particular dimension has a small preference toward positive values if its address bit is set to 1, and negative values otherwise.

So far we have looked only at a single dimension. Let us think about how the entire bitstring is affected by this result.

Theorem 1.2. Given a reading address x and a dimension i, the number of activated hard-locations with bitmatches at i follows a normal distribution with $\mathbf{E}[Z] = ph$ *and* $\mathbf{V}[Z] = hp(1-p) + p^2 H p_1 (1-p_1)$.

Let Z be the number of activated hard locations with the same bit as the reading address:

$Z = \sum_{i=1}^{h} X_i$,

where $\mathbf{E}[h] = h$,

$\mathbf{V}[h] = Hp_1(1-p_1)$,

$p_1 = 2^{-n} \sum_{k=0}^{r} \binom{n}{k}$, and

$X_i \sim \text{Bernoulli}(p)$.

By the central limit theorem, Z is normally distributed.

Applying the law of total average and the law of total variance, $\mathbf{E}[Z] = \mathbf{E}[\mathbf{E}[Z|h]] = \mathbf{E}[ph] = p\mathbf{E}[h] = ph$, and $\mathbf{V}[Z] = \mathbf{E}[\mathbf{V}[Z|h]] + \mathbf{V}[\mathbf{E}[Z|h]] = \mathbf{E}[hp(1-p)] + \mathbf{V}[ph] = p(1-p)\mathbf{E}[h] + p^2\mathbf{V}[h] = hp(1-p) + p^2 H p_1 (1-p_1)$.

See Figure 1 for a comparison between the theoretical model and a simulation.

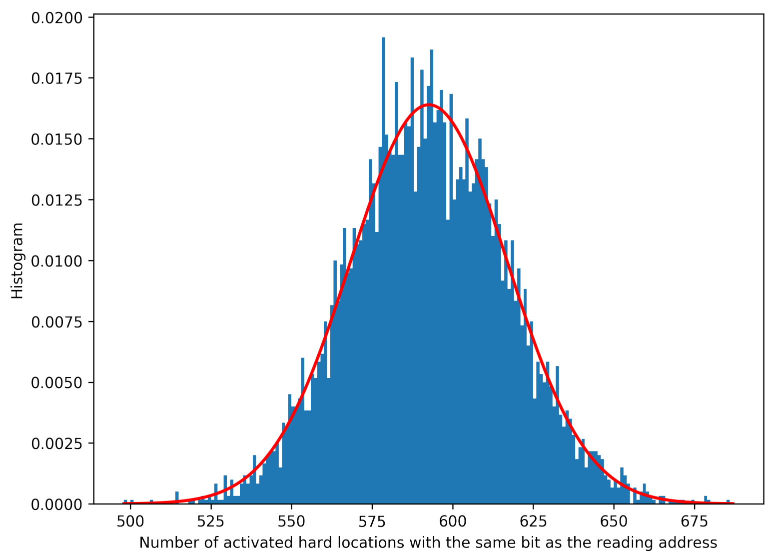


Figure 1: The histogram was obtained through simulation. The red curve is the theoretical normal distribution found in 1.2.

2 Counter bias

The bias begins in the counters. Let's analyze the ith counter of a hard location.

Let s be the number of bitstrings written into memory (in our case, $s = 10,000$), h be the average number of activated hard locations ($h = 1,071.85$), H be the number of hard locations ($H = 1,000,000$), and addr_i be the ith bit of the hard location's address.

Let $\theta = \frac{sh}{H}$ be the average number of bitstrings written in each hard location, and $X_k \sim \text{Bernoulli}(p)$ (where $p = P(x_i = y_i | d(x, y) \leq r)$). Thus,

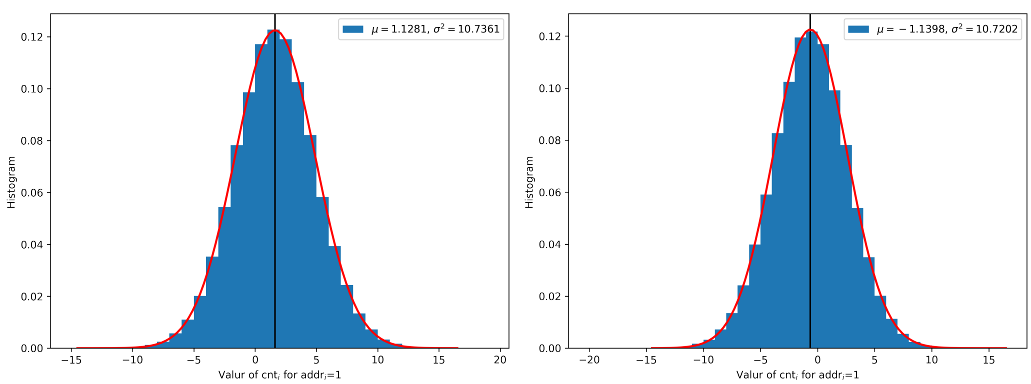
$$Y_i = \sum_{k=1}^{\theta} X_k \sim \mathcal{N}(\mu_1 = p\theta, \sigma_1^2 = p(1-p)\theta + p^2 s^2 p_1(1-p_1)/H) \tag{6}$$

During a write operation, the counters are incremented for every bit 1 and decremented for every bit 0. So, after s writes, there will be θ bitstrings written in each hard location, Y_i bits 1, and $\theta - Y_i$ bits 0. Thus, $[\text{cnt}_i|\text{addr}_i = 1] = (Y_i) - (\theta - Y_i) = 2Y_i - \theta$; and $[\text{cnt}_i|\text{addr}_i = 0] = \theta - 2Y_i$.

Hence, as $\text{cnt}_i = 2Y_i - \theta$, $\mathbf{E}[2Y_i - \theta] = 2\mathbf{E}[Y_i] - \theta$, and $\mathbf{V}[2Y_i - \theta] = 4\mathbf{V}[Y_i]$, then,

$$[\text{cnt}_i|\text{addr}_i = 1] \sim \mathcal{N}(\mu_2 = (2p-1)\theta, \sigma_2^2 = 4\sigma_1^2) \tag{7}$$
$$[\text{cnt}_i|\text{addr}_i = 0] \sim \mathcal{N}(\mu_2 = -(2p-1)\theta, \sigma_2^2 = 4\sigma_1^2) \tag{8}$$

In our case, $p = 0.5529$, $s = 10,000$, $h = 1,071.85$, and $H = 1,000,000$, so $\theta = 10.7185$ and $\text{cnt}_i \sim \mathcal{N}(\mu = 1.1341, \sigma^2 = 10.7294)$. For "random at x", $p = 0.5$, so $\mu = 0$ and $\sigma^2 = 10.8255$. See Figure 2.



(a) $\text{addr}_i = 1$ (b) $\text{addr}_i = 0$

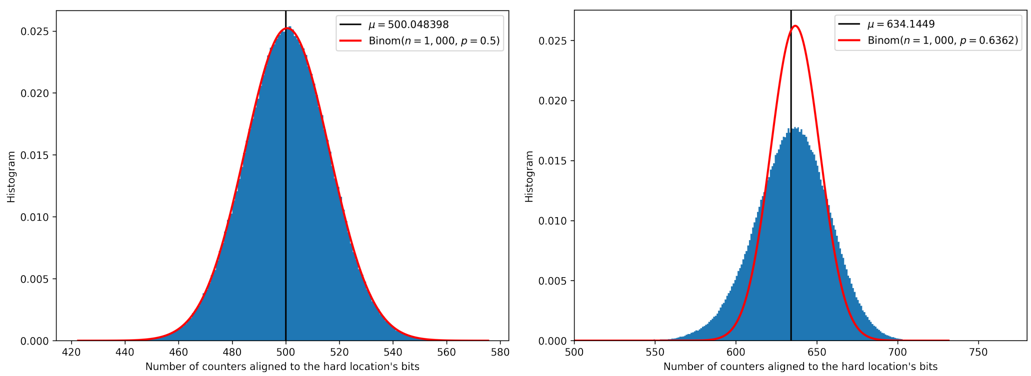
Figure 2: The value of the counters after $s = 10,000$ writes shows the autocorrelation in the counters in autoassociative memories ("x at x"). The histogram was obtained through simulation. The red curve is the theoretical normal distribution found in (7) and (8).

Finally,

$$P(\text{cnt}_i > 0|\text{addr}_i = 1) = P(\text{cnt}_i < 0|\text{addr}_i = 0) = 1 - \mathcal{N}.\text{cdf}(0) \tag{9}$$

For "random at x", $p = 0.5$ implies $\mu_2 = 0$, which implies $P(\text{cnt}_i > 0|\text{addr}_i = 1) = P(\text{cnt}_i < 0|\text{addr}_i = 0) = 0.5$, independently of the parameters because they will only affect the variance and the normal distribution is symmetrical around the average.

However, for "x at x", $p = 0.5529$ and the probabilities depend on s. For $s = 10,000$, they are equal to 0.6354. For $s = 20,000$, they are equal to 0.6867. For $s = 30,000$, they are equal to 0.7232. The more random bitstrings are written into the memory, the more the hard locations point to themselves. See Figure 3 — and notice that I still have to figure out why the mean is correct, but the standard deviation is not. As each of the n counters of a hard location may be equal or not with the same probability, I assumed it would follow a Binomial distribution (and it worked for "random at x").



(a) "random at x"

(b) "x at x"

Figure 3: Autocorrelation in the counters in autoassociative memories ("x at x"). The histogram was obtained through simulation. The red curve is the theoretical distribution.

3 Read bias

Now that we know the distribution of $\text{cnt}_i|\text{addr}_i$, we may go to the read operation. During the read operation, on average, h hard locations are activated and their counters are summed up. So, for the ith bit,

$$\text{acc}_i = \sum_{k=1}^{h} \text{cnt}_k \tag{10}$$

Let η be the reading address and η_i the ith bit of it. Then, let's split the h activated hard locations into two groups: (i) the ones with the same bit as η_i with ph hard locations, and (ii) the ones with the opposite bit as η_i with $(1-p)h$ hard locations.

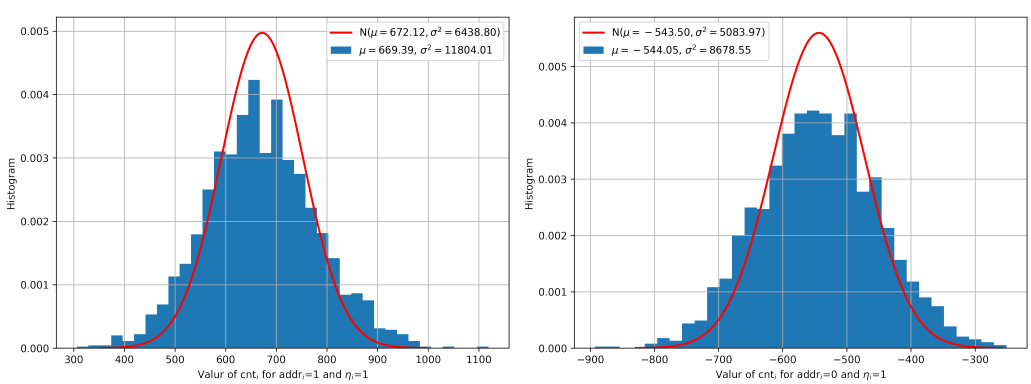
$$[\mathrm{acc}_i|\eta_i] = \sum_{k=1}^{ph} [\mathrm{cnt}_k|\mathrm{addr}_k = \eta_i] + \sum_{k=1}^{(1-p)h} [\mathrm{cnt}_k|\mathrm{addr}_k \neq \eta_i] \tag{11}$$

Each sum is a sum of normally distributed random variables, so

$$\sum_{k=1}^{ph} [\mathrm{cnt}_k|\mathrm{addr}_k = \eta_1] \sim \mathcal{N}(\mu_3 = \mu_2 ph, \sigma_3^2 = \sigma_2^2 ph + \mu_2^2 hp(1-p)) \tag{12}$$

$$\sum_{k=1}^{(1-p)h} [\mathrm{cnt}_k|\mathrm{addr}_k \neq \eta_1] \sim \mathcal{N}(\mu_3 = -\mu_2(1-p)h, \sigma_3^2 = \sigma_2^2(1-p)h + \mu_2^2 hp(1-p)) \tag{13}$$

In our case, $\sum_{k=1}^{ph} [\mathrm{cnt}_k|\mathrm{addr}_k = 1] \sim \mathcal{N}(\mu = 672.12, \sigma^2 = 6281.00)$, and $\sum_{k=1}^{ph} [\mathrm{cnt}_k|\mathrm{addr}_k = 1] \sim \mathcal{N}(\mu = -543.49, \sigma^2 = 5078.99)$. See Figure 4 — we can notice that the average is correct but the variance is too small.



(a) Equation 12 ($\mathrm{addr}_k = 1$)

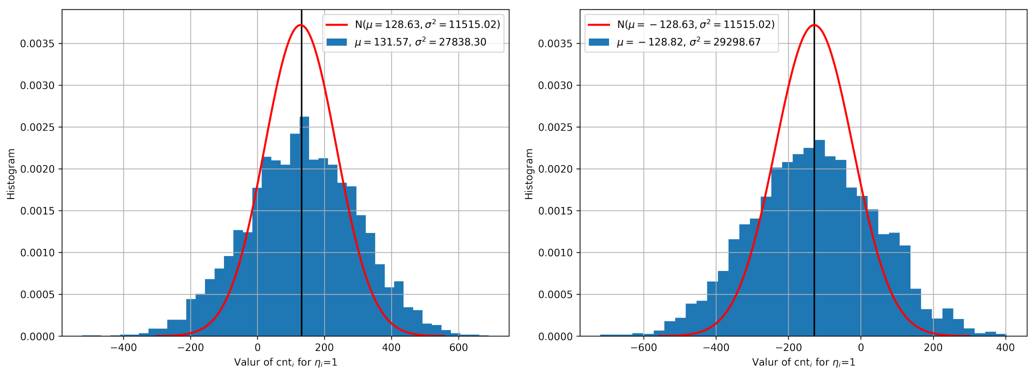
(b) Equation 13 ($\mathrm{addr}_k = 0$)

Figure 4: The histogram was obtained through simulation. The red curve is the theoretical normal distribution.

Hence,

$$[\mathrm{acc}_i|\eta_i = 1] \sim \mathcal{N}(\mu = (2p-1)^2\theta h, \sigma^2 = \sigma_2^2 h + 2\mu_2^2 hp(1-p)) \tag{14}$$
$$[\mathrm{acc}_i|\eta_i = 0] \sim \mathcal{N}(\mu = -(2p-1)^2\theta h, \sigma^2 = \sigma_2^2 h + 2\mu_2^2 hp(1-p)) \tag{15}$$

In our case, $[\mathrm{acc}_i|\eta_i = 1] \sim \mathcal{N}(\mu = 128.62, \sigma^2 = 12181.95)$, and $[\mathrm{acc}_i|\eta_i = 0] \sim \mathcal{N}(\mu = -128.62, \sigma^2 = 12181.95)$. See Figure 5 — we can notice that the variance issue from Figure 4 has propagated to these images.



(a) Equation 14 ($\eta_k = 1$)

(b) Equation 15 ($\eta_k = 0$)

Figure 5: The histogram was obtained through simulation. The red curve is the theoretical normal distribution.

Finally,

$$P(wrong) = P(\mathrm{acc}_i < 0|\eta_i = 1) \cdot P(\eta_i = 1) + P(\mathrm{acc}_i > 0|\eta_i = 0) \cdot P(\eta_i = 0) \tag{16}$$
$$= \frac{\mathcal{N}_{\eta_i=1}.\mathrm{cdf}(0)}{2} + \frac{1 - \mathcal{N}_{\eta_i=0}.\mathrm{cdf}(0)}{2} \tag{17}$$
$$= \frac{\mathcal{N}_{\eta_i=1}.\mathrm{cdf}(0)}{2} + \frac{\mathcal{N}_{\eta_i=1}.\mathrm{cdf}(0)}{2} \tag{18}$$
$$= \mathcal{N}_{\eta_i=1}.\mathrm{cdf}(0) \tag{19}$$

Using the empirical variance of $\sigma^2 = 27838.3029124$, we calculate $P(wrong) = 0.22037771219874325$.

In order to check this probability, I have run a simulation reading from 1,000 random bitstrings (which have never been written into memory) and calculate the distance from the result of a single read. As the $P(wrong) = 0.22037$, I expected to get an average distance of 220.37 with a standard deviation of 13.10. See Figure 6 for the comparison between the simulated and the theoretical outcomes.

Figure 7 shows the new distance between η_d and read(η_d), where η_d is d bits away from η. As for $d \geq 520$ there is no intersection between η and η_d, our models applies and explains the horizontal line around distance 220.

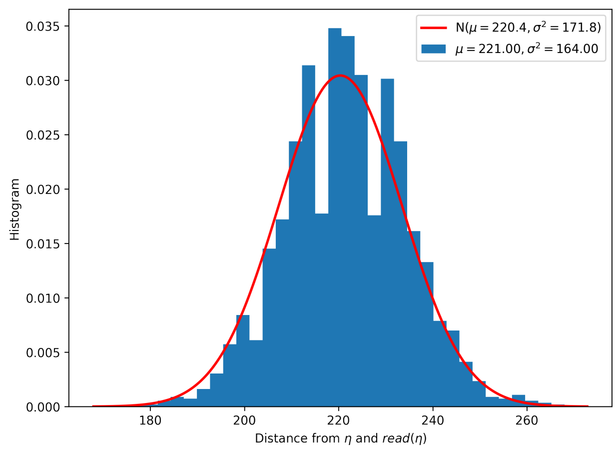


Figure 6: The histogram was obtained through simulation. The red curve is the theoretical normal distribution.

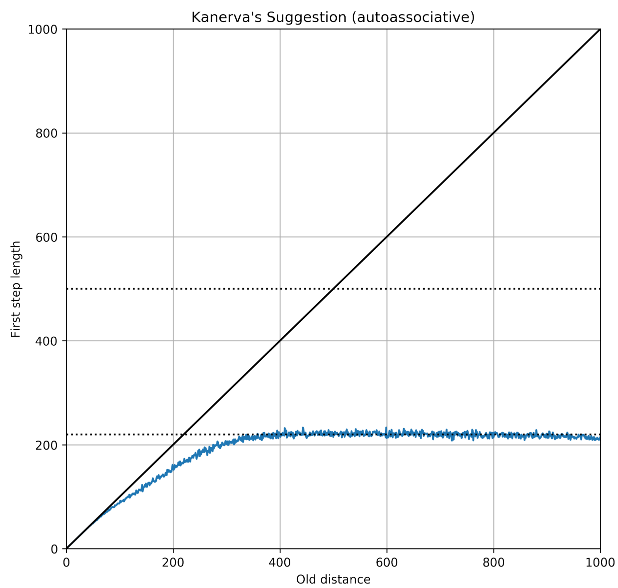


Figure 7: New distance after a single read operation in a bitstring η_d, which is d bits away from η. The new distance was calculated between η_d and read(η_d). Notice that when $d \geq 520$, the intersection between η and η_d is zero, which means there is only random bitstrings written into the activated hard locations. The distance 220 equals $1000 \cdot 0.220$ which is the probability find in Figure 6.