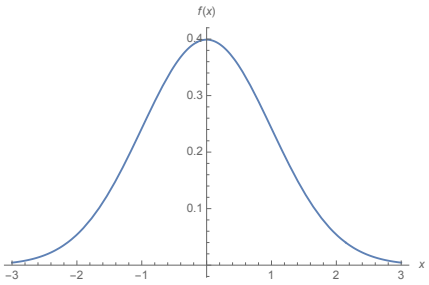
22 High Dimensions; Random Projection; Pseudoinverse; Personality

THE GEOMETRY OF HIGH-DIMENSIONAL SPACES

[High-dimensional geometry sometimes acts in ways that are completely counterintuitive, defying our intuitions from low-dimensional geometry.]

Consider a random point $p \sim \mathcal{N}(0, I) \in \mathbb{R}^d$.
What is the distribution of its length?

[Looking at the one-dimensional normal distribution, you would expect it to be very common that the length is close to zero, a bit less common that the length is close to 1 or −1, and not rare for the length to be close to 2 or −2. But in high dimensions, that intuition is completely wrong.]



normal.pdf [A one-dimensional normal distribution.]

[If the dimension is very high, the vast majority of the random points are at approximately the same distance from the mean. So they lie in a thin shell. Why? To answer that, let's study the square of the distance. By Pythagoras' Theorem, the squared distance from p to the mean is]

$$\|p\|^2 = p_1^2 + p_2^2 + \ldots + p_d^2$$

[Each component p_i is sampled independently from a univariate normal distribution with mean zero and variance one. The square of a component, p_i^2, is said to come from a chi-squared distribution.]

$$p_i \sim \mathcal{N}(0,1), \quad p_i^2 \sim \chi^2(1), \quad E[p_i^2] = 1, \quad \mathrm{Var}(p_i^2) = 2$$

[Recall that when you add d independent, identically distributed random numbers, you scale their mean and variance by d, and the standard deviation is the square root of the variance.]

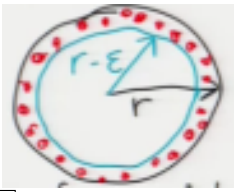
$$\begin{aligned} E[\|p\|^2] &= d\,E[p_1^2] = d \\ \mathrm{Var}(\|p\|^2) &= d\,\mathrm{Var}(p_1^2) = 2d \\ \mathrm{SD}(\|p\|^2) &= \sqrt{2d} \end{aligned}$$

For large d, $\|p\|$ is concentrated in a thin shell around radius $\sqrt{d}$ with a thickness proportional to $\sqrt[4]{2d}$.
[The mean value of $\|p\|$ isn't exactly $\sqrt{d}$, but it is close, because the mean of $\|p\|^2$ is d and the standard deviation is much, much smaller. Likewise, the standard deviation of $\|p\|$ isn't exactly $\sqrt[4]{2d}$, but it's close.]

[So if d is about a million, imagine a million-dimensional egg whose radius is about 1,000, and the thickness of the shell is about 67, which is about 10 times the standard deviation. The vast majority of random points are in the eggshell. Not inside the egg; actually in the shell itself. It is counterintuitive that random vectors sampled from a high-dimensional normal distribution almost all have almost the same length.]

[There is a statistical principle hiding here. Suppose you want to estimate the mean of a distribution—in this case, the chi-squared distribution. The standard way to do that is to sample very many numbers from the distribution and take their mean. The more numbers you sample, the more accurate your estimate is—that is, the smaller the standard deviation of your sample mean is. When we sample a vector from a million-dimensional normal distribution and compute its length, that's exactly what we're doing!]

What about a uniform distribution? Consider concentric spheres of radii r & $r - \epsilon$.



[Draw this by hand concentric.png] [Concentric balls. In high dimensions, almost every point chosen uniformly at random in the outer ball lies outside the inner ball.]

Volume of outer ball $\propto r^d$
Volume of inner ball $\propto (r-\epsilon)^d$
Ratio of inner ball volume to outer =

$$\frac{(r-\epsilon)^d}{r^d} = \left(1 - \frac{\epsilon}{r}\right)^d \approx \exp\left(-\frac{\epsilon d}{r}\right) \qquad \text{which is small for large } d.$$

E.g., if $\frac{\epsilon}{r} = 0.1$ & $d = 100$, inner ball has $0.9^{100} = 0.0027\%$ of volume.

Random points from uniform distribution in ball: nearly all are in outer shell.
" " " Gaussian " : nearly all are in some thin shell.

Lessons:
- In high dimensions, sometimes the nearest neighbor and 1,000th-nearest neighbor don't differ much!
- k-means clustering and nearest neighbor classifiers are less effective for large d.

Angles between Random Vectors

What is the angle θ between a random $p \sim \mathcal{N}(0, I) \in \mathbb{R}^d$ and an arbitrary $q \in \mathbb{R}^d$?

Without loss of generality, set $q = [1 \quad 0 \quad 0 \ldots 0]^\top$.
[The value of q doesn't matter, because the direction that p points in is uniformly distributed over all possible directions. By a formula we learned early this semester, the angle between p and q is θ, where . . .]

$$\cos\theta = \frac{p \cdot q}{\|p\|\,\|q\|} = \frac{p_1}{\|p\|}$$
$$E[\cos\theta] = 0; \quad \mathrm{Var}(\cos\theta) \approx \frac{1}{\sqrt{d}}$$

If d is large, $\cos\theta$ is almost always very close to zero; θ is almost always very close to 90°!

[In high-dimensional spaces, two random vectors are almost always very close to orthogonal. To put it another way, an arbitrary vector is almost orthogonal to the vast majority of all the other vectors!]

[A former CS 189/289A head TA, Marc Khoury, has a nice short essay entitled "Counterintuitive Properties of High Dimensional Space", which you can read at
https://marckhoury.github.io/blog/counterintuitive-properties-of-high-dimensional-space]

RANDOM PROJECTION

An alternative to PCA as preprocess for clustering, classification, regression.
Approximately preserves distances between points!

[We project onto a random subspace instead of the PCA subspace, but sometimes preserves distances better than PCA. It works best when you project a very high-dimensional space to a medium-dimensional space. Because it roughly preserves the distances, algorithms like k-means clustering and nearest neighbor classifiers will give similar results to what they would give in high dimensions, but they run much faster.]

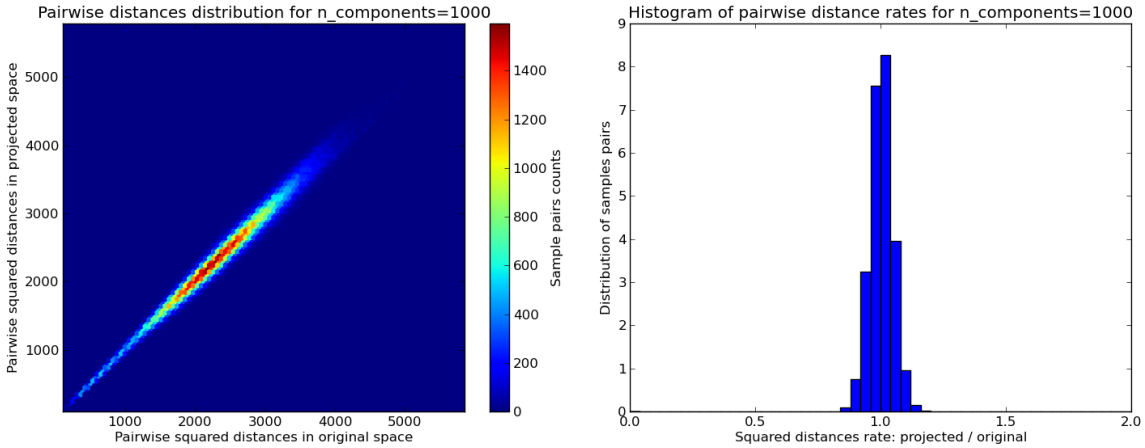
Pick a small ϵ, a small δ, and a random subspace $S \subset \mathbb{R}^d$ of dimension k, where $k = \left\lceil \frac{2\ln(1/\delta)}{\epsilon^2/2 - \epsilon^3/3} \right\rceil$.

For any pt q, let $\hat{q}$ be orthogonal projection of q onto S, multiplied by $\sqrt{\frac{d}{k}}$.

[The multiplication by $\sqrt{d/k}$ helps preserve the distances between points after you project.]

Johnson–Lindenstrauss Lemma (modified):
For any two pts $q, w \in \mathbb{R}^d$, $(1-\epsilon)\|q-w\|^2 \le \|\hat{q}-\hat{w}\|^2 \le (1+\epsilon)\|q-w\|^2$ with probability $\ge 1 - 2\delta$.
Typical values: $\epsilon \in [0.02, 0.5]$, $\delta \in [1/n^3, 0.05]$. [You choose ϵ and δ according to your needs.]

[With these ranges, the squared distance between two points after projecting might change by 2% to 50%. In practice, you can experiment with k to find the best speed-accuracy tradeoff. If you want all inter-sample-point distances to be accurate, you should set δ smaller than $1/n^2$, so you need a subspace of dimension $\Theta(\log n)$. Reducing δ doesn't cost much (because of the logarithm), but reducing ϵ costs more. You can bring 1,000,000 sample points down to a 10,000-dimensional space with at most a 6% error in the distances.]
[What is remarkable about this result is that the dimension d of the input points doesn't matter!]



100000to1000.pdf [Comparison of inter-point distances before and after projecting points in 100,000-dimensional space down to 1,000 dimensions.]

[Why does this work? A random projection of $q - w$ is like taking a random vector and selecting k components. The mean of the squares of those k components approximates the mean for the whole population.]

[How do you get a uniformly distributed random projection direction? You can choose each component from a univariate Gaussian distribution, then normalize the vector to unit length. How do you get a random subspace? You can choose k random directions, then use Gram–Schmidt orthogonalization to make them mutually orthonormal. Interestingly, Indyk and Motwani show that if you skip the expensive normalization and Gram–Schmidt steps, random projection still works almost as well, because random vectors in a high-dimensional space are nearly equal in length and nearly orthogonal to each other with high probability.]

THE PSEUDOINVERSE AND THE SVD

[We're done with unsupervised learning. For the rest of the semester, we go back to supervised learning.]

[The singular value decomposition can give us insight into the pseudoinverse and its use in least-squares linear regression. If you attended Discussion Section 6, you worked through an explanation of this, but now that I've introduced the SVD in Lecture 21, I'd like to summarize it.]

[Let's understand the psuedoinverse of a diagonal matrix first, then the pseudoinverse of a matrix in general.]

Let D be a diagonal $n \times d$ matrix. [Not necesarily square!]
Find its pseudoinverse D^+ by transposing D and replacing every nonzero entry with its reciprocal.

$$\text{E.g.,} \quad D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1/3 \\ 0 & 0 & 0 \end{bmatrix}, \quad D^+ = \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}, \quad DD^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D^+D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

[If D were a square matrix with no zeros on the diagonal, then D^+ would be the inverse of D, and DD^+ and D^+D would be the identity matrix. In general, DD^+ and D^+D are always diagonal matrices with 0's and 1's only.]

Observe that $DD^+D = D$ and $D^+DD^+ = D^+$ and $D^2D^+ = D$.
[Because the 0's and 1's in DD^+ line up with the 0's and nonzeros in D and D^+. This is as close to an "inverse" as a rank-deficient matrix can get. Now let's consider the pseudoinverse of an arbitrary matrix.]

Let X be any $n \times d$ matrix. Let $X = UDV^\top$ be its SVD. Recall that rank D = rank X.
The Moore–Penrose pseudoinverse of X is $X^+ = VD^+U^\top$.
Observe:

(1) $XX^+ = UDV^\top VD^+U^\top = U(DD^+)U^\top$ is symmetric & positive semidefinite.
    [Observe that this is an eigendecomposition of XX^+, and all the eigenvalues are 1 or 0.]
(2) $X^+X = VD^+U^\top UDV^\top = V(D^+D)V^\top$ is symmetric & PSD too.
(3) All have the same rank: D, D^+, DD^+, D^+D, X, X^+, XX^+, X^+X.
(4) If X has rank n, then $XX^+ = I_{n\times n}$ and X^+ is a right inverse.
(5) If X has rank d, then $X^+X = I_{d\times d}$ and X^+ is a left inverse.
(6) $XX^+X = X$. Proof: $XX^+X = U(DD^+)U^\top UDV^\top = U(DD^+D)V^\top = UDV^\top = X$.
(7) $X^+XX^+ = X^+$. [The proof is symmetric to the previous one.]

[Now, we can show that the pseudoinverse always gives a good solution in least-squares linear regression, even when $X^\top X$ is singular.]

Theorem: A solution to the normal equations $X^\top Xw = X^\top y$ is $w = X^+y$.

Proof: $X^\top Xw = X^\top XX^+y = VDU^\top U(DD^+)U^\top y = V(D^2D^+)U^\top y = VDU^\top y = X^\top y$.

If the normal eq'ns have multiple solutions, $w = X^+y$ is the least-norm solution; i.e., it minimizes $\|w\|$ among all solutions. [If you attended Discussion Section 6, you might have proven this yourself.]

[This way of solving the normal equations is very helpful when $X^\top X$ is singular because $n < d$ or the sample points lie on a subspace of the feature space. But observe that if X has a very small singular value, the reciprocal of that singular value will be very large and have a very large effect on w; but when that singular value is exactly zero, it has no effect on w! So when we have a really tiny singular value, should we pretend it is zero? Ridge regression implements this policy to some degree; review Discussion Worksheet 8 for details.]