TOPIC 2

Spectral and singular value decompositions

3 Singular value decomposition

Course: Math 535 (http://www.math.wisc.edu/~roch/mmids/) - Mathematical Methods in Data Science (MMiDS)
Author: Sebastien Roch (http://www.math.wisc.edu/~roch/), Department of Mathematics, University of Wisconsin-Madison
Updated: Sep 21, 2020

While solving the low-rank approximation problem in the previous section, we derived the building blocks of a matrix decomposition that has found many applications, the singular value decomposition (SVD) (https://en.wikipedia.org/wiki/Singular_value_decomposition). In this section, we define the SVD formally and describe a simple method to compute it. We also return to the application to dimensionality reduction.

3.1 Definition

First recall a matrix $D \in \mathbb{R}^{n\times m}$ is diagonal if its non-diagonal entries are zero. That is, $i \neq j$ implies that $D_{ij} = 0$. We now come to our main definition.

Definition (Singular Value Decomposition): Let $A \in \mathbb{R}^{n\times m}$ be a matrix with $m \leq n$. A singular value decomposition (SVD) of A is a matrix factorization

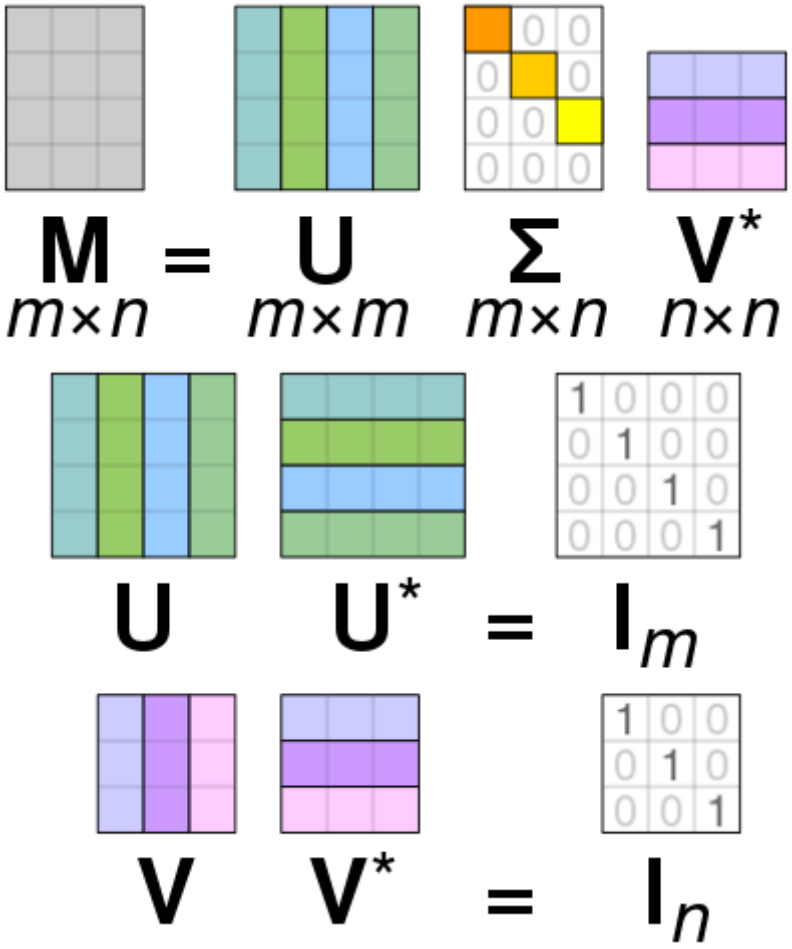
$$A = U\Sigma V^T = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

where the columns of $U \in \mathbb{R}^{n\times r}$ and those of $V \in \mathbb{R}^{m\times r}$ are orthonormal, and $\Sigma \in \mathbb{R}^{r\times r}$ is a diagonal matrix. Here the $\mathbf{u}_j$'s are the columns of U and are referred to as left singular vectors. Similarly the $\mathbf{v}_j$'s are the columns of V and are referred to as right singular vectors. The σ_j's, which are non-negative and in decreasing order

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

are the diagonal elements of Σ and are referred to as singular values. ◁

This type of matrix factorization is illustrated below (in its [full (https://en.wikipedia.org/wiki/Singular_value_decomposition#Reduced_SVDs)](https://en.wikipedia.org/wiki/Singular_value_decomposition#Reduced_SVDs) form).



([Source (https://upload.wikimedia.org/wikipedia/commons/c/c8/Singular_value_decomposition_visualisation.svg)](https://upload.wikimedia.org/wikipedia/commons/c/c8/Singular_value_decomposition_visualisation.svg))

Exercise: Let $A = U\Sigma V^T = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$ be an SVD of A.

(a) Show that $\{\mathbf{u}_j : j \in [r]\}$ is a basis of $\mathrm{col}(A)$ and that $\{\mathbf{v}_j : j \in [r]\}$ is a basis of $\mathrm{row}(A)$.

(b) Show that r is the rank of A. ◁

Remarkably, any matrix has an SVD.

Theorem (Existence of SVD): Any matrix $A \in \mathbb{R}^{n \times m}$ has a singular value decomposition.

The construction works as follows. Compute the greedy sequence $\mathbf{v}_1, \ldots, \mathbf{v}_r$ from the previous section until the first r such that

$$\max\{\|A\mathbf{v}\| : \|\mathbf{v}\| = 1,\ \langle \mathbf{v}, \mathbf{v}_j \rangle = 0, \forall j \leq r\} = 0$$

or, otherwise, until $r = m$. The $\mathbf{v}_j$'s are orthonormal by construction. For $j = 1, \ldots, m$, let

$$\sigma_j = \|A\mathbf{v}_j\| \quad \text{and} \quad \mathbf{u}_j = \frac{1}{\sigma_j} A\mathbf{v}_j.$$

Observe that, by our choice of r, the σ_j's are > 0. In particular, the $\mathbf{u}_j$'s have unit norm by definition. We show next that they are also orthogonal.

Lemma (Left Singular Vectors are Orthogonal): For all $1 \leq i \neq j \leq r$, $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$.

Proof idea: Quoting [BHK, Section 3.6]:

> Intuitively if $\mathbf{u}_i$ and $\mathbf{u}_j$, $i < j$, were not orthogonal, one would suspect that the right singular vector $\mathbf{v}_j$ had a component of $\mathbf{v}_i$ which would contradict that $\mathbf{v}_i$ and $\mathbf{v}_j$ were orthogonal. Let i be the smallest integer such that $\mathbf{u}_i$ is not orthogonal to all other $\mathbf{u}_j$. Then to prove that $\mathbf{u}_i$ and $\mathbf{u}_j$ are orthogonal, we add a small component of $\mathbf{v}_j$ to $\mathbf{v}_i$, normalize the result to be a unit vector $\mathbf{v}_i' \propto \mathbf{v}_i + \epsilon \mathbf{v}_j$ and show that $\|A\mathbf{v}_i'\| > \|A\mathbf{v}_i\|$, a contradiction.

Proof: We argue by contradiction. Let i be the smallest index such that there a $j > i$ such that $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \delta \neq 0$. Assume $\delta > 0$ (otherwise use $-\mathbf{u}_i$). For $\epsilon \in (0, 1)$, because the $\mathbf{v}_k$'s are orthonormal, $\|\mathbf{v}_i + \epsilon \mathbf{v}_j\|^2 = 1 + \epsilon^2$. Consider the vectors

$$\mathbf{v}_i' = \frac{\mathbf{v}_i + \epsilon \mathbf{v}_j}{\sqrt{1 + \epsilon^2}} \quad \text{and} \quad A\mathbf{v}_i' = \frac{\sigma_i \mathbf{u}_i + \epsilon \sigma_j \mathbf{u}_j}{\sqrt{1 + \epsilon^2}}.$$

Observe that $\mathbf{v}_i'$ is orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$, so that

$$\|A\mathbf{v}_i'\| \leq \|A\mathbf{v}_i\| = \sigma_i.$$

On the other hand, by the *Orthogonal Decomposition Lemma*, we can write $A\mathbf{v}_i'$ as a sum of its orthogonal projection on the unit vector $\mathbf{u}_i$ and $A\mathbf{v}_i' - \text{proj}_{\mathbf{u}_i}(A\mathbf{v}_i')$, which is orthogonal to $\mathbf{u}_i$. In particular, by *Pythagoras*, $\|A\mathbf{v}_i'\| \geq \|\text{proj}_{\mathbf{u}_i}(A\mathbf{v}_i')\|$. But that implies, for $\epsilon \in (0, 1)$,

$$\|A\mathbf{v}_i'\| \geq \|\text{proj}_{\mathbf{u}_i}(A\mathbf{v}_i')\| = \left\langle \mathbf{u}_i, A\mathbf{v}_i' \right\rangle = \frac{\sigma_i + \epsilon\sigma_j\delta}{\sqrt{1+\epsilon^2}} \geq (\sigma_i + \epsilon\sigma_j\delta)\,(1 - \epsilon^2/2)$$

where the second inequality follows from a Taylor expansion (https://en.wikipedia.org/wiki/Taylor_series) or the observation

$$(1+\epsilon^2)\,(1-\epsilon^2/2)^2 = (1+\epsilon^2)\,(1-\epsilon^2+\epsilon^4/4) = 1 - 3/4\epsilon^4 + \epsilon^6/4 \leq 1.$$

Now note that

$$\begin{aligned}
\|A\mathbf{v}_i'\| &\geq (\sigma_i + \epsilon\sigma_j\delta)\,(1 - \epsilon^2/2) \\
&= \sigma_i + \epsilon\sigma_j\delta - \epsilon^2\sigma_i/2 - \epsilon^3\sigma_i\sigma_j\delta/2 \\
&= \sigma_i + \epsilon\left(\sigma_j\delta - \epsilon\sigma_i/2 - \epsilon^2\sigma_i\sigma_j\delta/2\right) \\
&> \sigma_i
\end{aligned}$$

for ϵ small enough, contradicting the inequality above. □

Proof idea (Existence of SVD): We show that $\sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$ acts on vectors in the same way that A does, by an orthogonal decomposition over the span of the $\mathbf{v}_j$'s and its complement.

Proof (Existence of SVD): Let

$$C = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T.$$

We claim that $A = C$ as matrices. It suffices to show that the left-hand side and right-hand side are the same linear map, that is, $A\mathbf{v} = C\mathbf{v}$ for all $\mathbf{v} \in \mathbb{R}^m$. Indeed, one then has in particular that the i-th column in both cases is $A\mathbf{e}_i = C\mathbf{e}_i$ for all i.

Let $\mathbf{v} \in \mathbb{R}^m$ be any vector and let $\mathcal{Z} = \text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$. Decompose $\mathbf{v}$ into orthogonal components

$$\mathbf{v} = \text{proj}_{\mathcal{Z}}(\mathbf{v}) + (\mathbf{v} - \text{proj}_{\mathcal{Z}}(\mathbf{v})) = \sum_{j=1}^{r} \langle \mathbf{v}, \mathbf{v}_j \rangle \, \mathbf{v}_j + (\mathbf{v} - \text{proj}_{\mathcal{Z}}(\mathbf{v})).$$

Applying A we get

$$\begin{aligned} A\mathbf{v} &= \sum_{j=1}^{r} \langle \mathbf{v}, \mathbf{v}_j \rangle \, A\mathbf{v}_j + A(\mathbf{v} - \text{proj}_{\mathcal{Z}}(\mathbf{v})) \\ &= \sum_{j=1}^{r} \langle \mathbf{v}, \mathbf{v}_j \rangle \, \sigma_j \mathbf{u}_j + A(\mathbf{v} - \text{proj}_{\mathcal{Z}}(\mathbf{v})) \\ &= \sum_{j=1}^{r} \langle \mathbf{v}, \mathbf{v}_j \rangle \, \sigma_j \mathbf{u}_j \end{aligned}$$

where we used that, by definition of r, $A\mathbf{w} = 0$ for any $\mathbf{w}$ orthogonal to $\mathcal{Z}$, which includes $\mathbf{v} - \text{proj}_{\mathcal{Z}}(\mathbf{v})$ by the *Orthogonal Decomposition Lemma*.

Similarly,

$$\begin{aligned} \left(\sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \right) \mathbf{v} &= \left(\sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \right) \left(\sum_{k=1}^{r} \langle \mathbf{v}, \mathbf{v}_k \rangle \, \mathbf{v}_k + (\mathbf{v} - \text{proj}_{\mathcal{Z}}(\mathbf{v})) \right) \\ &= \sum_{j=1}^{r} \sum_{k=1}^{r} \langle \mathbf{v}, \mathbf{v}_k \rangle \, \sigma_j \mathbf{u}_j \, (\mathbf{v}_j^T \mathbf{v}_k) \\ &= \sum_{j=1}^{r} \langle \mathbf{v}, \mathbf{v}_j \rangle \, \sigma_j \mathbf{u}_j, \end{aligned}$$

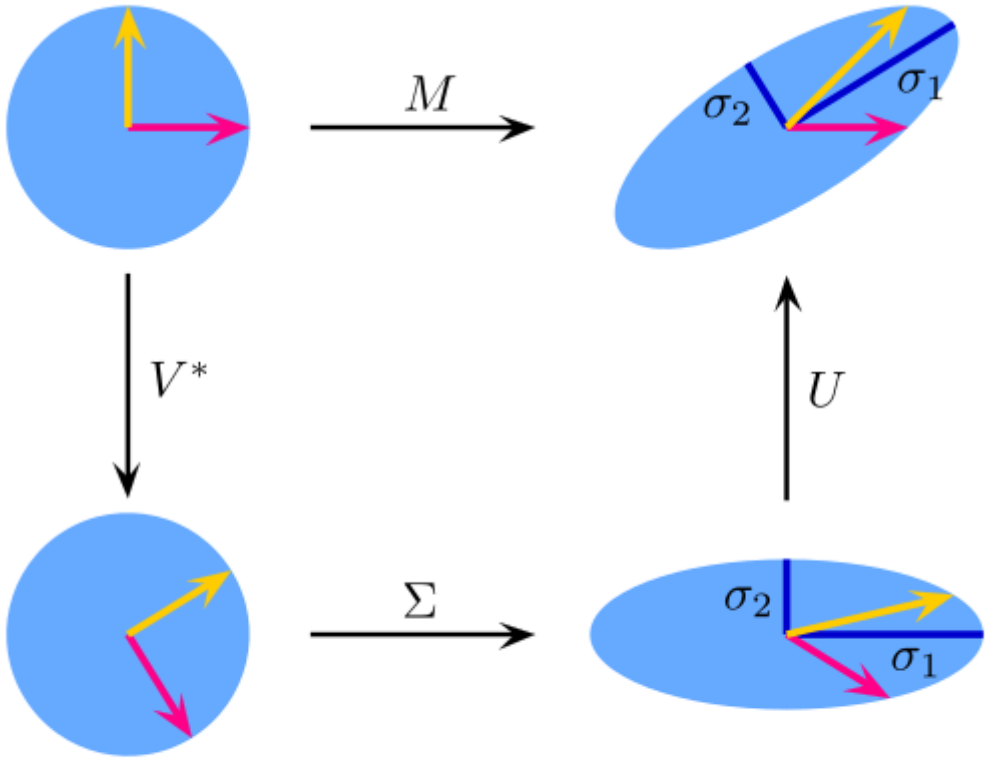
where, again, we used the *Orthogonal Decomposition Lemma* on the second line.

That establishes the claim. □

The SVD also has a natural geometric interpretation. To quote [Sol, p. 133]:

> The SVD provides a complete geometric characterization of the action of A. Since U and V are orthogonal, they have no effect on lengths and angles; as a diagonal matrix, Σ scales individual coordinate axes. Since the SVD always exists, all matrices $A \in \mathbb{R}^{n \times m}$ are a composition of an isometry, a scale in each coordinate, and a second isometry.

This sequence of operations is illustrated below.



$$M = U \cdot \Sigma \cdot V^*$$

(Source (https://commons.wikimedia.org/wiki/File:Singular-Value-Decomposition.svg))

3.2 Power iteration

There is in general no exact method (https://math.stackexchange.com/questions/2582300/what-does-the-author-mean-by-no-method-exists-for-exactly-computing-the-eigenva) for computing SVDs. Instead we must rely on iterative methods, that is, methods that approach the solution. Let $U\Sigma V^T$ be an SVD of A. To see how one might go about computing such an SVD, we start with the following observation. Because of the orthogonality of U and V, the powers of $A^T A$ have a simple representation. Indeed

$$B = A^T A = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^T,$$

so that

$$B^2 = (V\Sigma^T \Sigma V^T)(V\Sigma^T \Sigma V^T) = V(\Sigma^T \Sigma)^2 V^T$$

and repeating

$$B^k = V(\Sigma^T \Sigma)^k V^T.$$

Further,

$$\widetilde{\Sigma} = \Sigma^T \Sigma = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r^2 \end{pmatrix}$$

and

$$\widetilde{\Sigma}^k = \begin{pmatrix} \sigma_1^{2k} & 0 & \cdots & 0 \\ 0 & \sigma_2^{2k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r^{2k} \end{pmatrix}.$$

When $\sigma_1 > \sigma_2$, which is often the case with real datasets, we get that $\sigma_1^{2k} \gg \sigma_2^{2k}, \ldots, \sigma_r^{2k}$ when k is large. In that case, we get the approximation

$$B^k = \sum_{j=1}^{r} \sigma_j^{2k} \mathbf{v}_j \mathbf{v}_j^T . \approx \sigma_1^{2k} \mathbf{v}_1 \mathbf{v}_1^T .$$

This leads to the following idea to compute $\mathbf{v}_1$:

Lemma (Power Iteration): Let $A \in \mathbb{R}^{n \times m}$ be a matrix with $m \leq n$. Let $U \Sigma V^T$ be an SVD of A such that $\sigma_1 > \sigma_2$. Define $B = A^T A$ and assume that $\mathbf{x} \in \mathbb{R}^m$ is a vector satisfying $\langle \mathbf{v}_1, \mathbf{x} \rangle > 0$. Then

$$\frac{B^k \mathbf{x}}{\| B^k \mathbf{x} \|} \to \mathbf{v}_1$$

as $k \to +\infty$.

Proof idea: We use the approximation above and divide by the norm to get a unit norm vector in the direction of $\mathbf{v}_1$.

Proof: We have

$$B^k \mathbf{x} = \sum_{j=1}^{r} \sigma_j^{2k} \mathbf{v}_j \mathbf{v}_j^T \mathbf{x}$$

and, because the $\mathbf{v}_j$'s are an orthonormal basis,

$$\frac{1}{\sigma_1^{4k} (\mathbf{v}_1^T \mathbf{x})^2} \|B^k \mathbf{x}\|^2 = \frac{1}{\sigma_1^{4k} (\mathbf{v}_1^T \mathbf{x})^2} \sum_{j=1}^{r} \left|\sigma_j^{2k} (\mathbf{v}_j^T \mathbf{x})\right|^2 = 1 + \sum_{j=2}^{r} \frac{\sigma_j^{4k} (\mathbf{v}_j^T \mathbf{x})^2}{\sigma_1^{4k} (\mathbf{v}_1^T \mathbf{x})^2} \to 1$$

as $k \to +\infty$. Hence

$$\frac{B^k \mathbf{x}}{\|B^k \mathbf{x}\|} = \sum_{j=1}^{r} \mathbf{v}_j \frac{\sigma_j^{2k} (\mathbf{v}_j^T \mathbf{x})}{\|B^k \mathbf{x}\|} = \mathbf{v}_1 \frac{\sigma_1^{2k} (\mathbf{v}_1^T \mathbf{x})}{\|B^k \mathbf{x}\|} + \sum_{j=2}^{r} \mathbf{v}_j \frac{\sigma_j^{2k} (\mathbf{v}_j^T \mathbf{x})}{\|B^k \mathbf{x}\|} \to \mathbf{v}_1$$

as $k \to +\infty$, by the above and the assumption $\sigma_1 > \sigma_2$. □

That gives us a way to compute $\mathbf{v}_1$ (approximately). How do we find an appropriate vector $\mathbf{x}$? It turns out that a random vector will do. For instance, let $\mathbf{X}$ be an m-dimensional spherical Gaussian with mean 0 and variance 1. Then, $\mathbb{P}[\langle \mathbf{v}_1, \mathbf{X} \rangle] = 0$. We will show this when we discuss multivariate Gaussians. Note that, if $\langle \mathbf{v}_1, \mathbf{X} \rangle < 0$, we will instead converge to $-\mathbf{v}_1$ which is also a right singular vector.

How do we compute more singular vectors? One approach is to first compute $\mathbf{v}_1$ (or $-\mathbf{v}_1$), then find a vector $\mathbf{y}$ orthogonal to it, and proceed as above. And then we repeat until we have all m right singular vectors.

Exercise: Let A, $U \Sigma V^T$, B be as in the Power Iteration Lemma. Assume further that $\sigma_2 > \sigma_3$ and that $\mathbf{y} \in \mathbb{R}^m$ satisfies both $\langle \mathbf{v}_1, \mathbf{y} \rangle = 0$ and $\langle \mathbf{v}_2, \mathbf{y} \rangle > 0$. Show that $\frac{B^k \mathbf{y}}{\|B^k \mathbf{y}\|} \to \mathbf{v}_2$ as $k \to +\infty$. How would you find such a $\mathbf{y}$? ◁

But we are often interested only in the top, say $\ell < m$, singular vectors. An alternative approach in that case is to start with ℓ random vectors and, first, find an orthonormal basis for the space they span. Then to quote [BHK, Section 3.7.1]:

> Then compute B times each of the basis vectors, and find an orthonormal basis for the space spanned by the resulting vectors. Intuitively, one has applied B to a subspace rather than a single vector. One repeatedly applies B to the subspace, calculating an orthonormal basis after each application to prevent the subspace collapsing to the one dimensional subspace spanned by the first singular vector.

We will not prove here that this approach, known as orthogonal iteration, works. The proof is similar to that of the *Power Iteration Lemma*.

NUMERICAL CORNER We implement this last algorithm.

In [1]:

```
# Julia version: 1.5.1
using Plots, LinearAlgebra

function mmids_gramschmidt(A)
    n, m = size(A)
    Q = zeros(Float64, n, m)
    R = zeros(Float64, m, m)
    for j = 1:m
        v = A[:,j]
        for k = 1:j-1
            R[k,j] = dot(Q[:,k],A[:,j])
            v -= R[k,j]*Q[:,k]
        end
        R[j,j] = norm(v)
        Q[:,j] = v/R[j,j]
    end
    return Q, R
end

function two_clusters(d, n, offset)
    X1 = reduce(hcat, [vcat(-offset, zeros(d-1)) .+ randn(d) for i=1:n])
'
    X2 = reduce(hcat, [vcat(offset, zeros(d-1)) .+ randn(d) for i=1:n])'
    return X1, X2
end

function opt_clust(X, k, reps)
    n, d = size(X) # n=number of rows, d=number of columns
    dist = zeros(Float64, n) # distance to rep
    assign = zeros(Int64, n) # cluster assignments
    for i = 1:n
        dist[i], assign[i] = findmin([norm(X[i,:] .- reps[j,:]) for j=1:
k])
    end
    @show G = sum(dist.^2)
    return assign
end

function opt_reps(X, k, assign)
    n, d = size(X)
    reps = zeros(Float64, k, d) # rows are representatives
    for j = 1:k
        in_j = [i for i=1:n if assign[i] == j]
        reps[j,:] = sum(X[in_j,:],dims=1) ./ length(in_j)
    end
    return reps
end

function mmids_kmeans(X, k; maxiter=10)
    n, d = size(X)
    assign = [rand(1:k) for i in 1:n] # start with random assignments
    reps = zeros(Int64, k, d) # initialization of reps
    for iter = 1:maxiter

        # Step 1: Optimal representatives for fixed clusters (Lemma 1)
        reps = opt_reps(X, k, assign)
```

```
        # Step 2: Optimal clusters for fixed representatives (Lemma 2)
        assign = opt_clust(X, k, reps)
    end
    return assign
end
```

Out[1]: mmids_kmeans (generic function with 1 method)

In [2]:
```
function mmids_svd(A, l; maxiter=100)
    V = randn(size(A,2),l) # random initialization
    for _ = 1:maxiter
        W = A * V
        Z = A' * W
        V, R = mmids_gramschmidt(Z)
    end
    W = A * V
    S = [norm(W[:, i]) for i=1:size(W,2)] # singular values
    U = reduce(hcat,[W[:,i]/S[i] for i=1:size(W,2)]) # left singular vec
tors
    return U, S, V
end
```

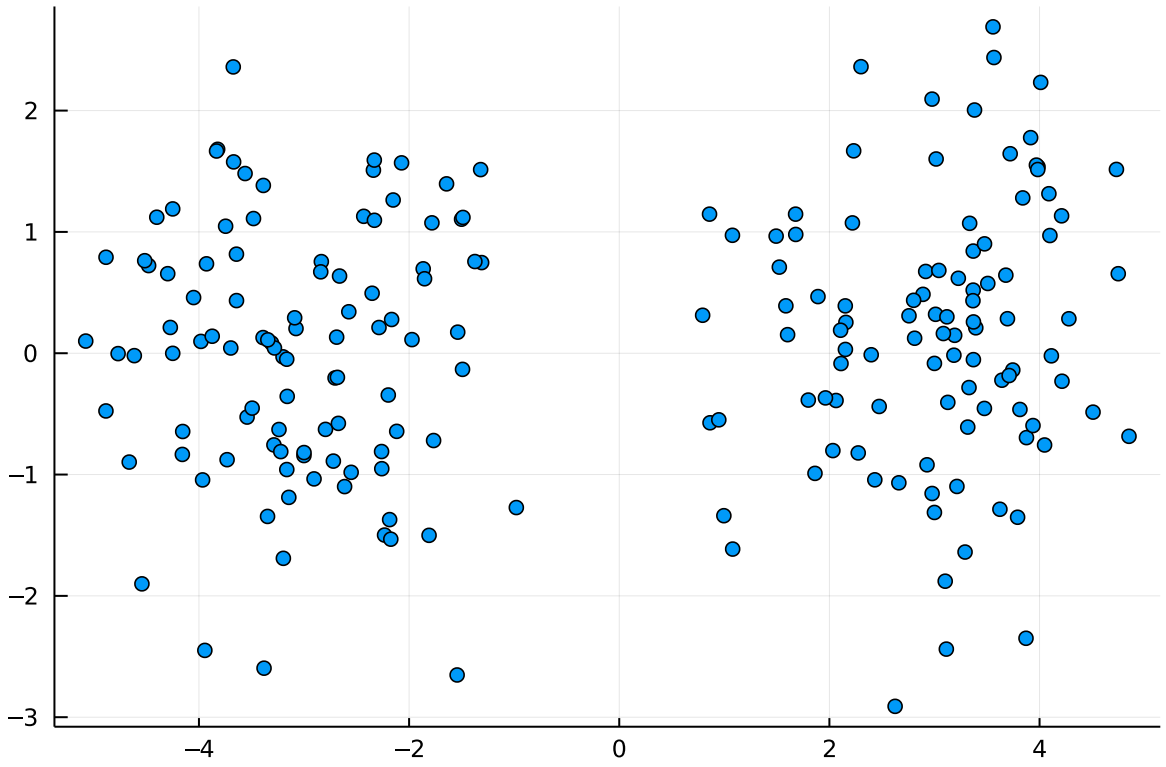
Out[2]: mmids_svd (generic function with 1 method)

Note above that we avoided forming the matrix $A^T A$. With a small number of iterations, that approach potentially requires fewer arithmetic operations overall and it allows to take advantage of the possible sparsity of A.

We will apply it to our previous two-cluster example. Let's compute the top singular vector.

In [3]:
```
d, n, offset = 10, 100, 3
X1, X2 = two_clusters(d, n, offset)
X = vcat(X1, X2)
scatter(X[:,1], X[:,2], legend=false)
```

Out[3]:



In [4]:
```
U, S, V = mmids_svd(X, 1)
V
```

Out[4]:
```
10×1 Array{Float64,2}:
  0.9962548680669375
  0.03175449922317592
 -0.046035453904927505
  0.025825801595636282
 -0.04449037015490855
 -0.029726316647995778
 -0.011654005208952954
 -0.015931727206384222
  0.015283439352855857
 -0.013978138721069698
```

This is approximately $\mathbf{e}_1$. We get roughly the same answer (possibly up to sign) from Julia's `svd` (https://docs.julialang.org/en/v1/stdlib/LinearAlgebra/#LinearAlgebra.svd) function.

In [5]:
```
F = svd(X)
F.V[:,1]
```

Out[5]:
```
10-element Array{Float64,1}:
  0.996254868066938
  0.03175449922317211
 -0.04603545390492926
  0.025825801595636723
 -0.04449037015490698
 -0.02972631664799404
 -0.011654005208954031
 -0.01593172720638476
  0.015283439352854134
 -0.013978138721068782
```

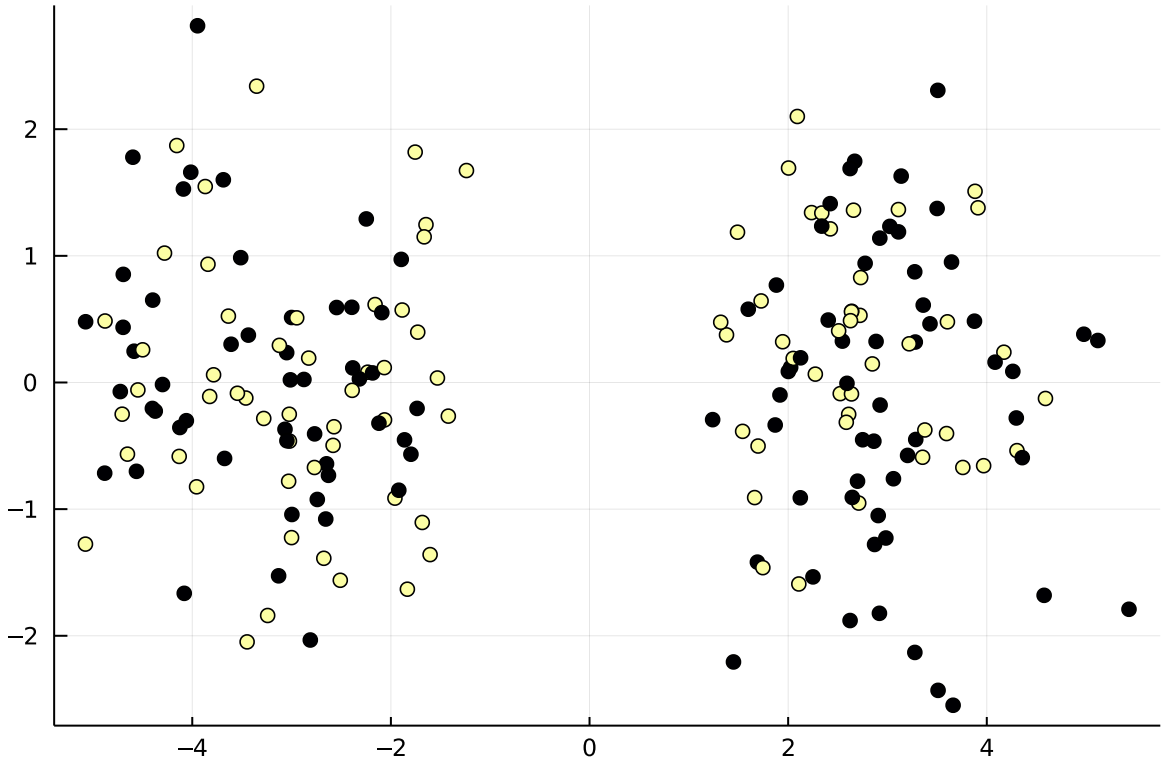
Recall that, when we applied k-means clustering to this example with $d = 1000$ dimension, we obtained a very poor clustering.

In [6]:
```
d, n, offset = 1000, 100, 3
X1, X2 = two_clusters(d, n, offset)
X = vcat(X1, X2)
assign = mmids_kmeans(X, 2);
```

```
G = sum(dist .^ 2) = 200285.18553307335
G = sum(dist .^ 2) = 200226.174659831
G = sum(dist .^ 2) = 200206.05494999306
G = sum(dist .^ 2) = 200206.05494999306
G = sum(dist .^ 2) = 200206.05494999306
G = sum(dist .^ 2) = 200206.05494999306
G = sum(dist .^ 2) = 200206.05494999306
G = sum(dist .^ 2) = 200206.05494999306
G = sum(dist .^ 2) = 200206.05494999306
G = sum(dist .^ 2) = 200206.05494999306
```

In [7]:
```
scatter(X[:,1], X[:,2], marker_z=assign, legend=false)
```

Out[7]:



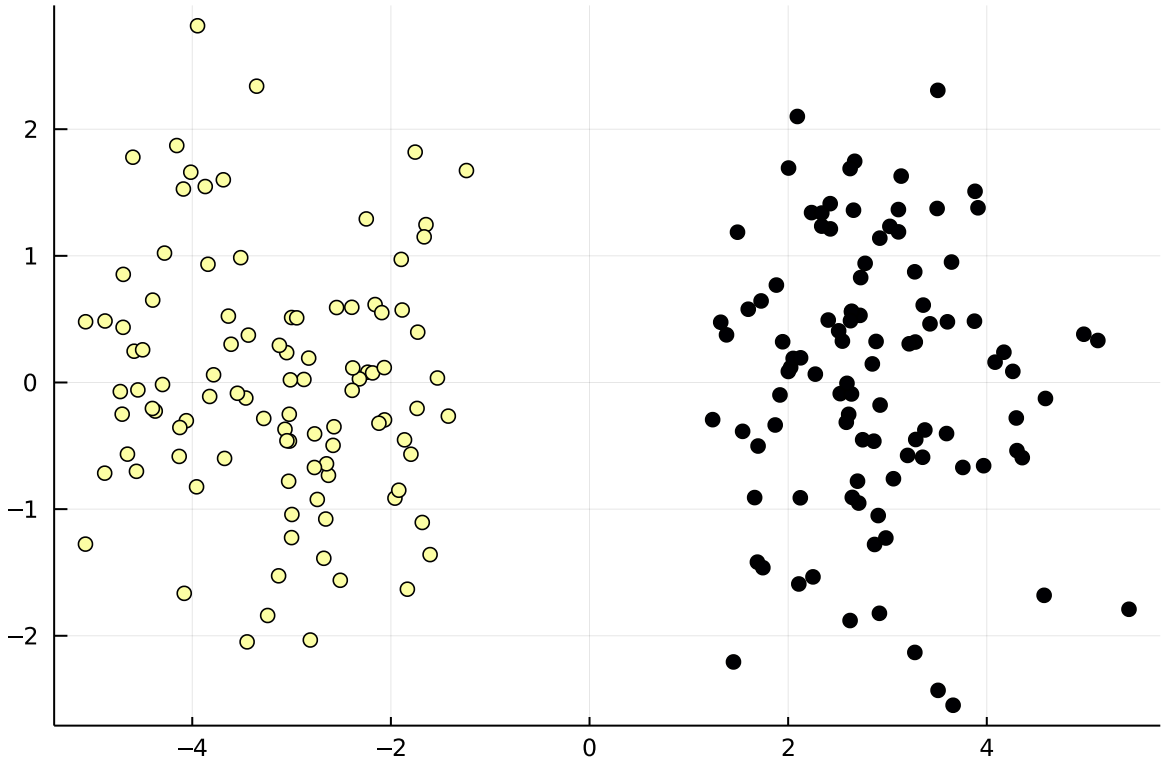
Let's try again, but after projecting on the top two singular vectors.

In [8]:
```
U, S, V = mmids_svd(X, 2)
Xproj = hcat(U[:,1].*S[1], U[:,2].*S[2])
assign = mmids_kmeans(Xproj, 2);
```

```
G = sum(dist .^ 2) = 4242.179608092223
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
G = sum(dist .^ 2) = 2467.0066479604775
```

In [9]:
```
scatter(X[:,1], X[:,2], marker_z=assign, legend=false)
```

Out[9]:



Better. We will give an explanation of this outcome below.

Finally, looking at the first two right singular vectors, we see that the first one does align quite well with the first dimension.

```
In [10]: hcat(V[:,1], V[:,2])
```

```
Out[10]: 1000×2 Array{Float64,2}:
 -0.789804      0.00688204
  0.0164863     0.0344699
 -0.00972216    0.0453034
  0.0228214    -0.0326633
  0.0141143     0.0211983
 -0.0135686     0.00555584
  0.00201696    0.0364882
 -0.000327212   0.0232046
  0.0241117     0.00355371
 -0.0178569     0.0485514
  0.0251782    -0.0305342
 -0.00246446    0.0167164
 -0.00569657    0.0260144
  ⋮
 -0.0184699     0.0056766
  0.00807559    0.0432008
 -0.0158732     0.00637732
 -8.32403e-5   -0.0512078
 -0.0133573    -0.0324199
  0.0246308    -0.046561
 -0.00152324   -0.0598089
 -0.0389301     0.0534447
  0.0234557     0.0429289
 -0.0317777    -0.0277843
  0.0188579     0.042175
  0.0029277     0.0249029
```

3.3 Low-rank approximation in the induced norm

Let $A \in \mathbb{R}^{n\times m}$ be a matrix with SVD

$$A = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T .$$

For $k < r$, truncate the sum at the k-th term

$$A_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T .$$

The rank of A_k is exactly k. Indeed, by construction,

1. the vectors $\{\mathbf{u}_j \,:\, j = 1, \ldots, k\}$ are orthonormal, and
2. since $\sigma_j > 0$ for $j = 1, \ldots, k$ and the vectors $\{\mathbf{v}_j \,:\, j = 1, \ldots, k\}$ are orthonormal, $\{\mathbf{u}_j \,:\, j = 1, \ldots, k\}$ spans the column space of A_k.

Exercise: Give the details of the previous argument. ◁

We have shown before that A_k is the best approximation to A among matrices of rank at most k in Frobenius norm. Specifically, the *Greedy Finds Best Fit Theorem* implies that, for any matrix $B \in \mathbb{R}^{n\times m}$ of rank at most k,

$$\|A - A_k\|_F \leq \|A - B\|_F.$$

We show in this section that the same holds in the induced norm. First, some observations.

Lemma (Matrix Norms and Singular Values): Let $A \in \mathbb{R}^{n\times m}$ be a matrix with SVD

$$A = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

where recall that $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r > 0$ and let A_k be the truncation defined above. Then

$$\|A - A_k\|_F^2 = \sum_{j=k+1}^{r} \sigma_j^2$$

and

$$\|A - A_k\|_2^2 = \sigma_{k+1}^2.$$

Proof: For the first claim, by definition, summing over the columns of $A - A_k$

$$\|A - A_k\|_F^2 = \left\| \sum_{j=k+1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \right\|_F^2 = \sum_{i=1}^{m} \left\| \sum_{j=k+1}^{r} \sigma_j v_{j,i} \mathbf{u}_j \right\|^2.$$

Because the $\mathbf{u}_j$'s are orthonormal, this is

$$\sum_{i=1}^{m} \sum_{j=k+1}^{r} \sigma_j^2 v_{j,i}^2 = \sum_{j=k+1}^{r} \sigma_j^2 \left(\sum_{i=1}^{m} v_{j,i}^2 \right) = \sum_{j=k+1}^{r} \sigma_j^2$$

where we used that the $\mathbf{v}_j$'s are also orthonormal.

For the second claim, recall that the induced norm is defined as

$$\|B\|_2 = \max_{\mathbf{x}\in\mathbb{S}^{m-1}} \|B\mathbf{x}\|.$$

For any $\mathbf{x} \in \mathbb{S}^{m-1}$

$$\|(A - A_k)\mathbf{x}\|^2 = \left\| \sum_{j=k+1}^{r} \sigma_j \mathbf{u}_j (\mathbf{v}_j^T \mathbf{x}) \right\|^2 = \sum_{j=k+1}^{r} \sigma_j^2 \langle \mathbf{v}_j, \mathbf{x} \rangle^2.$$

Because the σ_j's are in decreasing order, this is maximized when $\langle \mathbf{v}_j, \mathbf{x} \rangle = 1$ if $j = k+1$ and 0 otherwise. That is, we take $\mathbf{x} = \mathbf{v}_{k+1}$ and the norm is then σ_{k+1}^2, as claimed. □

Theorem (Low-Rank Approximation in the Induced Norm): Let $A \in \mathbb{R}^{n \times m}$ be a matrix with SVD

$$A = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

and let A_k be the truncation defined above with $k < r$. For any matrix $B \in \mathbb{R}^{n \times m}$ of rank at most k,

$$\|A - A_k\|_2 \leq \|A - B\|_2.$$

Proof idea: We know that $\|A - A_k\|_2^2 = \sigma_{k+1}^2$. So we want to lower bound $\|A - B\|_2^2$ by σ_{k+1}^2. For that, we have to find an appropriate $\mathbf{z}$ for any given B of rank at most k. The idea is to take a vector $\mathbf{z}$ in the intersection of the null space of B and the span of the singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}$. By the former, the squared norm of $(A - B)\mathbf{z}$ is equal to the squared norm of $A\mathbf{z}$ which lower bounds $\|A\|_2^2$. By the latter, $\|A\mathbf{z}\|^2$ is at least σ_{k+1}^2.

Proof: By the *Nullity Lemma*, the dimension of $\text{null}(B)$ is at least $n - k$ so there is a unit vector $\mathbf{z}$ in the intersection

$$\mathbf{z} \in \text{null}(B) \cap \text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}).$$

Then $(A - B)\mathbf{z} = A\mathbf{z}$ since $\mathbf{z} \in \text{null}(B)$. Also since $\mathbf{z} \in \text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_{k+1})$, and therefore orthogonal to $\mathbf{v}_{k+2}, \ldots, \mathbf{v}_r$, we have

$$\begin{aligned}
\|(A - B)\mathbf{z}\|^2 &= \|A\mathbf{z}\|^2 \\
&= \left\| \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \mathbf{z} \right\|^2 \\
&= \left\| \sum_{j=1}^{k+1} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \mathbf{z} \right\|^2 \\
&= \sum_{j=1}^{k+1} \sigma_j^2 \langle \mathbf{v}_j, \mathbf{z} \rangle^2 \\
&\geq \sigma_{k+1}^2 \sum_{j=1}^{k+1} \langle \mathbf{v}_j, \mathbf{z} \rangle^2 \\
&= \sigma_{k+1}^2.
\end{aligned}$$

By the *Matrix Norms and Singular Values Lemma*, $\sigma_{k+1}^2 = \|A - A_k\|_2$ and we are done. □

3.4 Why project?

We return to k-means clustering and why projecting to a lower-dimensional subspace can produce better results. We prove a simple inequality that provides some insight. Quoting [BHK, Section 7.5.1]:

> [...] let's understand the central advantage of doing the projection to [the top k right singular vectors]. It is simply that for any reasonable (unknown) clustering of data points, the projection brings data points closer to their cluster centers.

To elaborate, suppose we have n data points in d dimension in the form of the rows $\mathbf{a}_i^T$, $i = 1 \ldots, n$, of matrix $A \in \mathbb{A}^{n \times d}$, where we assume that $n > d$ and that A has full column rank. Imagine these data points come from an unknown ground-truth k-clustering assignment $g(i) \in [k]$, $i = 1, \ldots, n$, with corresponding unknown centers $\mathbf{c}_j$, $j = 1, \ldots, k$, for $g(i) = j$. Let $C \in \mathbb{R}^{n \times d}$ be the corresponding matrix, that is, row i of C is $\mathbf{c}_j^T$ if $g(i) = j$. The k-means objective of the true clustering is then

$$\begin{aligned}
\sum_{j \in [k]} \sum_{i : g(i) = j} \|\mathbf{a}_i - \mathbf{c}_j\|^2 &= \sum_{j \in [k]} \sum_{i : g(i) = j} \sum_{\ell = 1}^{d} (a_{i,\ell} - c_{j,\ell})^2 \\
&= \sum_{i=1}^{n} \sum_{\ell = 1}^{d} (a_{i,\ell} - c_{g(i),\ell})^2 \\
&= \|A - C\|_F^2.
\end{aligned}$$

The matrix A has an SVD

$$A = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

and for $k < r$ we have the truncation

$$A_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T .$$

It corresponds to projecting each row of A onto the linear subspace spanned by the first k right singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$. To see this, note that the i-th row of A is $\boldsymbol{\alpha}_i^T = \sum_{j=1}^{r} \sigma_j u_{j,i} \mathbf{v}_j^T$ and that, because the $\mathbf{v}_j$'s are linearly independent and in particular $\mathbf{v}_1, \ldots, \mathbf{v}_k$ is an orthonormal basis of its span, the projection of $\boldsymbol{\alpha}_i$ onto $\mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ is

$$\sum_{\ell=1}^{k} \left\langle \sum_{j=1}^{r} \sigma_j u_{j,i} \mathbf{v}_j, \mathbf{v}_\ell \right\rangle = \sum_{j=1}^{k} \sigma_j u_{j,i} \mathbf{v}_j$$

which is the i-th row of A_k. The k-means objective of A_k with respect to the ground-truth centers $\mathbf{c}_j$, $j = 1, \ldots, k$, is $\|A_k - C\|_F^2$.

One more observation: the rank of C is at most k. Indeed, there are k different rows in C so its row rank is k if these different rows are linearly independent and less than k otherwise.

Exercise: Show that for any matrices $A, B \in \mathbb{R}^{n \times m}$, the rank of the sum is less or equal than the sum of the ranks, that is,

$$\mathrm{rk}(A + B) \leq \mathrm{rk}(A) + \mathrm{rk}(B).$$

[Hint: Show that $\mathrm{col}(A + B) \subseteq \mathrm{col}(A) \cup \mathrm{col}(B)$.] ◁

Theorem (Why Project): Let $A \in \mathbb{A}^{n \times d}$ be a matrix and let A_k be the truncation above. For any matrix $C \in \mathbb{R}^{n \times d}$ of rank $\leq k$,

$$\|A_k - C\|_F^2 \leq 8k \|A - C\|_2^2.$$

The content of this inequality is the following. The quantity $\|A_k - C\|_F^2$ is the k-means objective of the projection A_k with respect to the true centers, that is, the sum of the squared distances to the centers. By the *Matrix Norms and Singular Values Lemma*, the inequality above gives that

$$\|A_k - C\|_F^2 \leq 8k\sigma_1(A - C)^2,$$

where $\sigma_j(A - C)$ is the j-th singular value of $A - C$. On the other hand, by the same lemma, the k-means objective of the un-projected data is

$$\|A - C\|_F^2 = \sum_{j=1}^{\mathrm{rk}(A-C)} \sigma_j(A - C)^2.$$

If the rank of $A - C$ is much larger than k and the singular values of $A - C$ decay slowly, then the latter quantity may be much larger. In other words, projecting may bring the data points closer to their true centers, potentially making it easier to cluster them.

Proof (Why Project): By the exercise preceding the statement, the rank of the difference $A_k - C$ is at most the sum of the ranks

$$\mathrm{rk}(A_k - C) \leq \mathrm{rk}(A_k) + \mathrm{rk}(-C) \leq 2k$$

where we used that the rank of A_k is k and the rank of C is $\leq k$ since it has k distinct rows. So by the *Matrix Norms and Singular Values Lemma*,

$$\|A_k - C\|_F^2 \leq 2k\|A_k - C\|_2^2.$$

By the triangle inequality for matrix norms,

$$\|A_k - C\|_2 \leq \|A_k - A\|_2 + \|A - C\|_2.$$

By the *Low-Rank Approximation in the Induced Norm Theorem*,

$$\|A - A_k\|_2 \leq \|A - C\|_2$$

since C has rank at most k. Putting these three inequalities together,

$$\|A_k - C\|_F^2 \leq 2k(2\|A - C\|_2)^2 = 8k\|A - C\|_2^2.$$

□

NUMERICAL CORNER We return to our example with the two Gaussian clusters.

In [11]:
```
d, n, offset = 1000, 100, 3
X1, X2 = two_clusters(d, n, offset)
X = vcat(X1, X2);
```

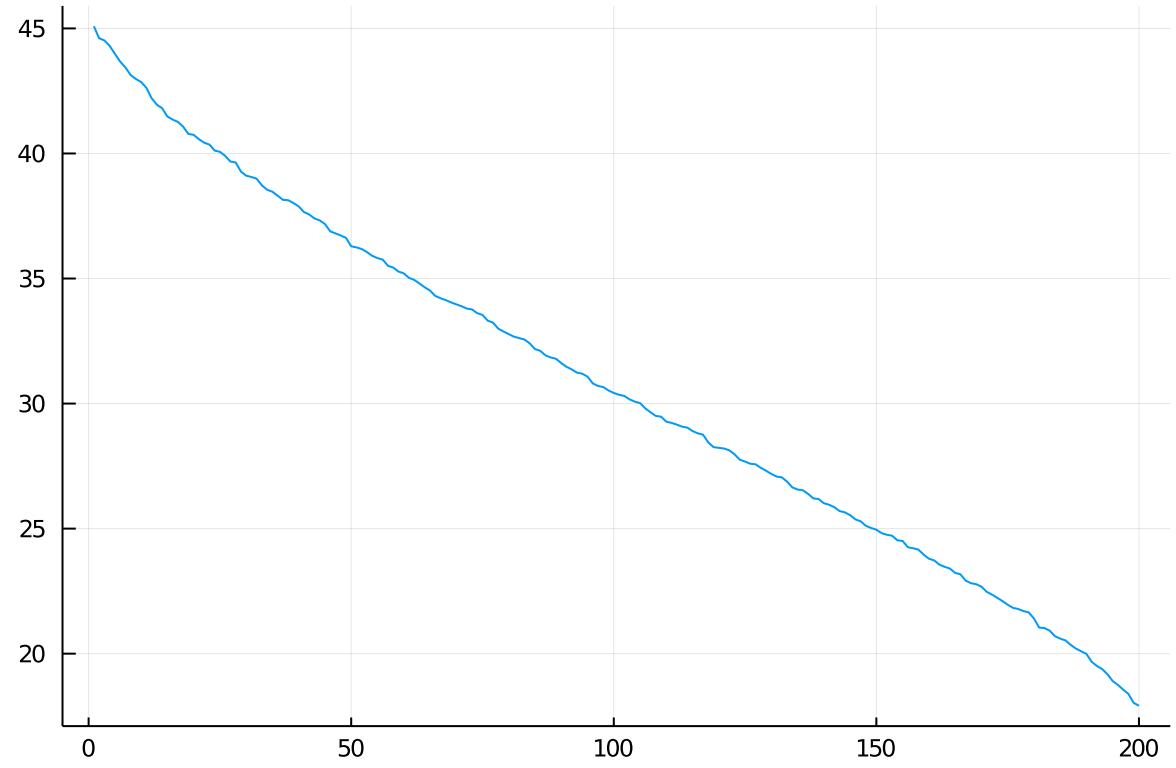
In reality, we cannot compute the matrix norms of $X - C$ and $X_k - C$ as the true centers are not known. But, because this is simulated data, we happen to know the truth and we can check the validity of our results in this case. The centers are:

In [12]:
```
C1 = reduce(hcat, [vcat(-offset, zeros(d-1)) for i=1:n])'
C2 = reduce(hcat, [vcat(offset, zeros(d-1)) for i=1:n])'
C = vcat(C1, C2);
```

We use Julia's `svd` function to compute the norms from the formulas in the Matrix Norms and Singular Values Lemma. First, we observe that the singular values of $X - C$ are decaying slowly.

In [13]:
```
Fc = svd(X-C)
plot(Fc.S, legend=false)
```

Out[13]:



The k-means objective with respect to the true centers under the full-dimensional data is:

In [14]:
```
frob = sum((Fc.S).^2)
```

Out[14]:
```
198953.15831164474
```

while the square of the top singular value of $X - C$ is only:

In [15]:
```
top_sval_sq = (Fc.S[1])^2
```

Out[15]:
```
2032.8205617726574
```

Finally, we compute the k-means objective with respect to the true centers under the projected one-dimensional data:

In [16]:
```
F = svd(X)
frob_proj = sum((F.S[1] * F.U[:,1] * F.Vt[1,:]' - C).^2)
```

Out[16]:
```
1477.471629781237
```