Contents

```
clear all
close all
```

Setup

```
load embeddings

fid = fopen('wordlist.txt');
data = textscan(fid,'%s');
fclose(fid);
words = data{1};
m = length(words);
embeddings = embeddings(1:m, :);
```

Map to 2D space

```
[U,S,V] = svds(embeddings,2);
emb2d = U*sqrt(S);
```

Initial Visualization

```
% figure(1)
% clf
% plot(emb2d(:,1),emb2d(:,2),'linestyle','none')
% hold on
% text(emb2d(:,1),emb2d(:,2), words)
% hold off
%
% fid = fopen('plotwords.txt');
% data = textscan(fid,'%s');
% fclose(fid);
% plotwords = data{1};
% toplot = false(n,1); %n is the number of words
% for k = 1:n
% word = words{k};
% toplot(k) = sum(strcmpi(word,plotwords))>0;
% end
% figure(1)
% clf
% plot(emb2d(toplot,1),emb2d(toplot,2),'linestyle','none')
% hold on
% text(emb2d(toplot,1),emb2d(toplot,2), words(toplot))
% hold off
```

K-means

```
n = m;
k = 1000;
d = 50;
X = embeddings;
```

```
P = randomP(n, k);
a = min(X(:));
b = max(X(:));
C = a + (b-a).*rand(k,d);

% figure(1)
pf = @(h, C, P) plotFunc(h, X, C, P, 0); % Putting a zero means do not plot
[C, P] = k_means(X, k, 100, C, P, pf);
```

3. Analysis

```
words = data{1};
topN = 100;
[M, ind] = maxk(sum(P), topN);
word_clusters = {};
for c = ind
    word_ind = find(P(:, c));
    word_cluster = words(word_ind);
    word_clusters{end+1} = word_cluster;
end

close all
```

Interesting Clusters

Biology

cell, cells, gene, dna, protein, nerve, genes, proteins, molecular, enzyme, viruses, molecules, organisms, genome, viral, synthesis, sperm, atom, membrane, receptor, acids, bind, domains, rna, bacterial, activation, receptors, nodes, peripheral, amino, node, neural, metabolism, transcription, antibodies, replication, antibody, kinase, mrna

Image and video

screen, images, camera, cameras, screens, lens, portable, projection, recorder, printer, stereo, televisions, lenses, disks, printers, tvs, recorders, lcd, handheld, scanner, scanners, zoom, pixels, hdtv, projector, cgi, projectors, widescreen, stylus, camcorder, camcorders, inkjet, epson, jpeg, dv, tft, gif

Spanish

el, en, y, o, que, se, sin, ha, mi, su, con, es, lo, para, una, por, mas, si, ya, dice, yo, ne, ri, tu, latina, ser, je, ver, filme

Different Initializations

I) Originally, I had all my clusters starting out near the mean of the dataset but this wasn't that effective because there would be tons of centroids near the center of the dataset but only a few around the outsides. Resulting in very big or very small (essentially zero item clusters).
I found a uniform distribution between the minimum and maximum values of the dataset worked best

II) I found that 10 iterations worked pretty well, a 100 definitely took too long, so I prefer the smaller iterations but the ability to run the entire algorithm multiple times instead

III) Using the emb2d instead embeddings unsuprisely, gave me much less sensible clusters and groupings, but it did allow me to visualize my work better

Published with MATLAB® R2018a

```
function [C, P] = k_means(X, k, max_iter, C, P, pf)
%UNTITLED5 Summary of this function goes here
%   Detailed explanation goes here
[n,d] = size(X);
h = 0;
for i = 1:max_iter
    P = calcP(X, C, n, d, k);
    C = calcC(X, P, n, d, k);
    h = pf(h,C,P);
end

end

function [P] = calcP(X, C, n, ~, k)
P = zeros(n, k);
for i = 1:n
        [~, min_k] = min(vecnorm(X(i, :) - C, 2, 2));
        P(i, min_k) = 1;
end
end

function [C] = calcC(X, P, ~, d, k)
C = zeros(k, d);
for ki = 1:k
    p = P(:, ki);
    sp = sum(p);
    if sp ~= 0
        C(ki, :) = p'*X/sp;
    end
end
end
```

Published with MATLAB® R2018a

```
function [h] = plotFunc(h, X, C, P, show)
    % Helper function to plot on each iteration of k-means
    if show == 0
        return
    end

    [n, ~] = size(X);
    [k, ~] = size(C);
    M = repmat(1:1:k,n, 1);
    clusters = sum(M.*P, 2);
    if h ~= 0
        delete(h)
        h = scatter(C(:,1), C(:,2), 'ko', 'filled');
        scatter(X(:, 1), X(:, 2), 20, clusters);
        pause(1);
    else
        h = scatter(C(:,1), C(:,2), 'ko', 'filled');
        scatter(X(:, 1), X(:, 2), 20, clusters);
    end
    colormap(jet(5))
end
```

Published with MATLAB® R2018a

```
function [P] = randomP(n, k)
    P = zeros(n, k);
    selections = randi([1, k], n, 1);
    for i = 1:n
        P(i, selections(i)) = 1;
    end
end
```

Published with MATLAB® R2018a

```
close all

d = 2;
n = 20;
k = 3;
X = rand(n, d)*20 - 20;
P = randomP(n, k);
C = rand(k, d)*mean2(X) - std2(X);

figure(1);
pf = @(h, C, P) plotFunc(h, X, C, P);
hold on
[C, P] = k_means(X, 3, 10, C, P, pf);
hold off
```

Published with MATLAB® R2018a