1 Von Mises distribution

Intuitively, the Von Mises distribution [?] is a simple approximation for the normal distribution on a circle (known as the *wrapped normal distribution*). The Von Mises probability density function is defined by the following equation:

$$P(x) = \frac{e^{b\cos(x-a)}}{2\pi I_0(b)}$$

where $I_0(x)$ is the modified Bessel function of the first kind; the Von Mises cumulative density function has no closed form.

The mean $\mu = a$ (intuitively, the angle that the distribution clustered around), and the circular variance $\sigma^2 = 1 - \frac{I_1(b)}{I_0(b)}$ (intuitively, b is the "concentration" parameter). Therefore, as $b \to 0$, the distribution becomes uniform; as $b \to \infty$, the distribution becomes normal with $\sigma^2 = 1/b$.

2 Von Mises-Fisher distribution

The Von-Mises Fisher distribution is the generalization of the Von Mises distribution to n-dimensional hyperspheres. It reduces to the Von-Mises distribution with $n = 2$. The probability density function is defined by the following equation:

$$f_p(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = C_p(\kappa)\exp(\kappa\boldsymbol{\mu}^T\boldsymbol{x})$$

where

$$C_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^p I_{p/2-1}(\kappa)}$$

and intuitively is an approximation for the normal distribution on the hypersphere.

2.1 Implementation of random spherical data in MATLAB

By using the Circular Statistics Toolbox in MATLAB, we can construct random spherical data through exploitation of the of Von Mises (circular) distribution.

```
%-------------------------------------------------------------------------
% Function:    random_spherical_data
% Description: Draws a line on a sphere.
%
% Inputs:
%
% numClusters     - Number of clusters.
%
% numPoints       - Number of points per cluster.

% numPoints       - Kappa (concentration parameter);
%                     higher means closer clusters.

% means1, means2  - Optional vector of means.
%
%
% Outputs
%
% dataMatrix      - An nxd dataMatrix containing the random points.
% meanMatrix      - A matrix containing the means of each of the clusters.
% memMatrix       - An array of labels for each randomly generated datapoint.
%
%
% Usage: Used in hierarchical clustering on the unit hypersphere.
%
% Authors(s):
%   Mark Moyou - markmmoyou@gmail.com
% Yixin Lin - yixin1996@gmail.com
%   Glizela Taino - glizelataino@gmail.com
%
% Date: Monday 6th June, 2016 (2:34pm)
%
% Affiliation: Florida Institute of Technology. Information
%              Characterization and Exploitation Laboratory.
%              http://research2.fit.edu/ice/
% --------------------------------------------------------------------------



function [dataMatrix, meanMatrix, memMatrix] = random_spherical_data(...
  numClusters, numPoints, kappa, means1, means2)

if nargin <= 3
  means1 = nan(numClusters); means2 = nan(numClusters);
  for i = 1:numClusters
    means1(i) = rand_angle();
    means2(i) = rand_angle();
  end
end

if length(means1) ~= numClusters || length(means2) ~= numClusters
  error(’Means vectors are not the right length!’);
end

dataMatrix = []; meanMatrix = []; memMatrix = [];
for i = 1:numClusters
    mean1 = means1(i); mean2 = means2(i);
    % Create random angles using the von Mises distributions
    angles = [circ_vmrnd(mean1, kappa, numPoints) circ_vmrnd(mean2, kappa, numPoints)];
    % Convert to Cartesian coordinates
    [x ,y, z] = sph2cart(angles(:,1), angles(:, 2), 1);
    [muX, muY, muZ] = sph2cart(mean1, mean2, 1);

    meanMatrix = [meanMatrix; muX, muY, muZ];
    dataMatrix = [dataMatrix; [x y z]];
    memMatrix = [memMatrix; (i + zeros(numPoints, 1))];
end

end
```