Contented-Based Image Retrieval with Manifold Technique
Why choose manifold data structure?
What is the basic idea of the manifold?

• Build a weighted network on the data. (Affinity Matrix)
• Assign a positive ranking score to each query and zeros to the remaining points
• All points spread their score to their nearby neighbors via the weighted network
Manifold Algorithm on CBIR

1. Sort the pairwise distance among points in ascending order. Repeat connecting the two points with an edge according the order until a connected graph is obtained.

2. Form the affinity matrix $W$ defined by $W_{ij} = \exp[-d^2(x_i, x_j / 2\sigma^2)]$ if there is an edge linking $x_i$ and $x_j$. Let $W_{ii} = 0$

3. Symmetrically normalize $W$ by $S = D^{-1/2} WD^{-1/2}$ in which $D$ is the diagonal matrix with $(i,i)$-element equal to the sum of the $i$-th row of $W$.

4. Iterate $f(t+1) = \alpha S f(t) + (1-\alpha) y$ until convergence, where alpha is a parameter in [0,1).

5. Let $f_i^*$ denote the limit of the sequence $\{f_i(t)\}$. Rank each point $x_i$ according its ranking scores $f_i^*$ (largest ranked 1st).
Some analysis of manifold

1. The sequence $\{f_i(t)\}$ converges to $f^* = (1 - \alpha S)^{-1} y$ where $y$ is the label vector.

   $y = [y_1, y_2, ..., y_n]$, in which $y_i = 1$ if $x$ is a query, and $y_i = 0$ otherwise.

2. Using L1 distance take place of L2 distance in the algorithm

   L1 distance is based on the Laplacian kernel

   $$k_L(x_i, x_j) = \prod_{l=1}^{m} \frac{1}{2\sigma_l} \exp(-\frac{|x_{il} - x_{jl}|}{\sigma_l})$$

where $x_{il}$ and $x_{jl}$ are the $l$-th dimension of $x_i$ and $x_j$ respectively; $m$ is the dimensionality of the feature space; and $\sigma_l$ is a positive parameter that reflects the scope of different dimensions.

   $$W_{ij} = k_L(x_i, x_j) = \prod_{l=1}^{m} \exp(-\frac{|x_{il} - x_{jl}|}{\sigma_l})$$
How to code the manifold algorithm

1. Built Affinity matrix,
   \[ W = \text{getAffinity}() \], according to step 2 in the algorithm
2. Normalize the affinity matrix,
   \[ S = \text{normAffinity}() \], according to step 3 in the algorithm
3. Continue to process the S
   \[ A = \text{inv}(1-a*S) \]; according to the analysis 1
4. Initialize the label vector,
   \[ y = \text{zeros}(1,\text{length(NumImages)}) \];
   \[ y(\text{query}) = 1 \];
5. Propagate the ranking score,
   \[ r = A * y \];
How to code the manifold algorithm

6. Return top K images with high ranking scores to be judged by users.

7. Update the label vector y according to the users’ judgment.
   
   \[ y(\text{positive feedback}) = 1; \]

8. Repeat step 5-7 till retrievals are satisfied.
One more variant of manifold algorithm

Same steps of 1-3

4. Initialize two label vectors, one for positive feedbacks, one for negative feedbacks.
   \[ y_{\text{Pos}} = \text{zeros}(\text{length}(\text{NumImages}),1) \]
   \[ y_{\text{Pos}}(\text{query}) = 1 \]
   \[ y_{\text{Neg}} = \text{zeros}(\text{length}(\text{NumImages}),1) \]

5. Propagate the ranking score.
   \[ r = A \times y_{\text{Pos}} - \text{weight} \times A \times y_{\text{Neg}} \]
One more variant of manifold algorithm

6. Return top K images with high ranking scores to be judged by users.

7. Update the label vector $y$ according to the users’ judgment.

   \[ y_{\text{Pos}}(\text{positive feedback}) = 1; \]
   \[ y_{\text{Neg}}(\text{negative feedback}) = -1; \]

8. Repeat step 5-7 till retrievals are satisfied.
Question?