Gene sorting project

1 Sorting on a two-dimensional lattice

problem statement

Let {x[i]} collect points in Rd. 2D sorting involves a task of placing each point on a node within a lattice. The placement is realized by assigning x to a cortical point y[n] on the lattice such that d[n] = min_m d[m], d[m] =llx-y[m]ll.

2D sorting is indeed a task of determining all y[n] on a lattice

2 Enumerate nodes on an MxM lattice

Two index enumeration specifies the joint position of the ith row and jth column on a lattice by (i,j). Let n =h(i,j)= M * (i -1)+ j. h induces one index enumeration of nodes on a lattice.

3 Neighboring relation

Each cortical point y[n] has it's neighboring cortical points on a lattice according to contexts of INA gene sorting 38-39. Let NB(n) collect neighboring indices of n on a lattice.

Problem a1: Let Y be a matrix with rows collecting all y[n]. Y has M*M rows and d columns . Draw a flow chart to determine the mean of the distance between neighboring cortical points for given Y.

Problem a2: write Matlab codes to implement the flow chart. Assign y[n] to vector z[n]=((i-1)/(M-1),(j-1)/(M-1)), where n=h(i,j). Apply Matlab codes to calculate the mean of the distance between neighboring cortical points and verify the answer.

Gene expressions

Yeast_822 collects 822 yeast gene expressions at seven time courses. Try to sort them on an MxM lattice, where M = round(√822) and cortical points belong R7.

Problem b1: Apply SOM toolbox to sort yeast822 on a lattice.

Problem b2: Calculate the mean of the distance between neighboring cortical points.

Function approximation

Let {y[n]} collect cortical points after sorting yeast822 on a lattice by SOM toolbox.

Problem c1: prepare paired data S(k)={(z[n], y[n,k])} for k=1,...,d

Problem c2: learn an RBF network subject to paired data in S(k) for k=1,...,d

Problem c3: Let fk denote the mapping derived by learning an RBF network subject to S(k) and e(k,t) denote the mean square error of fitting fk to paired data in S(t). Display e(k,t) for all k and t.
Problem c4: Let U collect e(k,t) for k <= t and L for k <= t. Check if elements in each row of U increase and those in each row of L decrease strictly.

Classification of SRBCT cancers
There are four types of cancers in the SRBCT database.

Problem d1: For each cancer type, randomly pick five patients and sort their gene expressions on a two-dimensional lattice.

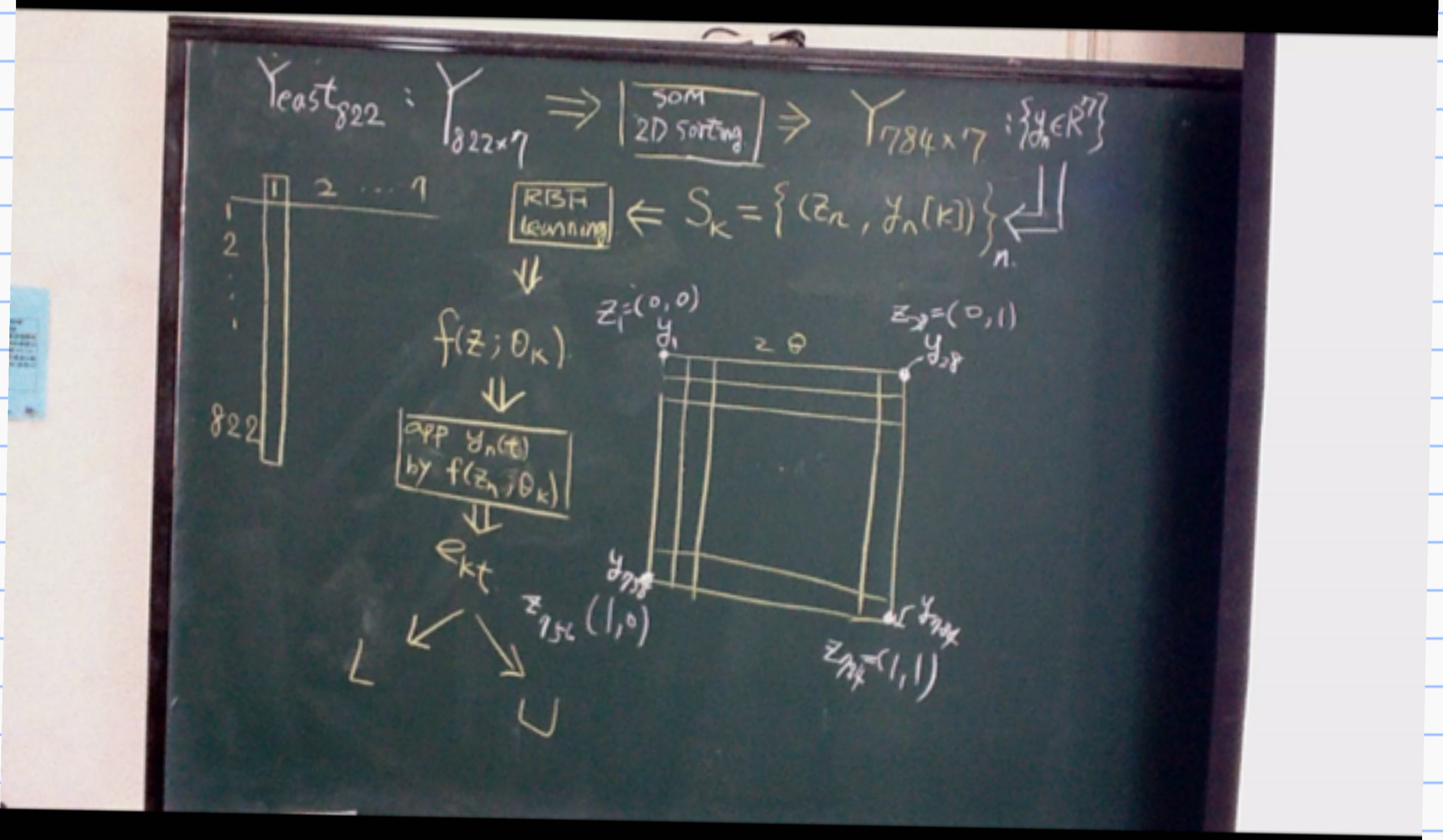
Problem d2: for each cancer c, display approximating functions fk(c), k=1...5, where c=1...4
Problem d3: According to gene sorting of each cancer type c on a lattice, gene expressions of a patient form five sets of paired data, denoted by S(c), c=1...4. Fitting fk(c) of different k to S(c) lead to a vector of 20 features.
Problem d4: how to classify four types of SRBCT cancers based on 20 features derived by gene sorting.

Classification of colon cancers
There are two types of patients in the colon database.

Problem e1: For each type, randomly pick five patients and sort their gene expressions on a two-dimensional lattice.

Problem e2: For each type c, display approximating functions fk(c), k=1...5, where c=1...2
Problem e3: According to gene sorting of each cancer type c on a lattice, gene expressions of a patient form two sets of paired data, denoted by S(c), c=1,2. Fitting fk(c) to S(c) lead to a vector of 10 features.
Problem d4: how to classify two types of patients based on features derived by gene sorting.

Problem f1: refer to INA gene clustering 60
Problem f2: refer to INA gene clustering 61



$$\text{Yeast}_{822} : Y_{822\times 7} \Rightarrow \boxed{\begin{array}{c}\text{SOM}\\ \text{2D sorting}\end{array}} \Rightarrow Y_{784\times 7} : \{y_n \in R^7\}$$

$$\boxed{\begin{array}{c}\text{RBF}\\ \text{learning}\end{array}} \Leftarrow S_K = \{(z_n, y_n[k])\}_n$$

$$\Downarrow$$

$$f(z;\theta_K)$$

$$\Downarrow$$

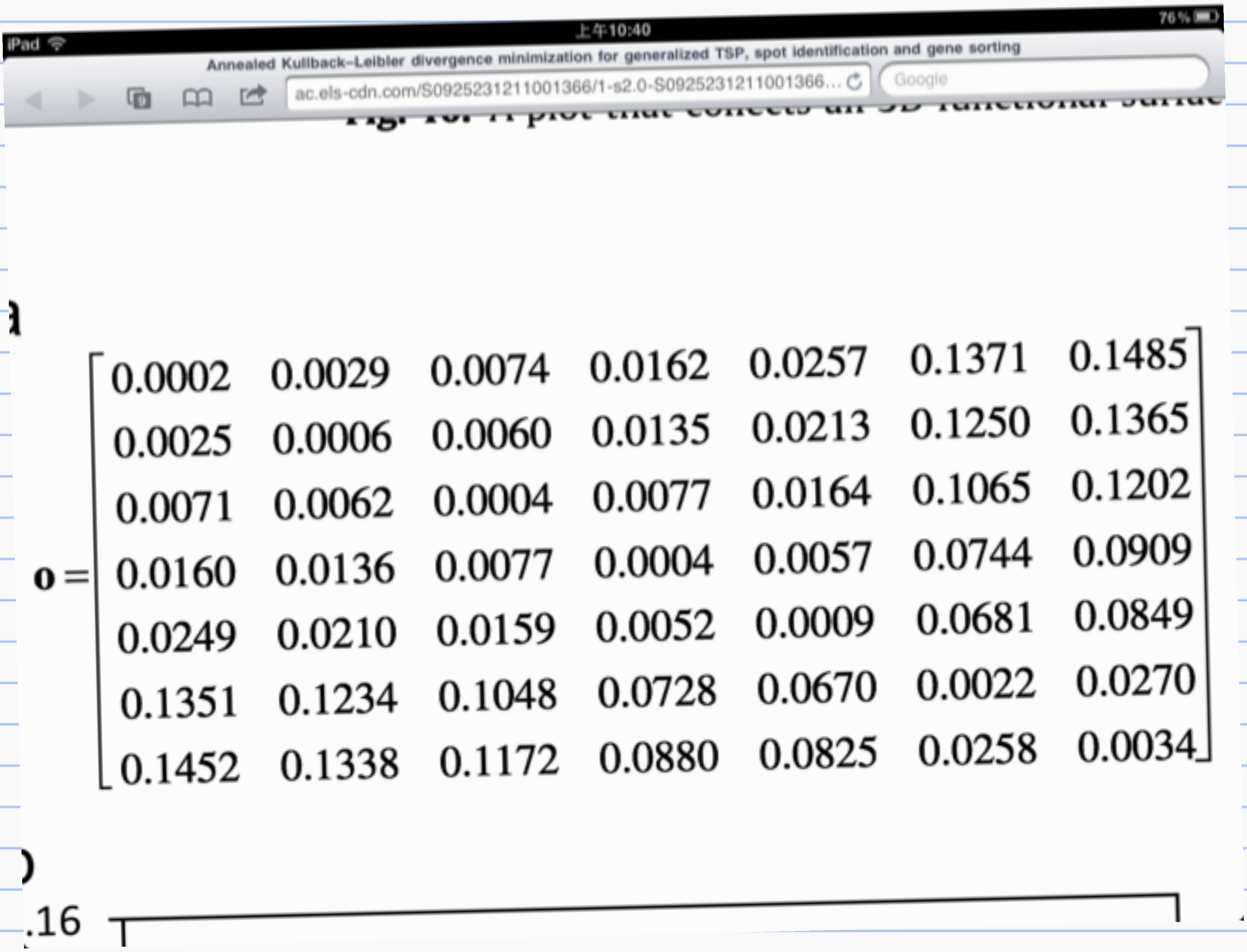
$$\boxed{\begin{array}{l}\text{app } y_n(t)\\ \text{by } f(z_n;\theta_K)\end{array}}$$

$$\Downarrow$$

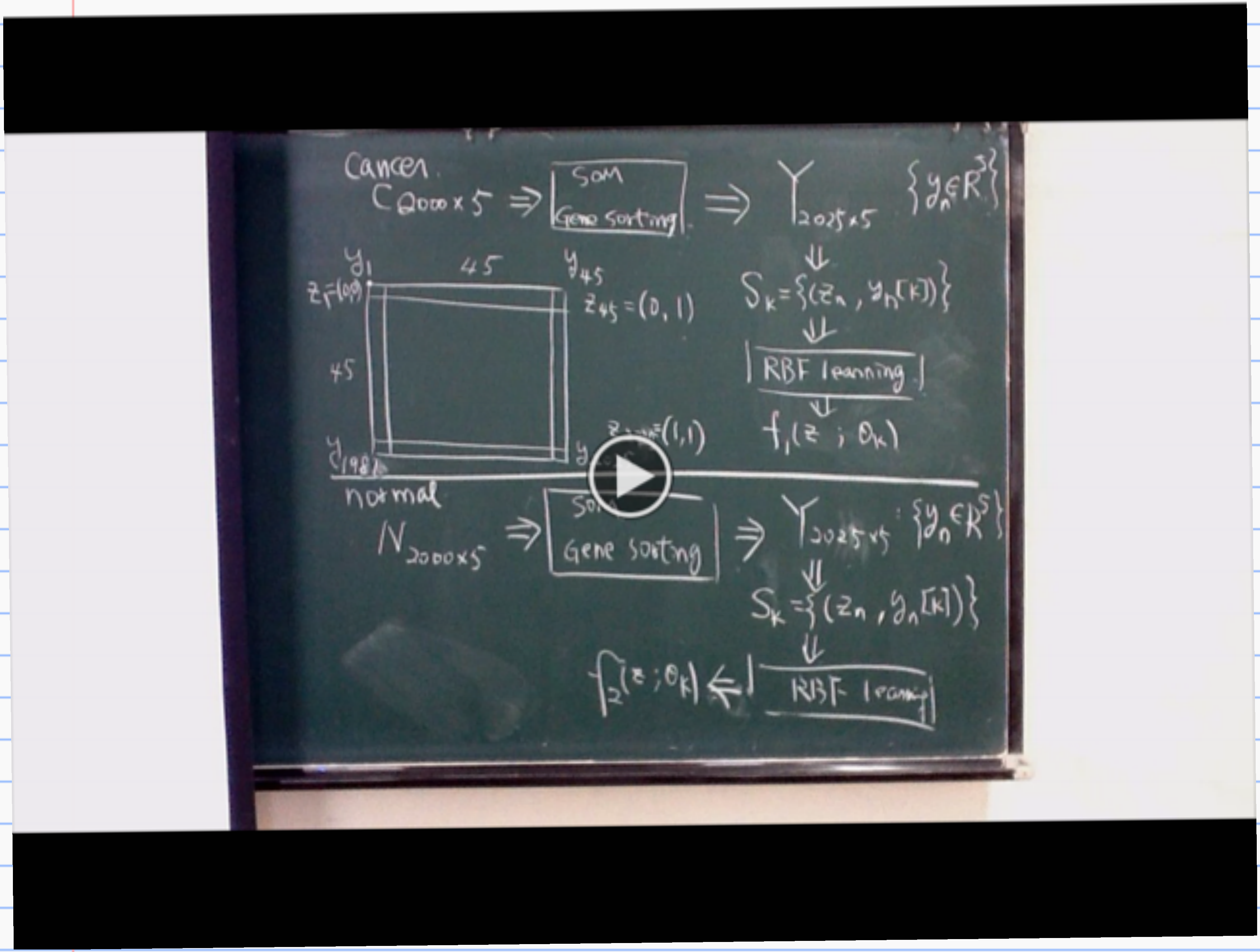
$$e_{kt}$$

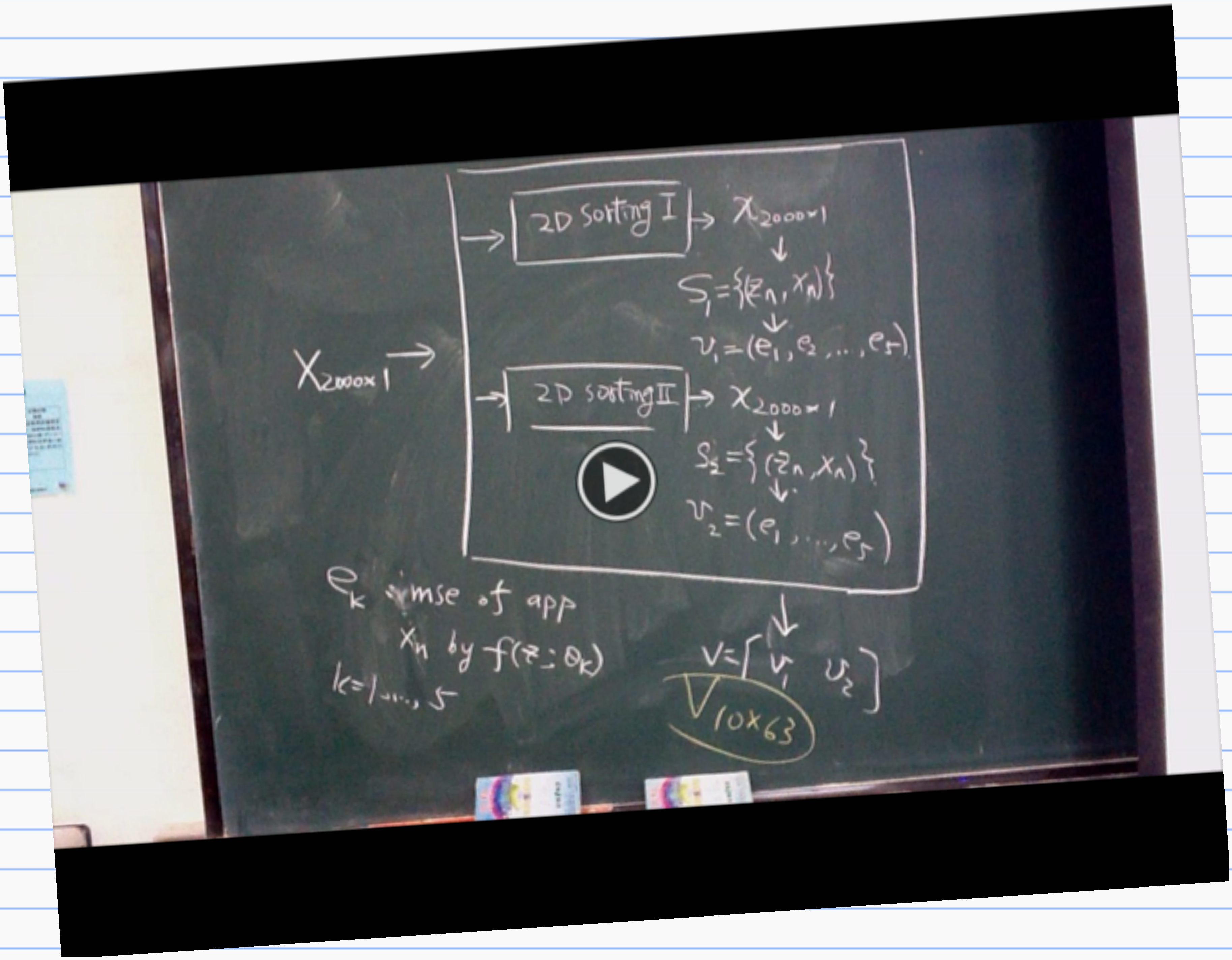
L

U


iPad
上午10:40
76 %
Annealed Kullback–Leibler divergence minimization for generalized TSP, spot identification and gene sorting
ac.els-cdn.com/S0925231211001366/1-s2.0-S0925231211001366...
Google
a
b
.16

$$\mathbf{o} = \begin{bmatrix} 0.0002 & 0.0029 & 0.0074 & 0.0162 & 0.0257 & 0.1371 & 0.1485 \\ 0.0025 & 0.0006 & 0.0060 & 0.0135 & 0.0213 & 0.1250 & 0.1365 \\ 0.0071 & 0.0062 & 0.0004 & 0.0077 & 0.0164 & 0.1065 & 0.1202 \\ 0.0160 & 0.0136 & 0.0077 & 0.0004 & 0.0057 & 0.0744 & 0.0909 \\ 0.0249 & 0.0210 & 0.0159 & 0.0052 & 0.0009 & 0.0681 & 0.0849 \\ 0.1351 & 0.1234 & 0.1048 & 0.0728 & 0.0670 & 0.0022 & 0.0270 \\ 0.1452 & 0.1338 & 0.1172 & 0.0880 & 0.0825 & 0.0258 & 0.0034 \end{bmatrix}$$


Cancer
$C_{2000\times5}$ ⇒ SOM Gene sorting ⇒ $Y_{2025\times5}$ $\{y_n \in R^5\}$
⇓
$S_k = \{(z_n, y_n[k])\}$
⇓
RBF learning
⇓
$f_1(z\ ; \theta_k)$
y_1
45
y_{45}
$z_1=(0,0)$
$z_{45}=(0,1)$
45
y_{1981}
$(1,1)$
normal
$N_{2000\times5}$ ⇒ SOM Gene sorting ⇒ $Y_{2025\times5}$ $\{y_n \in R^5\}$
⇓
$S_k = \{(z_n, y_n[k])\}$
⇓
$f_2(z\ ; \theta_k)$ ⇐ RBF learning



$X_{2000\times1}$ →

2D Sorting I → $x_{2000\times1}$ ↓ $S_1=\{(z_n,x_n)\}$ ↓ $v_1=(e_1,e_2,\dots,e_5)$

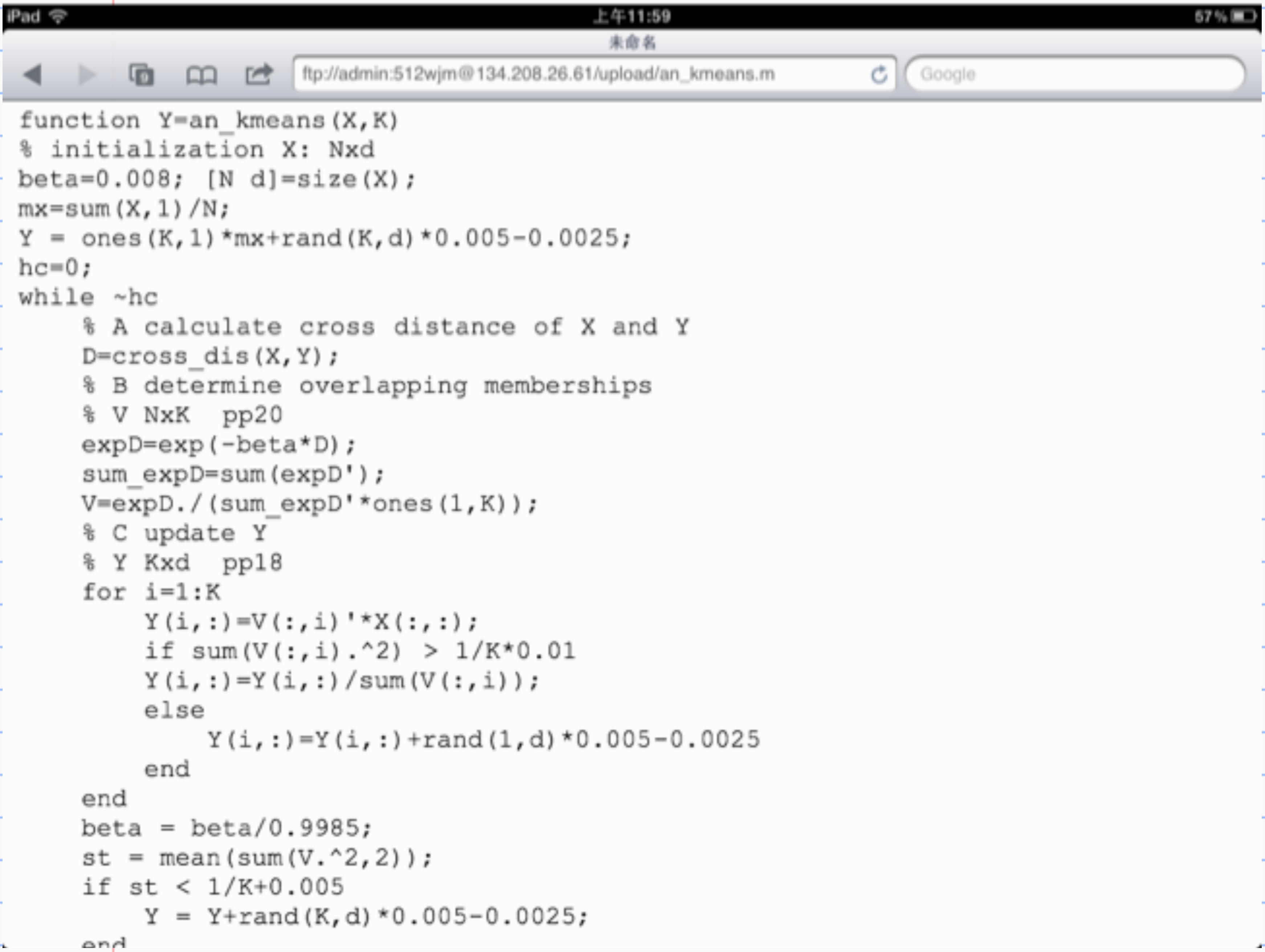
2D sorting II → $x_{2000\times1}$ ↓ $S_2=\{(z_n,x_n)\}$ ↓ $v_2=(e_1,\dots,e_5)$

↓ $V=[v_1\ \ v_2]$

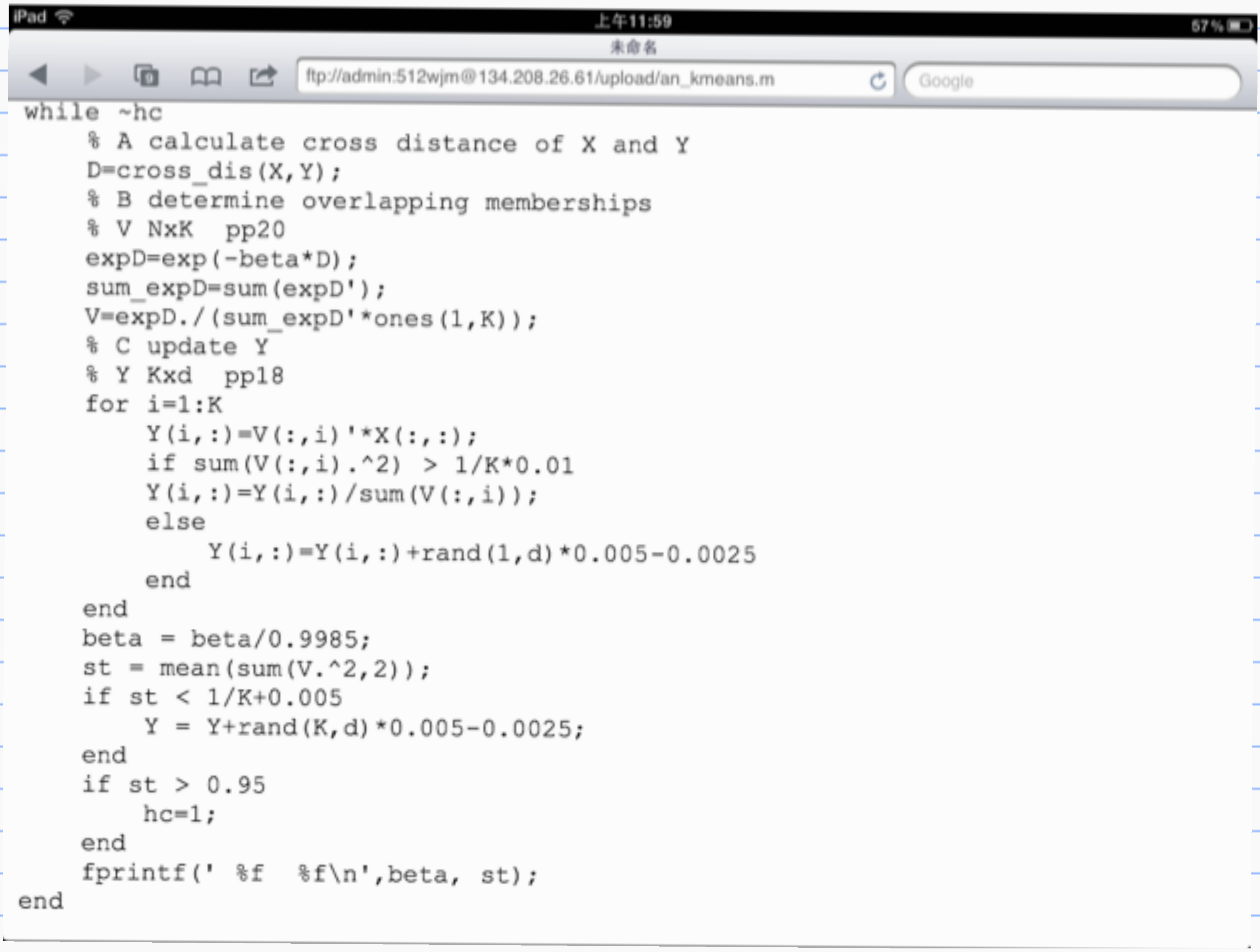
$V_{10\times63}$

e_k: rmse of app x_n by $f(z;\theta_k)$

$k=1,\dots,5$



```
function Y=an_kmeans(X,K)
% initialization X: Nxd
beta=0.008; [N d]=size(X);
mx=sum(X,1)/N;
Y = ones(K,1)*mx+rand(K,d)*0.005-0.0025;
hc=0;
while ~hc
    % A calculate cross distance of X and Y
    D=cross_dis(X,Y);
    % B determine overlapping memberships
    % V NxK  pp20
    expD=exp(-beta*D);
    sum_expD=sum(expD');
    V=expD./(sum_expD'*ones(1,K));
    % C update Y
    % Y Kxd  pp18
    for i=1:K
        Y(i,:)=V(:,i)'*X(:,:);
        if sum(V(:,i).^2) > 1/K*0.01
        Y(i,:)=Y(i,:)/sum(V(:,i));
        else
            Y(i,:)=Y(i,:)+rand(1,d)*0.005-0.0025
        end
    end
    beta = beta/0.9985;
    st = mean(sum(V.^2,2));
    if st < 1/K+0.005
        Y = Y+rand(K,d)*0.005-0.0025;
    end
```



ftp://admin:512wjm@134.208.26.61/upload/an_kmeans.m

```
while ~hc
    % A calculate cross distance of X and Y
    D=cross_dis(X,Y);
    % B determine overlapping memberships
    % V NxK  pp20
    expD=exp(-beta*D);
    sum_expD=sum(expD');
    V=expD./(sum_expD'*ones(1,K));
    % C update Y
    % Y Kxd  pp18
    for i=1:K
        Y(i,:)=V(:,i)'*X(:,:);
        if sum(V(:,i).^2) > 1/K*0.01
        Y(i,:)=Y(i,:)/sum(V(:,i));
        else
            Y(i,:)=Y(i,:)+rand(1,d)*0.005-0.0025
        end
    end
    beta = beta/0.9985;
    st = mean(sum(V.^2,2));
    if st < 1/K+0.005
        Y = Y+rand(K,d)*0.005-0.0025;
    end
    if st > 0.95
        hc=1;
    end
    fprintf(' %f  %f\n',beta, st);
end
```