

# Representative Selection:

See all by looking at a few

# Outline

- 1 Introduction and Motivation
- 2 Unsupervised Representative Selection
- 3 Supervised Representative Selection
- 4 Related issues on Social Networks

# Motivation: why we bother to select a subset of data?

**Popular Belief:** data are very redundant, and could be represented well by a relative small subset.

1

the selected instances are expected to be more interpretable

2

Reduce the memory cost of storing data and improve computation efficiency

3

Obtain better performance in classification with a discriminative subset selected

# Some examples: Unsupervised Representative Selection

Goal: find  
unk  
ana



entire  
for data

## Video Summarization



Digital Recognition



A high-probability set of size  $k = 10$  selected for the "safety" category

Recommendation System

# Low-Rankness based method: Rank Revealing QR factorization

## Idea:

the data come from a low-rank model and try to find a subset of instances that capture as much of the whole data set as possible in a projection sense.

**DEFINITION 1. (The CSSP)** Given a matrix  $A \in \mathbb{R}^{m \times n}$  and a positive integer  $k$ , pick  $k$  columns of  $A$  forming a matrix  $C \in \mathbb{R}^{m \times k}$  such that the residual

$$\|A - P_C A\|_{\xi}$$

is minimized over all possible  $\binom{n}{k}$  choices for the matrix  $C$ . Here,  $P_C = CC^+$  denotes the projection onto the  $k$ -dimensional space spanned by the columns of  $C$  and  $\xi = 2$  or  $F$  denotes the spectral norm or Frobenius norm.

**DEFINITION 2. (The RRQR factorization)** Given a matrix  $A \in \mathbb{R}^{m \times n}$  ( $m \geq n$ ) and an integer  $k$  ( $k \leq n$ ), assume partial QR factorizations of the form:

$$A\Pi = QR = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

it satisfies

$$\frac{\sigma_k(A)}{p_1(k, n)} \leq \sigma_{\min}(R_{11}) \leq \sigma_k(A)$$
$$\sigma_{k+1}(A) \leq \sigma_{\max}(R_{22}) \leq p_2(k, n)\sigma_{k+1}(A)$$

$C = A\Pi_k$  is the approximation of representative subset

# Dictionary learning based method

**Kmedoids**: assume that data are distributed around centers, called medoids, so each data point is represented by a medoid.

$$\min_{D,a} \sum_{i=1}^n \|x_i - Da_i\|_2^2, \quad \text{s.t. } \|a_i\|_0 = 1, \quad D \in X$$

It can be solved by two steps iteratively:

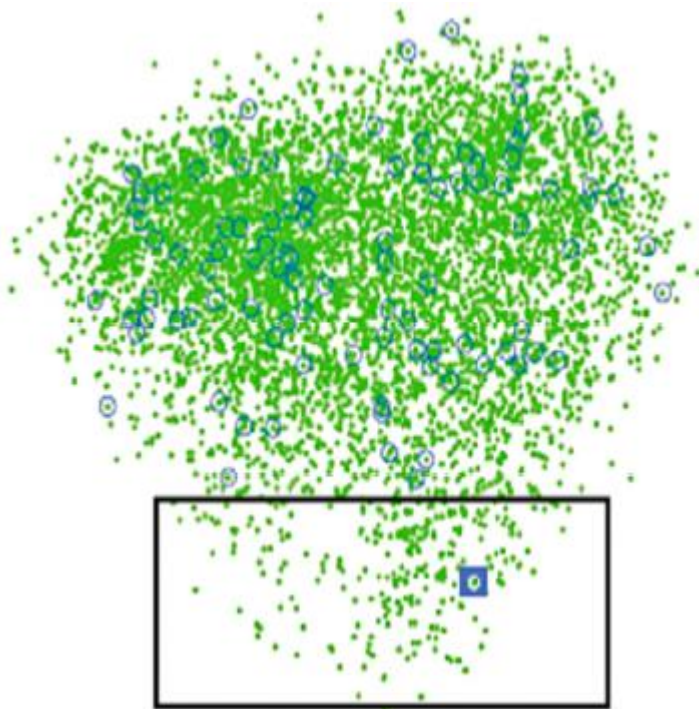
$$\min_a \sum_{i=1}^n \|x_i - Da_i\|_2^2, \quad \text{s.t. } \|a_i\|_0 = 1,$$

$$\min_D \sum_{i=1}^n \|x_i - Da_i\|_2^2, \quad \text{s.t. } D \in X,$$

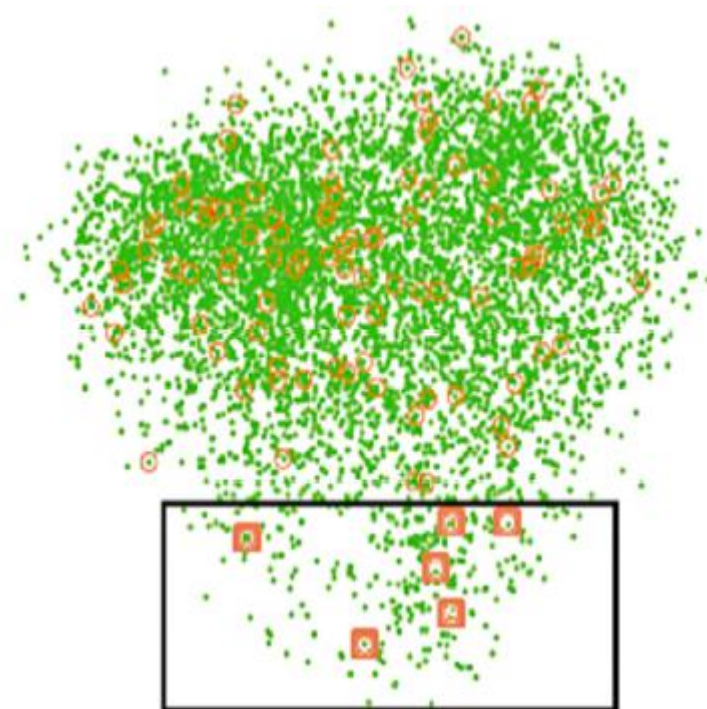
Kmedoids algorithm finds representative data mainly in high density region, so can't represent the global distribution of the dataset.

We can relax the only one representative constraint to allow that a data point can be a linear combination of multiple representatives.

$$\min_{D,A} \|X - DA\|_2^2 + \lambda \|A\|_1, \quad \text{s.t. } D \in X, A \geq 0$$



Kmedoids



Modified  
Kmedoids

## Sparse Modeling Representative Selection(SMRS):

By taking the original dataset as a dictionary, representatives are selected to approximately express all the data by linear combination with a row sparsity constraint.

$$\min \|C\|_{1,q} \quad \text{s.t.} \quad \|Y - YC\|_F \leq \varepsilon, \mathbf{1}^\top C = \mathbf{1}^\top$$

$\|C\|_{1,q} \triangleq \sum_{i=1}^N \|c^i\|_q$  encourages fewer non-zero rows, or the number of representatives

Representative is ranked by the norm of its corresponding row in coefficient matrix  $C$ , representative that has many nonzero elements with large values gets higher rank.



# Representative Selection with Structured Sparsity:

To encourage dissimilar samples to be selected, add regularizers for diversity and locality-sensitivity.

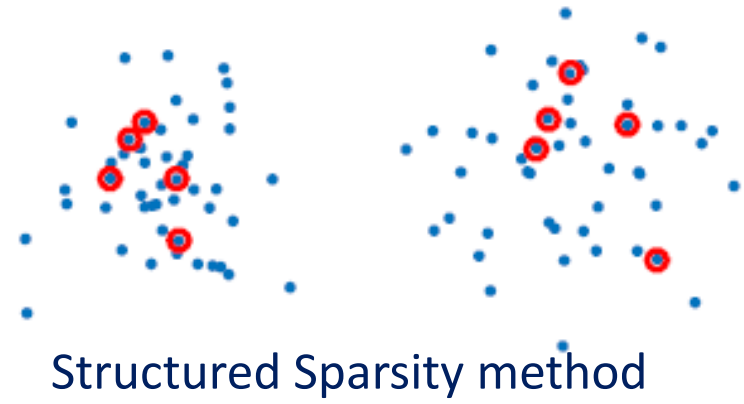
it prevents

$$\min_{\mathbf{V} \in \mathbb{R}^{n \times n}} \frac{1}{2} \|\mathbf{X} - \mathbf{XV}\|^2 + \lambda_1 \|\mathbf{V}\| + \lambda_2 \sum_{i=1}^n \|\mathbf{v}_i\| + \lambda_3 \sum_{i,j} \rho_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|_1.$$

representatives

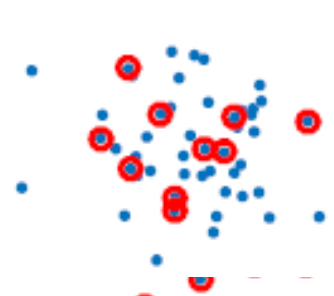
1 Row sparsity

2 Diversity regularization  
sample  $x_j$  is selected



selected when its dissimilar

$$\theta_{ij} = \begin{cases} dSim_{i,l} & \text{if } dSim_{i,l} < dSim_{j,l} \\ 0, & \text{otherwise} \end{cases}$$



$$\rho_{ij} = \begin{cases} \theta_{ij} & \text{if } \theta_{ij} > 0 \\ 0, & \text{otherwise} \end{cases}$$

3 locality-sensitivity regularization  
making sample coding natural and meaningful

generate similar codes,

$$\rho_{ij} = \begin{cases} Sim(\mathbf{x}_i, \mathbf{x}_j), & \text{if } Sim(\mathbf{x}_i, \mathbf{x}_j) \geq \max(Sim_{i,s}, Sim_{j,s}) \\ 0, & \text{otherwise} \end{cases}$$

## Decremental SMRS:

Perform iterative sample elimination, re-ranking, and re-weighting to remove the outliers and put different weights on data points according to their relevance scores.

$$\min_{\mathbf{B}} \left( \frac{1}{2} \|(\mathbf{Y} - \mathbf{Y B}) \mathbf{W}\|^2 + \lambda \|\mathbf{B}\|_{1,q} \right)$$

$$\text{Where, } \|(\mathbf{Y} - \mathbf{Y B}) \mathbf{W}\|^2 = \sum_{i=1}^{N_{\text{current}}} W_{ii}^2 \|\mathbf{y}_i - \mathbf{Y b}_i\|^2$$

In every iteration:

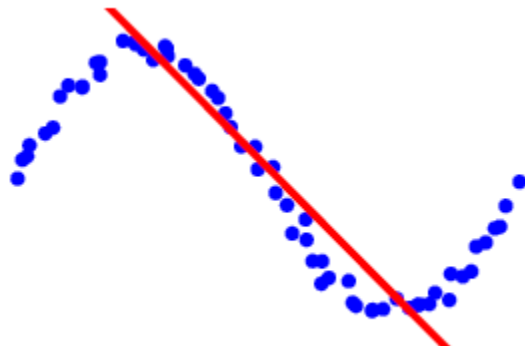
1. Calculate the coefficient matrix B and compute the L2 norms of the rows of B as relevance scores of data points.
2. Remove the data points with m lowest scores from Y.
3. Compute a new diagonal weight matrix W such that  $W(i,i)$ =relevance score of data point i.

# Dissimilarity based method

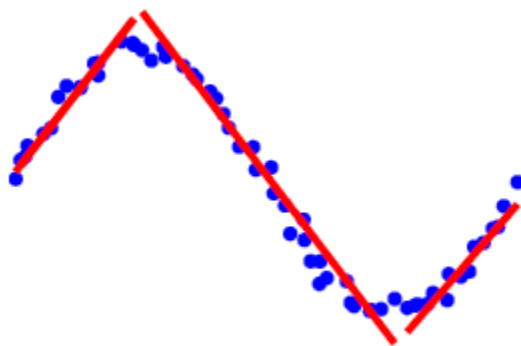
## Goal:

Assume we have a source set  $X = \{x_1, \dots, x_M\}$  and a target set  $Y = \{y_1, \dots, y_N\}$ , and we can get the pairwise dissimilarity  $d_{ij}$  indicating how well  $x_i$  represents  $y_j$ , i.e. the smaller the value of  $d_{ij}$ , the better  $x_i$  represents  $y_j$ . Arrange it into a matrix  $D \in R^{M \times N}$ . Our goal is to select a subset of  $X$  that efficiently represents  $Y$ .

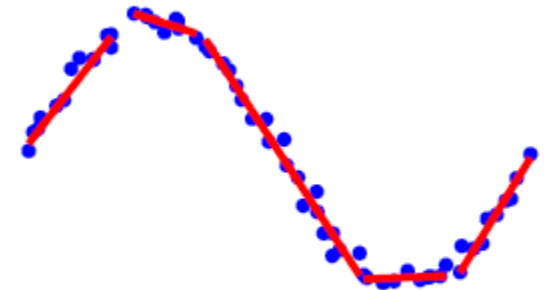
There,  $X$  and  $Y$  may not necessarily be the same type. For example,  $X$  can be a set of models and  $Y$  be a set of data points, in which case we select a few models that well represent the collection of data points.



(a)  $\lambda = \lambda_{\max, \infty}$



(b)  $\lambda = 0.1 \lambda_{\max, \infty}$



(c)  $\lambda = 0.01 \lambda_{\max, \infty}$

# Dissimilarity based method

Define  $Z \in R^{M \times N}$  as the indication matrix and if  $x_i$  is the representative of  $y_j$ ,  $z_{ij} = 1$ ; else  $z_{ij} = 0$ .  $z_i$  is the  $i$ -th row of  $Z$ .

Convex relaxation:

$$\min_{\{z_{ij}\}} \lambda \sum_{i=1}^M I(\|z_i\|_p) + \sum_{j=1}^N \sum_{i=1}^M d_{ij} z_{ij}$$

$$\text{s. t. } \sum_{i=1}^M z_{ij} = 1, \forall j; \quad z_{ij} \in \{0, 1\}, \forall i, j,$$



$$\min_{\{z_{ij}\}} \lambda \sum_{i=1}^M \|z_i\|_p + \sum_{j=1}^N \sum_{i=1}^M d_{ij} z_{ij}$$

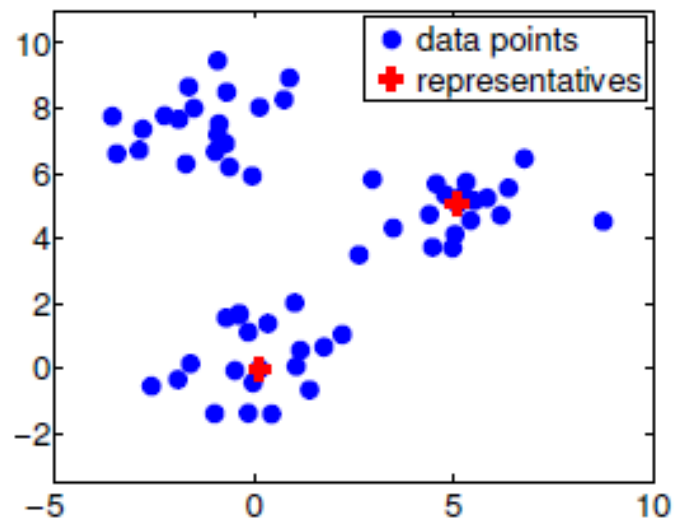
$$\text{s. t. } \sum_{i=1}^M z_{ij} = 1, \forall j; \quad z_{ij} \geq 0, \forall i, j,$$

To deal with outlier, we introduce a new variable  $e_j \in [0, 1]$  associated with each  $y_j$  indicating whether  $y_i$  is a outlier or not.

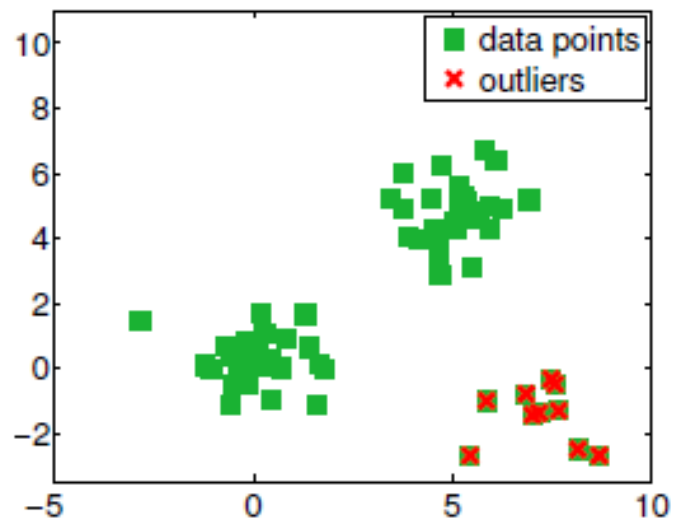
$$\min_{\{z_{ij}\}, \{e_j\}} \lambda \sum_{i=1}^M \|z_i\|_p + \sum_{j=1}^N \sum_{i=1}^M d_{ij} z_{ij} + \sum_{j=1}^N w_j e_j$$

$$\text{s. t. } \sum_{i=1}^M z_{ij} + e_j = 1, \forall j; \quad z_{ij} \geq 0, \forall i, j; \quad e_j \geq 0, \forall j.$$

# Dissimilarity based method



(a) Source set



(b) Target set

# Supervised Representative Selection

For data points with labels, we want to select representatives from all categories, which can be used for classification.

Here, we focus on the problems in the context of K-NN and SVM.

# Representative Selection for K-NN

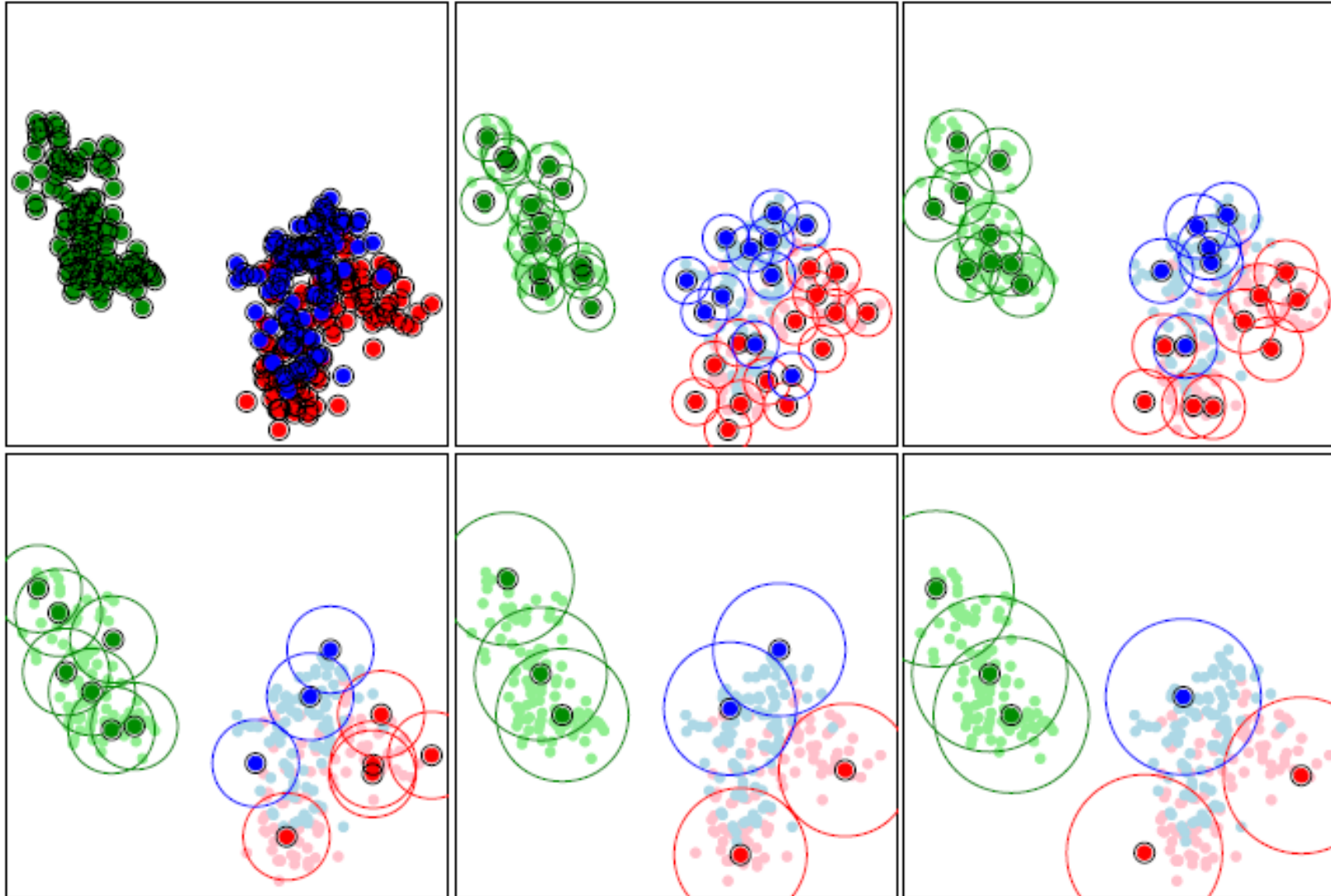
The introduction of a good selection method consists of two main steps:

- 1) consistency
- 2) fairness

We can think of this as a two-step process:

Given a target set  $\{1, \dots, L\}$ , a *ball* of radius  $r$  is the set of points  $x$  such that  $\|x - y_n\| \leq r$ .

- 1) covers the target set
- 2) covers the target set
- 3) is sparse



$\{y_n\} \in \mathcal{X}$   
 are the  $\epsilon$ -representatives  
 for class  $c$

It is actually a integer program problem:

$$\text{minimize } \sum_i \xi_i + \sum_i \eta_i + \lambda \sum_{j,l} \alpha_j^{(l)} \quad \text{s.t.}$$

$$\alpha_j^{(l)}, \xi_i, \eta_i$$

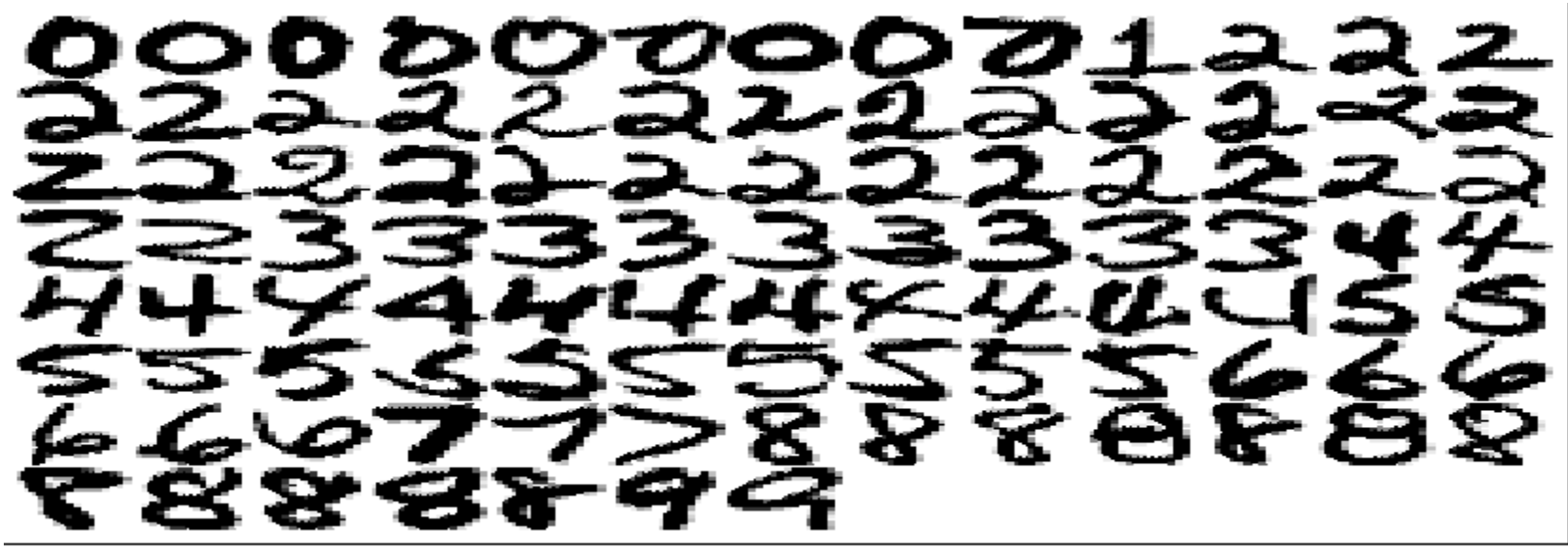
$$(a) \quad \sum_{j: \mathbf{x}_i \in B(\mathbf{x}_j)} \alpha_j^{(y_i)} \geq 1 - \xi_i \quad \forall \mathbf{x}_i \in \mathcal{X},$$

$$(b) \quad \sum_{\substack{j: \mathbf{x}_i \in B(\mathbf{x}_j) \\ l \neq y_i}} \alpha_j^{(l)} \leq 0 + \eta_i \quad \forall \mathbf{x}_i \in \mathcal{X},$$

$$\alpha_j^{(l)} \in \{0, 1\} \quad \forall j, l, \quad \xi_i, \eta_i \geq 0 \quad \forall i.$$

Where,  $\alpha_j^{(l)} \in [0,1]$  indicate whether we choose  $x_j$  to be a representative of class  $l$ .





100 representatives selected from USPS handwritten digits dataset

# Representative Selection for SVM

For SVM, the data points near the class boundary is more informative, and the non-boundary instances are considered redundant and do not affect the decision surface.

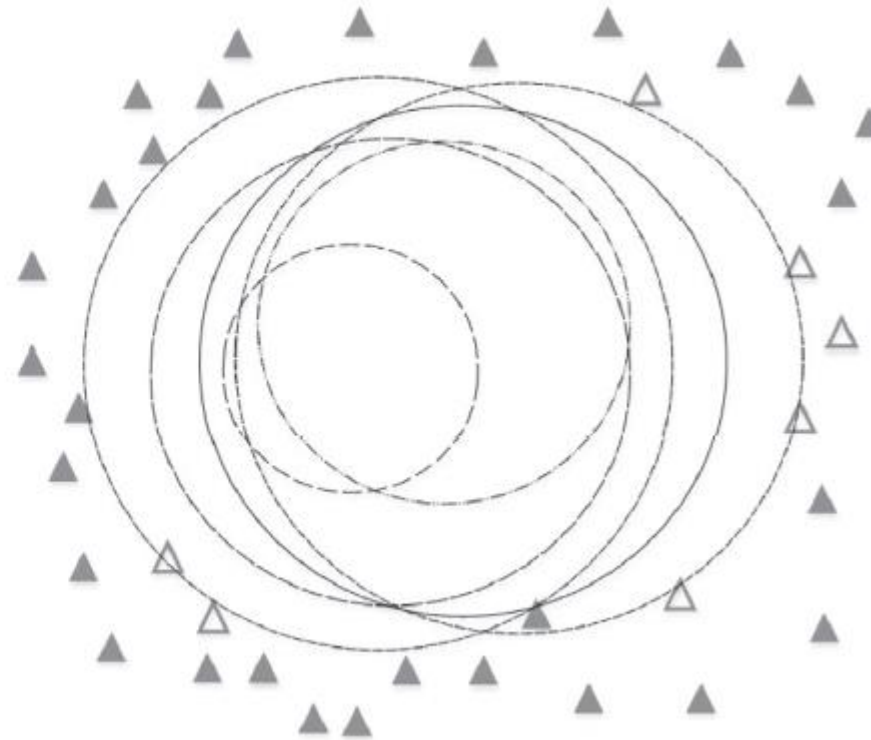
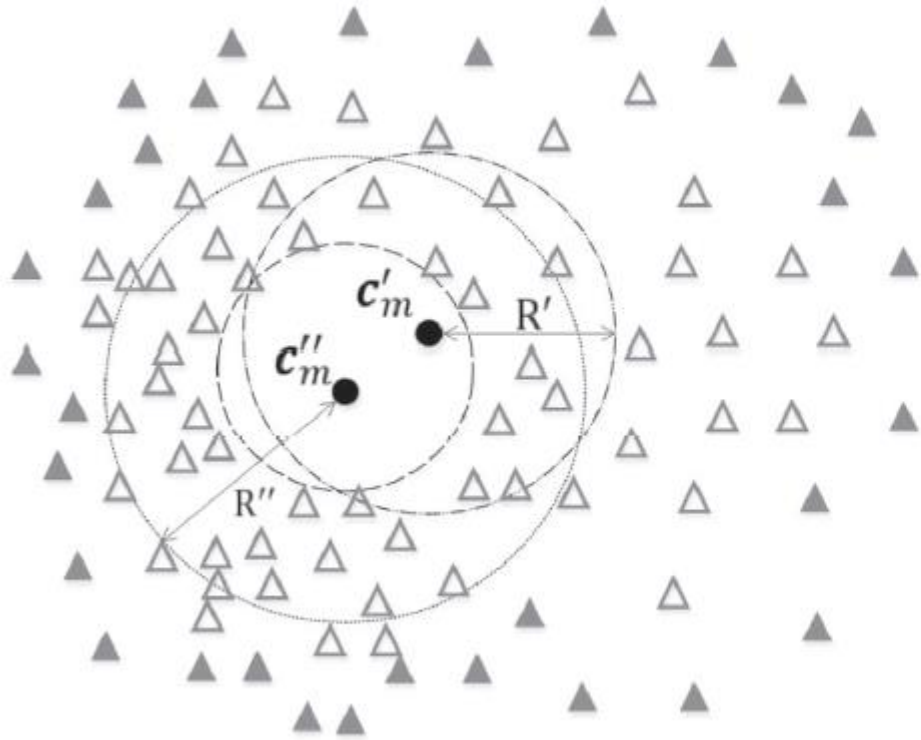
Goal:

Select representatives that preserve class boundary.

# Some simple approaches:

## ① Shell Extraction algorithm

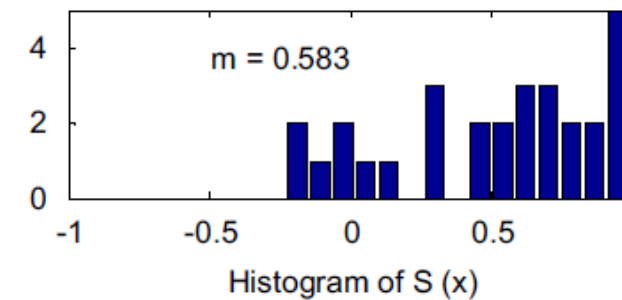
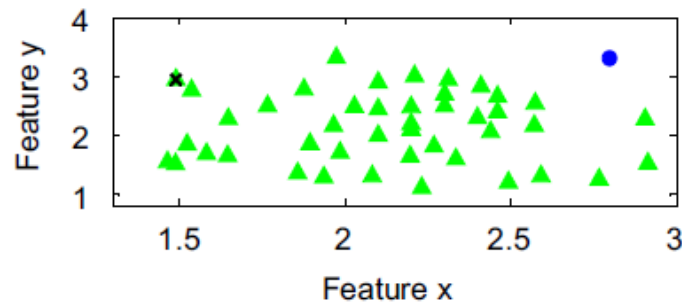
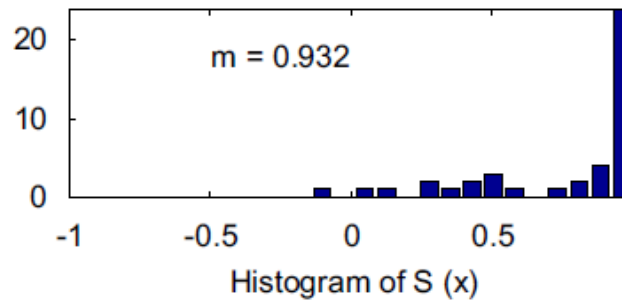
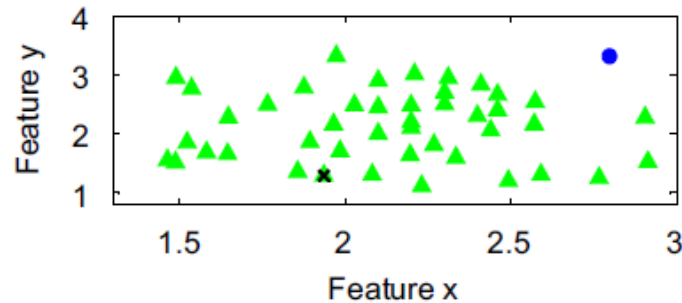
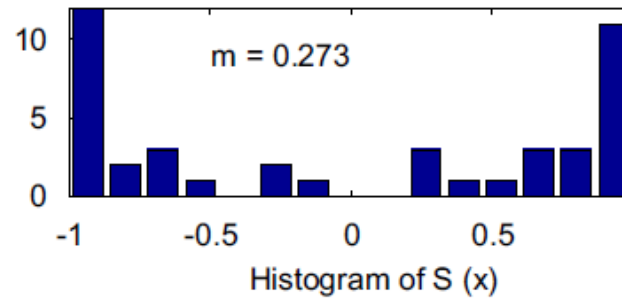
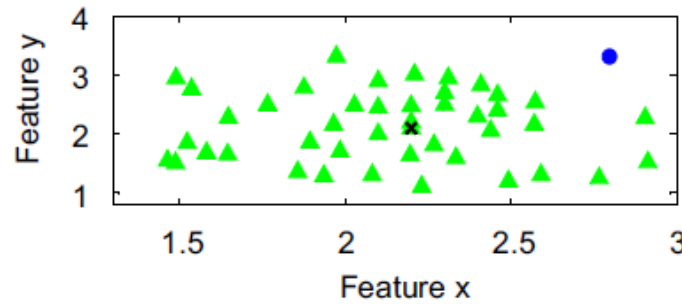
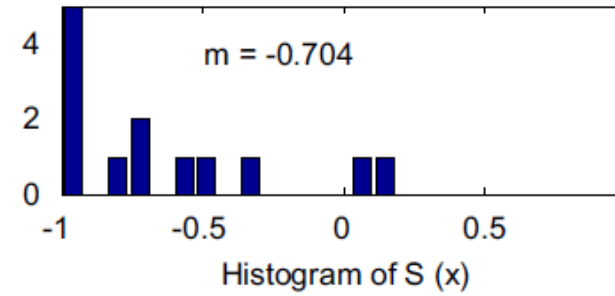
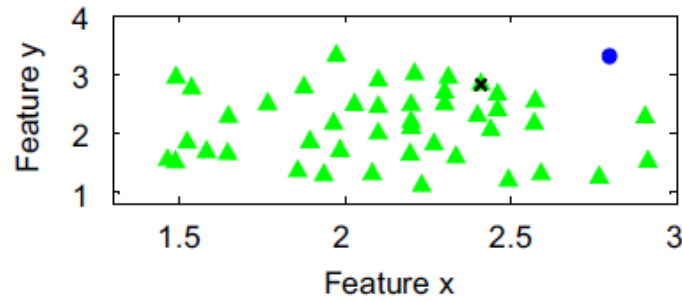
Assume that each class distribution is spherical



# Some simple

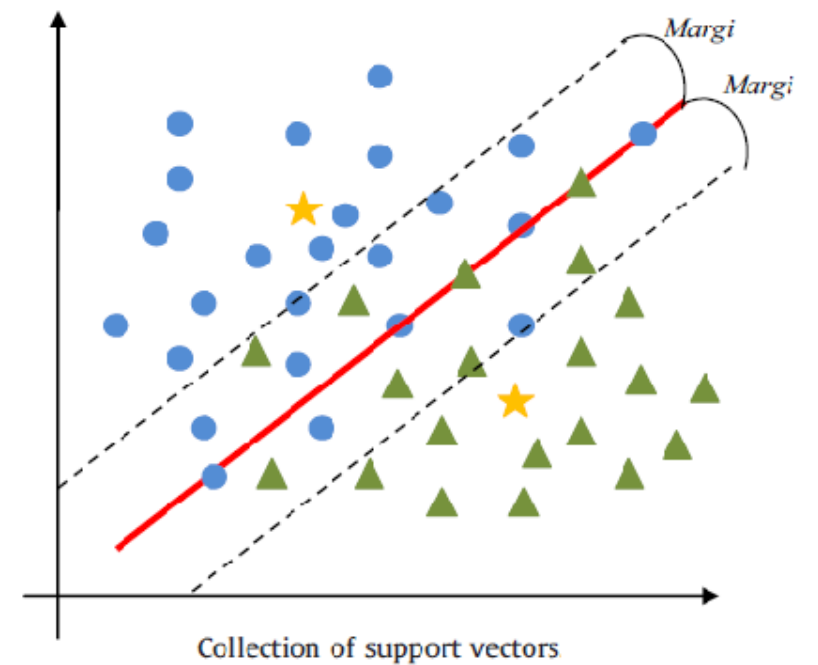
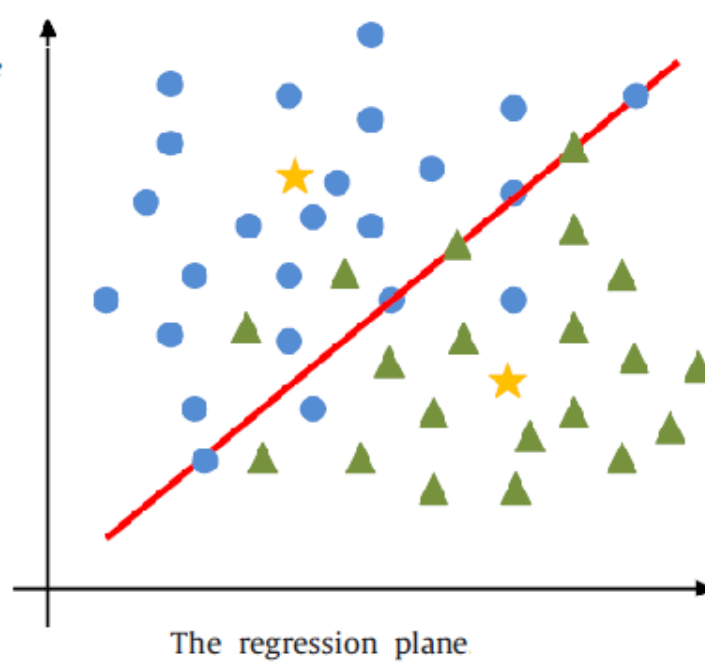
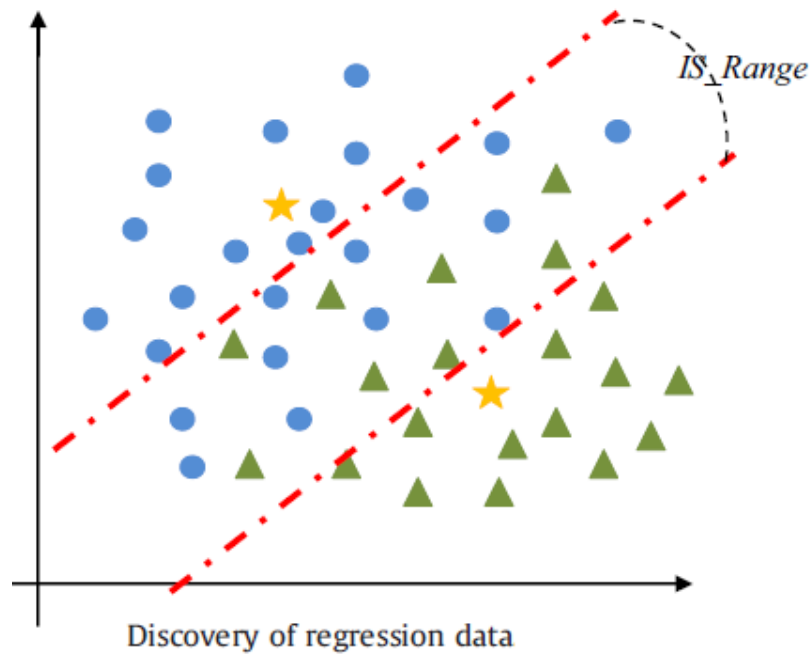
## 2 class boundary

Distinguish the



# Some simple approaches:

## ③ Support Vector Oriented Selection



# Related issues on Social Networks

- 1) How to select a subset of nodes from a social network which retains the underlying community structure?

The representative nodes should:

- 1) Contain nodes from all or most of the communities
- 2) if executing a community detection algorithm separately on both the sampled subgraph and the original network, we would like vertices grouped together in the subgraph to be also grouped together in the larger network.

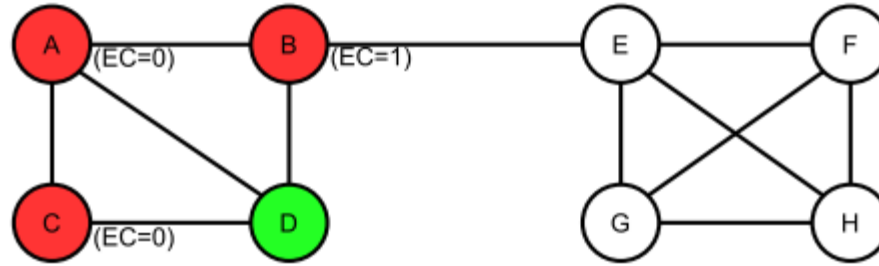
# Some approaches:

## ● Snowball Expansion sampling:

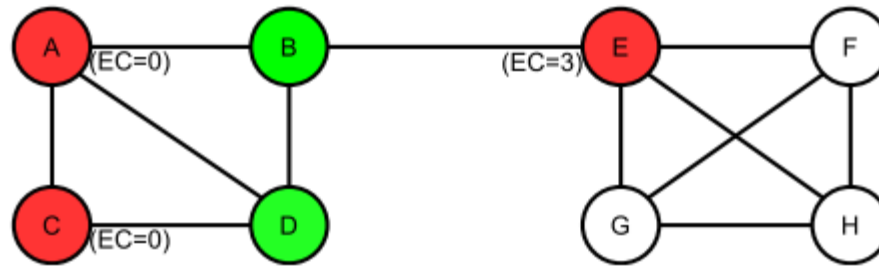
based on the notion that a small subset is representative of the maximum expansion.

$$\operatorname{argmax}_{S: |S|=k}$$

The term “snowball” is selected from current node  $v \in N(S)$  containing the maximum number of neighbors not in  $S$ .



(a) Step 1:  $D$  added to  $S$ .  $S = \{D\}$ ,  $N(S) = \{A, B, C\}$



(b) Step 2:  $B$  added to  $S$ .  $S = \{D, B\}$ ,  $N(S) = \{A, C, E\}$

nodes tend to be more representative of the maximum expansion subset with

neighbors of subgraph  $S$

nodes of the subset ( $S$ ) are selected from those that agree to which a neighbor ( $N(S) \cup S$ ).

# Some approaches:

- degree centrality based selection:

based on the notion that The nodes with high-degree centrality for each community are usually located at the center rather than the periphery and can better capture the community structure.

The algorithm can be divided into 3 steps:

- 1) Hub Selection: select the node  $v$  with highest degree centrality
- 2) Deactivation: deactivate the neighbors of  $v$ , and do not consider these nodes for selection
- 3) Reactivation: when there is no node being active, we reactivate all deactivated nodes.



# Related issues on Social Networks

- 2 How to find a set of users in a social network, such that by targeting this set, one maximizes the expected spread of influence in the network.

For example:

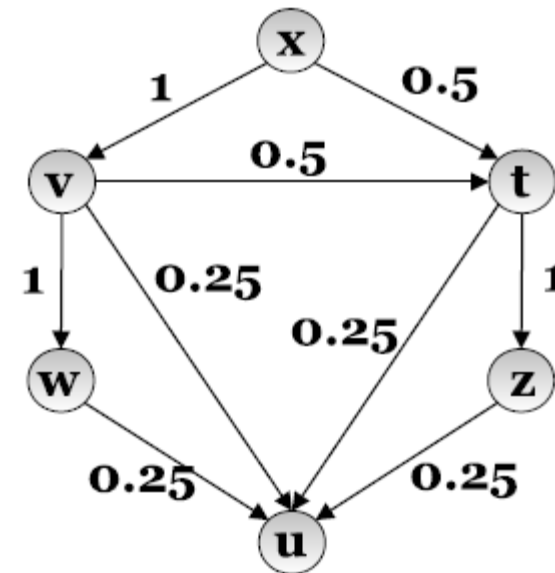
- 1) Viral marketing: by initially targeting a few influential members will trigger a cascade of adoption of the products.
- 2) Epidemic diffusion: by immunize the key nodes, we can prevent a large scale epidemic.

# influence maximization problem

## Goal:

Given a directed graph  $G = (V, E, p)$ , where nodes are users and edges are labeled with influence probabilities among users, the influence maximization problem asks for a *seed set* of users, that maximizes the *expected spread* of influence in the social network, under a given propagation model.

Here, we are mainly concerned about *Independent Cascade diffusion model*: when node  $v$  first becomes active in step  $t$ , it is given a single chance to active its inactive neighbor  $w$ , succeeding with a probability  $p_{v,w}$ .



# greedy hill-climbing algorithm

Given a subset  $S$ , the expected spread with a diffusion model  $m$  is defined as :

$$\sigma_m(S) = \sum_{X \in \mathbb{G}} Pr[X] \cdot \sigma_m^X(S)$$

Where  $\sigma_m^X(S)$  is the number of nodes reachable from  $S$  in the possible world  $X$ .

## Greedy algorithm:

start with the empty set, and repeatedly add an node  $x$  that gives the maximum marginal gain to the current set  $S$ :


$$x = \arg \max_{w \in V - S} (\sigma_m(S + w) - \sigma_m(S))$$

greedy algorithm can be used because  $\sigma_m$  is:

- 1) submodular:  $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$  for any  $v$  and  $S \subseteq T$ .
- 2) *monotone*:  $f(S \cup \{v\}) \geq f(S)$

# Data-based credit distribution model

Given user action log data  $L(\text{user}, \text{action}, \text{time})$ , directly predict the influence spread of node sets, without any need for learning edge probabilities or conducting MC simulations.

$$\sigma_m(S) = \sum_{X \in \mathcal{G}} Pr[X] \cdot \sigma_m^X(S), \quad \sigma_m^X(S) = \sum_{u \in V} path_X(S, u)$$

$$\sigma_m(S) = \sum_{u \in V} \sum_{X \in \mathcal{G}} Pr[X] path_X(S, u)$$
$$= \sum_{u \in V} E[path(S, u)] = \sum_{u \in V} Pr[path(S, u) = 1]$$

To estimate  $Pr[path(S, u) = 1]$ , we use the concept “Credit Distribution”.

$$\sigma_{cd}(S) = \sum_{u \in V} \kappa_{S, u}$$

Using the action log  $L(\text{user}, \text{action}, \text{time})$ , we define the propagation graph of action  $a$  as directed graph  $G(a) = (V(a), E(a))$ , with  $V(a) = \{v \in V \mid \exists t: (v, a, t) \in L\}$  and  $E(a) = \{(u, v) \in E \mid t(u, a) < t(v, a)\}$ .  $N_{in}(u, a) = \{v \mid (v, u) \in E(a)\}$ .

Credit Distribution: When a user  $u$  performs an action  $a$ , we give some **direct influence credit**  $\gamma_{v,u}(a)$  (i.e.  $\gamma_{v,u}(a) = 1/d_{in}(u, a)$ ) to its neighbor  $v$  in  $N_{in}(u, a)$ . Then user  $v$  in turn passes on the credit to its predecessors in  $G(a)$ . So we define the **total credit**  $\Gamma_{v,u}(a)$  given to a user  $v$  for influencing  $u$  on action  $a$  as:

$$\Gamma_{v,u}(a) = \sum_{w \in N_{in}(u, a)} \Gamma_{v,w}(a) \cdot \gamma_{w,u}(a) \quad \text{and} \quad \Gamma_{v,v}(a) = 1$$

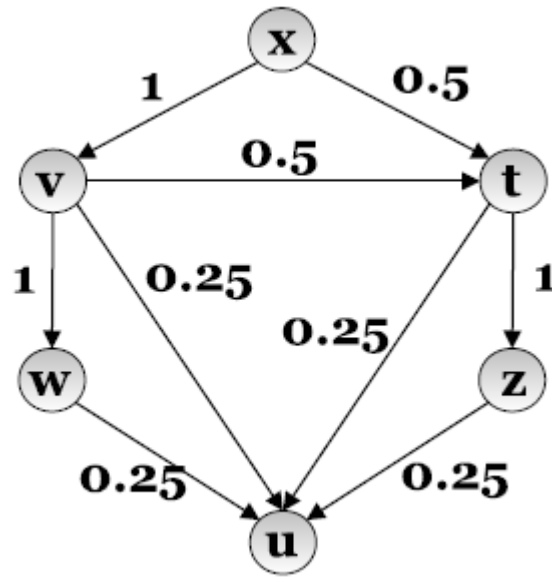
Then define the total credit given to a set of nodes  $S$   $\Gamma_{S,u}(a)$  as:

$$\Gamma_{S,u}(a) = \begin{cases} 1 & \text{if } v \in S; \\ \sum_{w \in N_{in}(u, a)} \Gamma_{S,w}(a) \cdot \gamma_{w,u}(a) & \text{otherwise} \end{cases}$$

Aggregates all actions,  $\kappa_{S,u} = \frac{1}{\mathcal{A}_u} \sum_{a \in \mathcal{A}} \Gamma_{S,u}(a)$  corresponds to  $\Pr[\text{path}(S, u) = 1]$ .

For example:

Given graph G(a):



$$\begin{aligned}\Gamma_{v,u} &= \Gamma_{v,v} \cdot \gamma_{v,u} + \Gamma_{v,t} \cdot \gamma_{t,u} + \Gamma_{v,w} \cdot \gamma_{w,u} + \Gamma_{v,z} \cdot \gamma_{z,u} \\ &= 1 \cdot 0.25 + 0.5 \cdot 0.25 + 1 \cdot 0.25 + 0.5 \cdot 0.25 = 0.75.\end{aligned}$$

Let  $S = \{v, z\}$ ,

$$\begin{aligned}\Gamma_{S,u} &= \Gamma_{S,w} \cdot \gamma_{w,u} + \Gamma_{S,v} \cdot \gamma_{v,u} + \Gamma_{S,t} \cdot \gamma_{t,u} + \Gamma_{S,z} \cdot \gamma_{z,u} \\ &= 1 \cdot 0.25 + 1 \cdot 0.25 + 0.5 \cdot 0.25 + 1 \cdot 0.25 = 0.875.\end{aligned}$$

THANK YOU