Semi-supervised learning: Clustering and Regression

What is semi-supervised learning?

Quotes from your HW answers

… is a mixture between supervised and unsupervised learning. Typically, it uses some labeled and a lot of unlabeled data as a training set. This kind of technique has proved to be efficient in practice, improving learning accuracy and shortening the setup time fo the training set, as all the data does not have to be manually labeled.

… is a combination of supervised and unsupervised learning where the trianing data is only partially labeled.

Semi-Supervised Learning

D. Zhou, O Bousquet, T. Navin Lan,

J. Weston, B. Schokopf

Presents: Tal Babaioff

Downloaded from http://www.cs.huji.ac.il/course/2004/learns/

Based on the following papers:
Zhou, Bousquet, Lal, Weston and Scholkopf,
Learning with Local and Global Consistency, NIPS*03
Zhu, Ghahramani and Lafferty,
Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions, 2003

Semi Supervised Learning

- Use small number of labeled data to label large amount of cheap unlabeled data.
- Basic idea: similar examples should be given the same classification.
- Typical example :

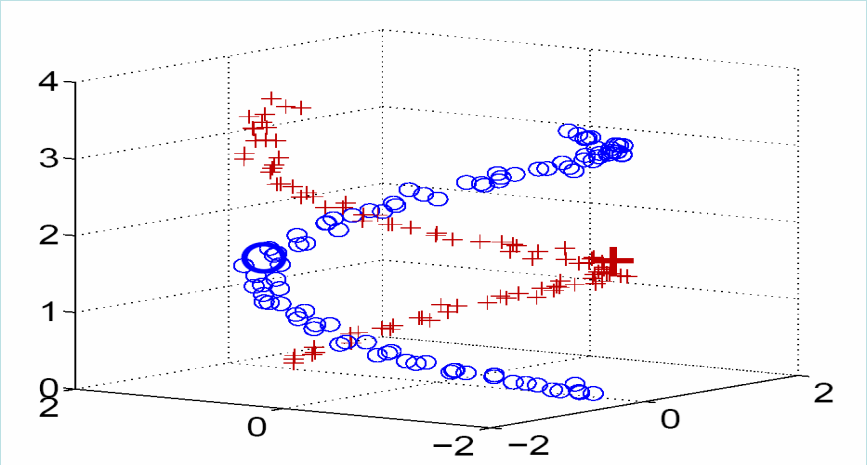
  web page classification: unlimited amount of cheap unlabeled data, while labeling is expensive.

The Cluster Assumption

- The basic assumption of most Semi-Supervised learning algorithms:

  Two points that are connected by a path going through high density regions should have the same label.

Example



Basic Approaches

- Using a weighted graph with weights representing point similarity:
- K nearest neighbors – the most naive approach.
- *Random walk on graph*:

  A particle starts from unlabeled node i and moves to node j with probability P_{ij}.

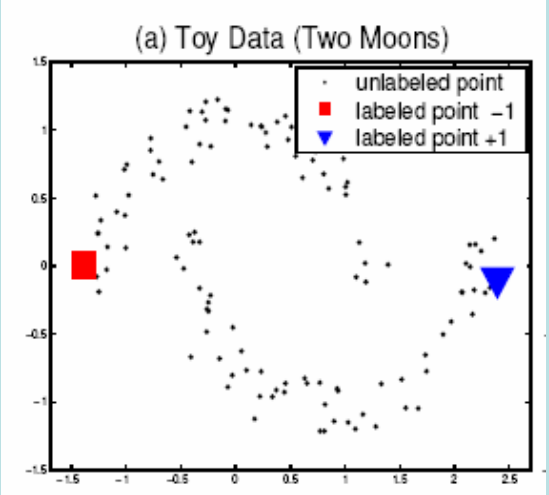
  The walk continues until the particle hits a labeled node.

  The classification of node i is based on the label with maximum probability to hit.

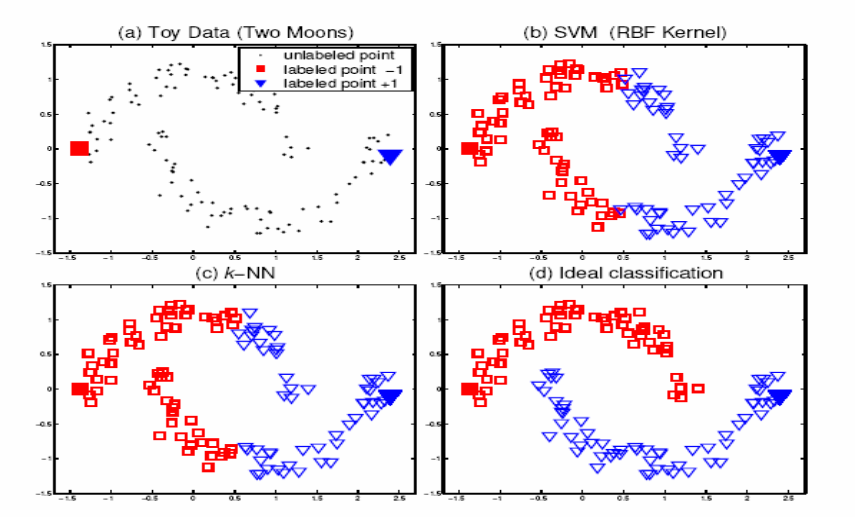
The Consistency Assumption

- Points in the same local high density region are more similar to each other (and thus likely to have the same label) than to points outside this region (*local consistency*).
- Points on the same global structure (a cluster or a manifold) are more similar to each other than to points outside of this structure *(global consistency).*

Consistency Assumption Example



Consistency Assumption Example



Formal Representation

- $X = \{x_1..x_l, x_{l+1}..x_n\} \in R^m$
- Label set $L = \{1,..c\}$
- The first i points have labeled $y_i \in \{1,..c\}$
- For points with $i>l$ y_i is unknown.
- The error is checked on the unlabeled examples only.

Basic Ideas For The Algorithm

- Define a similarity function that changes slowly locally in high density regions and changes globally on the manifold in which the data points lie.
- Define an activation network represented as a graph with weights determined by the similarity of each two points.

Basic Ideas For The Algorithm

- Use the labeled points as sources to pump the different classes labels via the graph, and use the new labeled points as additional source until a stable stage has been reached.
- The label of each unlabeled point is set to be the class of which it has received most information during the iteration process.

Algorithm : Data Structure

- Given a set of points: $X = \{x_1..x_l, x_{l+1}..x_n\}$
- The first l points have labeled $Y_i \in \{1,..c\}$ the rest are unlabeled.
- The classification will be presented on an [n x c] non negative matrix F.

  The classification of point x_i will be
  $y_i = \text{argmax}_{j<c} F_{ij}$.

  Let $Y \in F$ be a [n x c] matrix with elements
  Y_{ij} =1 if point i has a label y_i = j or 0 otherwise.

The Consistency Algorithm

1. Form the affinity matrix W defined by $W_{ij} = \exp(- \|x_i - x_j\|^2 / 2\sigma^2)$ if $i \neq j$ and $W_{ii} = 0$.
2. Compute the matrix S defined by
   $S = D^{-1/2} W D^{-1/2}$
   D is a diagonal matrix with its (i,i) element equal to the sum of the i-th row of W.
   The eigenvalues of S represents the spectral clusters of the data.

The Consistency Algorithm

3. Iterate $F(t+1) = \alpha SF(t) + (1-\alpha)Y$ until convergence. $\alpha \in (0, 1)$.
4. Let F* denote the limit of the sequence {F(t)}.
   Label the unlabeled point x_i by
   $y_i = \text{argmax}_{j \leq c} F^*_{ij}$

Consistency Algorithm – Convergence

- Show the algorithm convergence to:
  $F^* = (1-\alpha)(I - \alpha S)^{-1}Y$
- Without loss of generality, let $F(0) = Y$.
- $F(t+1) = \alpha S F(t) + (1-\alpha)Y$
- And therefore

  $F(t) = (\alpha S)^t Y + (1-\alpha)\Sigma^{t-1}_{i=0} (\alpha S)^i Y.$

Consistency Algorithm – Convergence

Show the algorithm convergence to:
$F^* = (1-\alpha)(I - \alpha S)^{-1}Y$

$F(t) = (\alpha S)^t Y + (1-\alpha)\Sigma^{t-1}_{i=0} (\alpha S)^i Y.$

Since:

$0 < \alpha < 1$

and the eigenvalues of S is in [-1, 1]:

$\lim_{t\to\infty} (\alpha S)^{t-1} = 0$

$\lim_{t\to\infty} \Sigma_{i=0}^{t-1} (\alpha S)^i = (I - \alpha S)^{-1}$

Hence: $F^* = \lim_{t\to\infty} F(t) = (1-\alpha)(I - \alpha S)^{-1}Y$

Regularization Framework

- Define a cost function for the iteration stage:

$$\mathcal{Q}(F) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \right\|^2 + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2 \right)$$

- The classifying function is $F^* = \arg\min_{F \in \mathcal{F}} \mathcal{Q}(F)$

$$\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \right\|^2$$

- *smoothness constraint*: a good classifying function should not change too much between nearby points.

Regularization Framework

$$\mathcal{Q}(F) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \right\|^2 + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2 \right)$$

$$\sum_{i=1}^{n} \|F_i - Y_i\|^2$$

- *fitting constraint*: a good classifying function should not change too much from the initial label assignment.
- μ >0 : Trade off between constraints

Regularization Framework

$$\mathcal{Q}(F) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \right\|^2 + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2 \right)$$

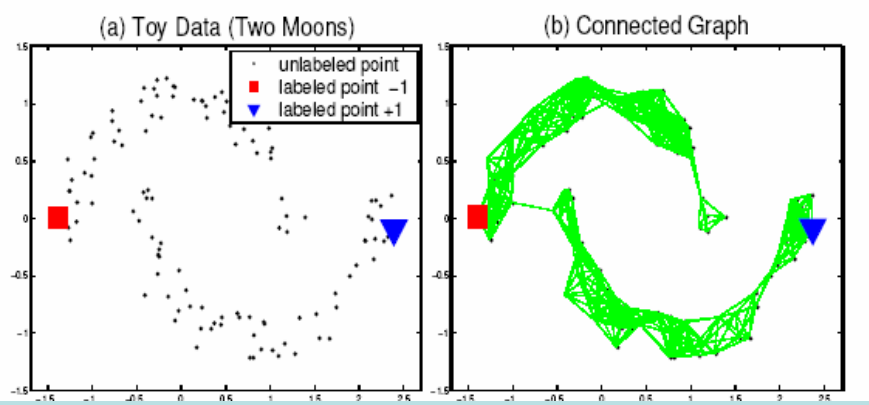
$$\left. \frac{\partial \mathcal{Q}}{\partial F} \right|_{F=F^*} = F^* - SF^* + \mu(F^* - Y) = 0,$$

$$F^* - \frac{1}{1+\mu} SF^* - \frac{\mu}{1+\mu} Y = 0.$$

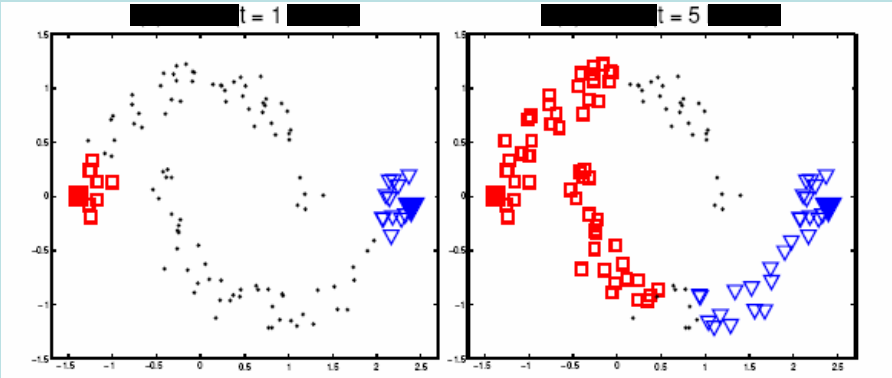
$$\alpha = \frac{1}{1+\mu}, \text{ and } \beta = \frac{\mu}{1+\mu} \qquad \alpha + \beta = 1$$

$$(I - \alpha S)F^* = \beta Y \qquad F^* = \beta(I - \alpha S)^{-1} Y.$$

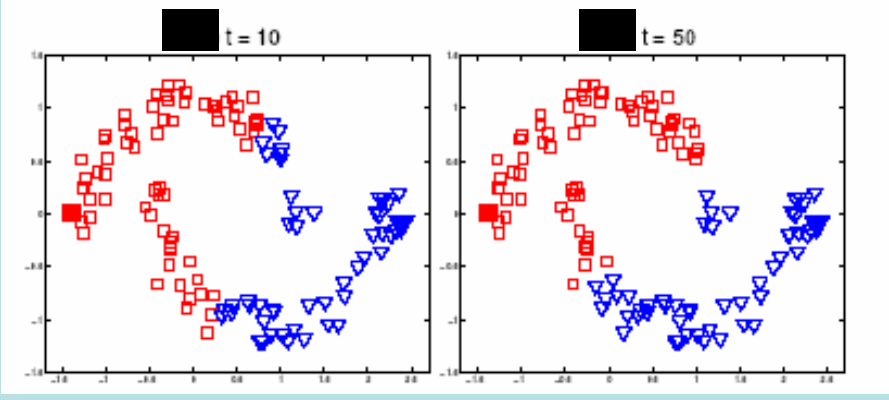
Results
Two Moon Toy Problem



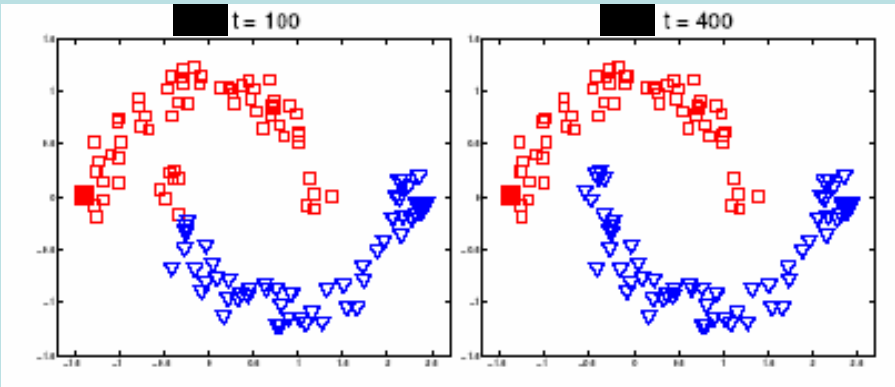
Results
Two Moon Toy Problem



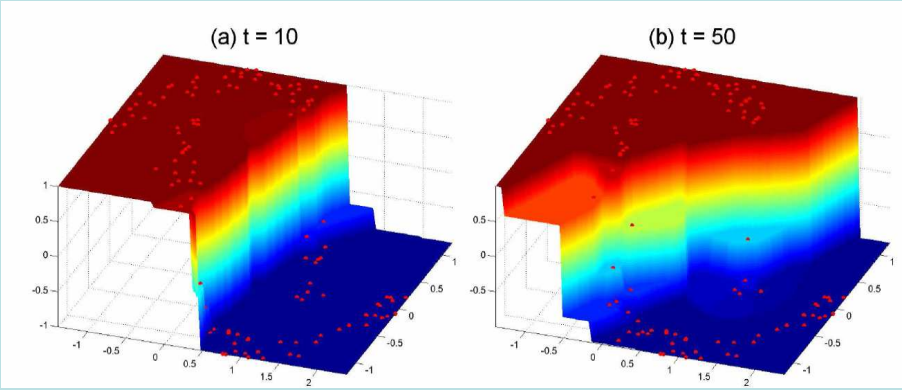
Results
Two Moon Toy Problem



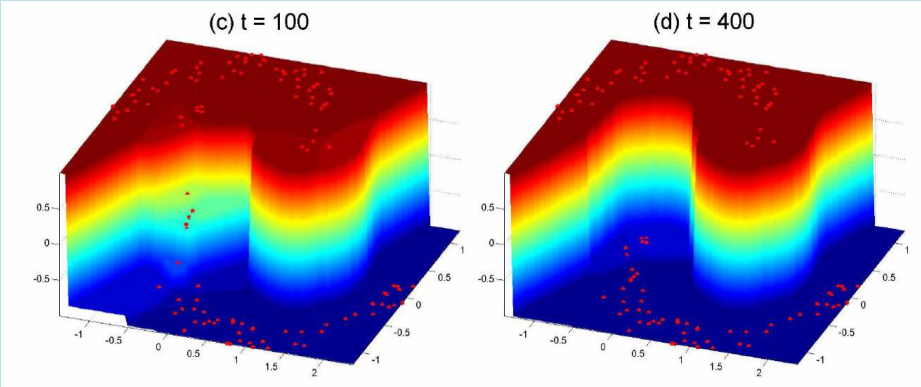
Results
Two Moon Toy Problem



Results
Two Moon Toy Problem



Results
Two Moon Toy Problem



Results: Digit Recognition

- Run the algorithm over USPS database with digits 1, 2, 3, 4.
- Class sizes are 1269, 929, 824, 852 (Total 3874).
- The test errors are averaged over 30 trials.
- The samples were chosen so that they contain at least one labeled point of each class.

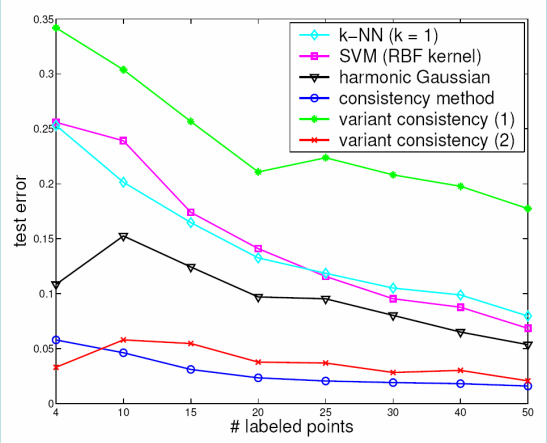
Results: Digit Recognition

Number of labeled data	4	8	12	16
K-nearest neighbor	25.45 (±1.30)	20.23 (±1.17)	15.52 (±0.87)	13.87 (±0.70)
Normal SVM	25.89 (±1.30)	16.89 (±1.14)	11.15 (±0.53)	10.10 (±0.52)
Cluster Kernel	16.24 (±1.40)	10.94 (±1.03)	7.52 (±0.29)	7.20 (±0.33)
LP (label propagation)	62.24 (±1.60)	59.12 (±2.07)	52.54 (±2.15)	46.78 (±2.07)
Consistency algorithm	8.02 (±1.51)	4.11 (±0.42)	2.76 (±0.15)	2.73 (±0.27)

Results: Digit Recognition

Number of labeled data	20	24	28	32
K-nearest neighbor	12.08 (±0.51)	10.93 (±0.50)	10.03 (±0.37	9.30 (±0.68)
Normal SVM	8.71 (±0.35)	7.91 (±0.29)	7.52 (±0.38)	7.37 (±0.31)
Cluster Kernel	6.55 (±0.22)	6.13 (±0.20)	6.04 (±0.25	5.97 (±0.24)
LP (label propagation)	44.88 (±2.33)	39.28 (±2.12)	35.18 (±1.55)	30.67 (±1.67)
Consistency algorithm	2.19 (±0.11)	2.04 (±0.10)	1.97 (±0.11)	1.79 (±0.27)

Results: Digit Recognition



Results averaged over 100 trials

Results: Text classification

- Use Mac & Windows subsets from 20 newsgroups data set.
- There are 961 and 985 examples in the two classes with 7511 dimensions.

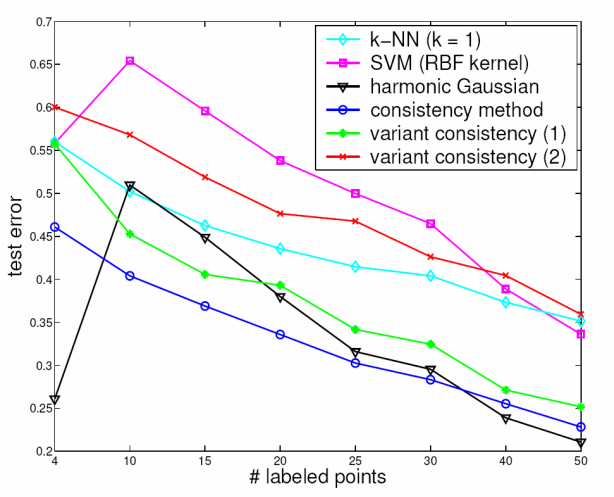
Results: Text Classification

Number of labeled data	2	4	8	16
K-nearest neighbor	37.40 (±0.0)	35.97 (±0.50)	32.80 (±0.62)	29.92 (±0.68)
Normal SVM	50.17 (±0.06)	35.66 (±1.12)	32.17 (±1.43)	28.92 (±1.39)
Cluster Kernel	45.28 (±1.47)	23.89 (±2.06)	15.13 (±0.99)	11.04 (±0.45)
LP (label propagation)	49.11 (±1.10)	49.08 (±0.11)	49.29 (±0.08)	49.04 (±0.07)
Consistency algorithm	24.05 (±1.96)	20.83 (±1.60)	15.56 (±0.79)	13.52 (±0.64)

Results: Text Classification 2

- Use the topic "rec" which contains *autos*, *motorcycles*, *baseball* and *hockey* subsets.
- Preprocessing:
  - Remove ending from all words (like ing, ed,…)
  - Don't pass words on the SMART list (the, of …)
  - Ignore the headers
  - Use only words that appear in 5 or more articles.
- Data base size: 3970 document vectors in a 8014-dimensional space

Results: Text Classification 2



References:

- Learning with Local and Global Consistency: Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, Bernhard Scholkopf
- http://www.kyb.mpg.de/publications/pdfs/pdf2333.pdf
- Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions:

  Xiaojin Zhu, Zoubin Ghahramani, John Lafferty
- http://www.hpl.hp.com/conferences/icml2003/papers/132.pdf

Semi-supervised Regression

Literature: *A Semi-Supervised Regression Model for Mixed Numerical and Categorical Variables*

Michael K. Ng, Elain Y. Chan, Meko M.C. So, Wai-Ki Ching

Regression

Regression analysis is a statistical technique that allows one to asses the relationship between a dependent variable and several independent variables.

Classical regression - we've seen this already

Logistic regression – find the probability of the occurrence of an event.
The dependent variable takes a value 1 with a probability of success θ, or value 0 with probability 1- θ. Use the following logistic function

$$\log\left(\frac{\theta}{1-\theta}\right) = \exp(\alpha + \beta_1 x_1 + \cdots + \beta_i x_i)$$

where x are the independent variables, and α and β are parameters to be estimated by the model

What about building regression models for ordinal data?

Numerical, Nominal and Ordinal data

R - Numerical attributes: data represented by real numbers
M - Categorical attribute: finite and discrete (aka ordinal)
N = number of records

Example Different types of data in customer purchasing records
Numerical data – Eg. Amount spent, time spent, age … etc
Categorical data – Eg. Gender, residential area … etc
Want to classify customers into two groups: Loyal or Contingent
This is unsupervised learning (clustering)

Instead, authors propose a novel model called **semi-supervised regression**.
Idea is that the algorithm is activated by a limited amount of supervisions.
These supervisions are in the form of constraints established by the labeled data

Notation

$$Z = (Z_1, Z_2, \ldots, Z_N)^T$$ *N x R* matrix for numerical attributes

$$C = (C_1, C_2, \ldots, C_N)^T$$ *N x M* matrix for categorical attributes

$$X = (Z, C) = (x_{nj}), \text{ where } n = 1, 2, \ldots, N; j = 1, 2, \ldots, R + M,$$

N x (R + M) matrix for all the records:
first *R* columns are numerical attributes, followed by *M* categorical attributes

The Regression Model

Regression coefficients

$$B_k = (\beta_0^k, \beta_1^k, \ldots, \beta_L^k)^T$$

Cluster k

L=R-1

Independent variables

$$Z = (1, Z_1, Z_2, \ldots, Z_L)^T.$$

$$\sum_{n=1}^{N} \left(Y^{(n)} - B_k^T Z^{(n)}\right)^2,$$

Prediction error for numerical data

where

$Y^{(n)}$ and $Z^{(n)} = [1, Z_1^{(n)}, Z_2^{(n)}, \ldots, Z_L^{(n)}]$

refer to the attributes of the nth record.

Clustering Algorithm

Want to **partition the data** in K clusters

K-modes algorithm: This uses a Kmeans paradigm to cluster categorical data

Denote the K-modes (centers) by V, a *KxM* matrix:

$$V = (V_1, V_2, \ldots, V_K)^T$$

And let W=[$w_{n,k}$] be an *NxK* matrix representing the partitioning of all records into K clusters.

Goal: find W and V that minimize the sum of the distances between V and its members …

Clustering Optimization Problem

min $$F(W,V) = \sum_{k=1}^{K}\sum_{n=1}^{N} w_{n,k} d(V_k, C_n),$$

subject to $$w_{n,k} \in \{0,1\} \quad , 1 \le k \le K, \quad 1 \le n \le N,$$
$$\sum_{l=1}^{K} w_{n,k} = 1, \qquad 1 \le n \le N.$$

where $$d(X_i, X_l) = \sum_{r=R+1}^{M+R} \delta(x_{i,r}, x_{l,r}) \qquad \delta(x_{i,r}, x_{l,r}) = \begin{cases} 0, & x_{i,r} = x_{l,r}, \\ 1, & x_{i,r} \neq x_{l,r}. \end{cases}$$

Clustering Optimization Problem

min $$F(W,V) = \sum_{k=1}^{K}\sum_{n=1}^{N} w_{n,k} d(V_k, C_n),$$

w represents cluster membership

subject to $$w_{n,k} \in \{0,1\} \quad , 1 \le k \le K, \quad 1 \le n \le N,$$
$$\sum_{l=1}^{K} w_{n,k} = 1, \qquad 1 \le n \le N.$$

where $$d(X_i, X_l) = \sum_{r=R+1}^{M+R} \delta(x_{i,r}, x_{l,r}) \qquad \delta(x_{i,r}, x_{l,r}) = \begin{cases} 0, & x_{i,r} = x_{l,r}, \\ 1, & x_{i,r} \neq x_{l,r}. \end{cases}$$

Compare only Categorical data

Same category or not?

Semi-Supervised Regression Model

SSRM integrates clustering and regression.

Best fit parameters are estimated by an iterative algorithm:
-Minimize the least square errors for regression
-Minimize the K-modes dissimilarity measures for categorical attributes.

$$\text{Min} \quad F(W,B,V,\lambda,\gamma) = \sum_{k=1}^{K}\sum_{n=1}^{N} w_{n,k}\left\{\lambda_{n,k}^{\eta}\{\left[Y^{(n)} - B_k^T Z^{(n)}\right]^2 + \gamma_{n,k}^{\eta} d(V_k, C_n)\right\}$$

subject to (2), (3) and

$$\begin{cases} \lambda_{n,k}, \gamma_{n,k} \geq 0, & 1 \geq k \geq K, \quad 1 \leq n \leq N, \\ \lambda_{n,k} + \gamma_{n,k} = 1, & 1 \leq k \leq K. \end{cases}$$

Semi-Supervised Regression Model

SSRM integrates clustering and regression.

Best fit parameters are estimated by an iterative algorithm:
-Minimize the least square errors for regression
-Minimize the K-modes dissimilarity measures for categorical attributes.

Cluster members

Regression

Clustering

$$\text{Min} \quad F(W,B,V,\lambda,\gamma) = \sum_{k=1}^{K}\sum_{n=1}^{N} w_{n,k}\left\{\lambda_{n,k}^{\eta}\{\left[Y^{(n)} - B_k^T Z^{(n)}\right]^2 + \gamma_{n,k}^{\eta} d(V_k, C_n)\right\}$$

subject to (2), (3) and

$$\begin{cases} \lambda_{n,k}, \gamma_{n,k} \geq 0, & 1 \geq k \geq K, \quad 1 \leq n \leq N, \\ \lambda_{n,k} + \gamma_{n,k} = 1, & 1 \leq k \leq K. \end{cases}$$

Weights for regression and clustering

P. Agius – L7semiSVR, Spring 2008

SSRM algorithm

The SSRM Algorithm.

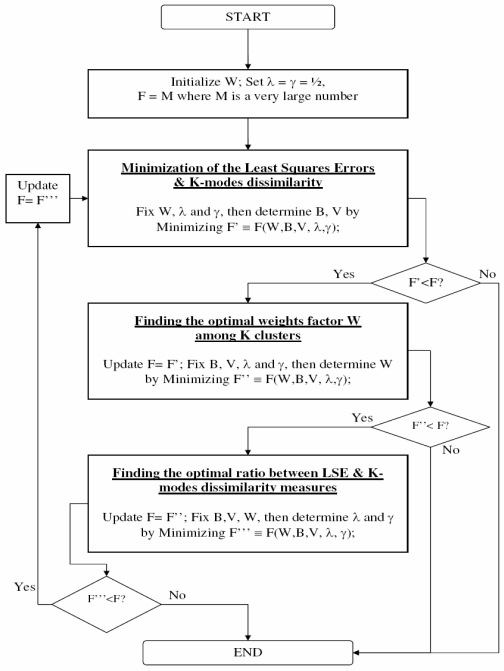
Step 1. Choose an initial matrix W and set $\lambda_{n,k} = \gamma_{n,k} = \frac{1}{2}$, for all n and k

Step 2. Given W, λ and γ, determine B and V such that $F(W, B, V, \lambda, \gamma)$ is minimized. If the objective function value is improved, goto Step 3, otherwise stop.

Step 3. Given B, V, λ and γ, determine W such that $F(W, B, V, \lambda, \gamma)$ is minimized. If the objective function value is improved, goto Step 4, otherwise stop.

Step 4. Given B, V, W, determine λ and γ such that $F(W, B, V, \lambda, \gamma)$ is minimized. If the objective function value is improved, goto Step 2, otherwise stop.

SSRM flowchart



Results – synthetic data

100 test cases for each σ where σ is the variation from synthetic linear equations

	Sample 1			Sample 2		
σ	Accuracy (%)	# of iteration	(λ, γ)	Accuracy (%)	# of iteration	(λ, γ)
0.1	99.00	49.27	(0.92,0.08)	95.93	48.50	(0.93,0.07)
0.3	96.30	24.97	(0.75,0.25)	92.03	25.24	(0.77,0.23)
0.5	92.68	19.74	(0.63,0.37)	86.34	19.77	(0.65,0.35)
0.8	88.35	20.27	(0.50,0.50)	78.06	18.90	(0.52,0.48)
1.0	85.12	46.38	(0.42,0.58)	76.81	19.34	(0.45,0.55)
1.5	80.72	46.38	(0.29,0.71)	72.06	25.79	(0.31,0.69)
2.0	68.14	111.5	(0.21,0.79)	70.33	37.89	(0.22,0.78)

Linear regression

Table 3: Average clustering results.

	Sample 3			Sample 4		
σ	Accuracy (%)	# of iteration	(λ, γ)	Accuracy (%)	# of iteration	(λ, γ)
0.5	80.22	24.85	(0.55,0.45)	73.87	29.65	(0.55,0.45)
1.0	71.02	33.84	(0.40,0.60)	60.74	42.44	(0.40,0.60)
1.5	61.27	53.29	(0.29,0.71)	53.29	57.77	(0.29,0.71)
2.0	56.47	71.09	(0.21,0.79)	48.76	67.61	(0.21,0.79)

Table 4: Average clustering results.

Synthetic results – quadratic regression

	Sample	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	$\sigma = 0.8$	$\sigma = 1.0$	$\sigma = 1.5$	$\sigma = 2.0$
Accuracy (%)	A	99.11	97.84	96.89	95.37	93.88	85.84	77.54
	B	98.94	97.58	96.58	94.88	93.06	89.22	85.14
	C	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	D	100.0	100.0	99.94	99.84	99.84	99.44	99.32
Number of Iterations	A	69.20	31.44	20.37	17.09	15.28	10.74	10.38
	B	67.17	30.15	19.06	15.67	13.94	11.72	10.67
	C	118.9	38.29	27.74	19.91	16.55	12.94	11.33
	D	228.1	130.1	92.78	67.89	67.89	42.83	34.73
(λ, γ)	A	(0.87, 0.13)	(0.61, 0.39)	(0.46, 0.54)	(0.33, 0.67)	(0.28, 0.72)	(0.14, 0.80)	(0.09, 0.91)
	B	(0.87, 0.13)	(0.62, 0.38)	(0.47, 0.53)	(0.33, 0.67)	(0.28, 0.72)	(0.20, 0.80)	(0.15, 0.85)
	C	(0.72, 0.28)	(0.47, 0.53)	(0.35, 0.65)	(0.25, 0.75)	(0.22, 0.78)	(0.16, 0.84)	(0.13, 0.87)
	D	(0.56, 0.44)	(0.25, 0.75)	(0.16, 0.84)	(0.09, 0.91)	(0.09, 0.91)	(0.05, 0.95)	(0.03, 0.97)

Table 5: Average clustering results for quadratic regression models.

Real data

German data set

– 1000 consumer credit records with numerical and categorical attributes
– Classifiable into ***Good Credit*** and ***Bad Credit***

Some Categorical attributes:
Status of checking account
Credit history
…
Job
Foreign worker

Some Numerical attributes:
Disposable income
Age
Number of existing credits
….

Results

Average clustering accuracy is 68.4% compared with given Good/Bad clusters

Authors discuss various differences and similarities