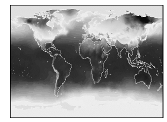
CS454 Topics in Advanced Computer Science
Introduction to Data Mining

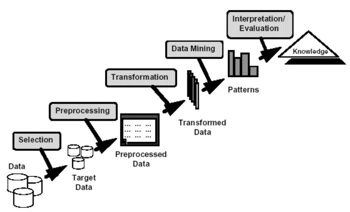
Chengyu Sun
California State University, Los Angeles

Why Data Mining?



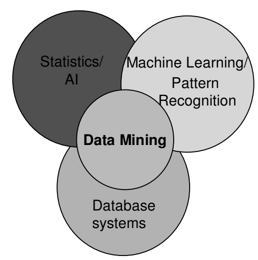
Data Mining

- Extracting knowledge from large amounts of data



Origins of Data Mining

- Traditional techniques may not be suitable due to
  - Enormity of data
  - High dimensionality of data
  - Heterogeneous, distributed nature of the data



Data Mining Problems

- Mining frequent patterns
- Prediction
  - Classification and regression
- Clustering

Mining Frequent Itemsets

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Beer, Coke, Diaper, Milk

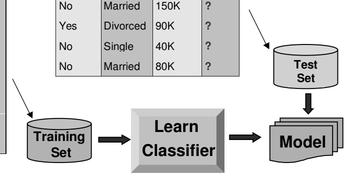
Frequent Itemsets:
{Coke, Milk}
{Beer, Diaper, Milk}

Association Rules:
{Milk} --> {Coke}
{Diaper, Milk} --> {Beer}

Classification

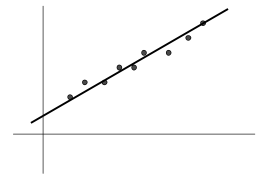
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



Regression

- Record (x,y)
  - x: predictor variable
  - y: response variable
- Model
  - $y = w_0 + w_1 x$

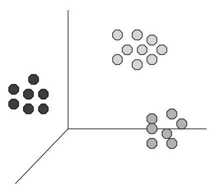


Clustering

- Euclidean Distance Based Clustering in 3-D space.

Intracluster distances are minimized

Intercluster distances are maximized



Mining Frequent Patterns

Sales Transactions

TID	Transactions
1	Beef, Chicken, Milk
2	Beef, Cheese
3	Cheese, Boots
4	Beef, Chicken, Cheese
5	Beef, Chicken, Clothes, Cheese, Milk
6	Chicken, Clothes, Milk
7	Chicken, Clothes, Milk
8	Beef, Milk

Support Count

- The support count, or frequency, of a itemset is the number of the transactions that contain the itemset
  - Item, Itemset, and Transaction
- Examples:
  - `support_count({beef})=5`
  - `support_count({beef,chicken,milk})=??`

Frequent Itemset

- An itemset is frequent if its support count is greater than or equals to a minimum support count threshold
  - `support_count(X)≥min_sup`

The Need for Closed Frequent Itemsets

- Two transactions
  - $\langle a_1, a_2, \ldots, a_{100} \rangle$ and $\langle a_1, a_2, \ldots, a_{50} \rangle$
- `min_sup=1`
- # of frequent itemsets??

Closed Frequent Itemset

- An itemset **X** is closed if there exists no *proper superset* of **X** that has the same support count
- A closed frequent itemset is an itemset that is both *closed* and *frequent*

Closed Frequent Itemset Example

- Two transactions
  - $\langle a_1, a_2, \ldots, a_{100} \rangle$ and $\langle a_1, a_2, \ldots, a_{50} \rangle$
- `min_sup=1`
- Closed frequent itemset(s)??

Maximal Frequent Itemset

- An itemset **X** is a maximal frequent itemset if **X** is frequent and there exists no *proper superset* of **X** that is also frequent
- Example: if `{a,b,c}` is a maximal frequent itemset, which one of these *cannot* be a MFI
  - `{a,b,c,d}, {a,c}, {b,d}`

Maximal Frequent Itemset Example

- Two transactions
  - $\langle a_1, a_2, \ldots, a_{100} \rangle$ and $\langle a_1, a_2, \ldots, a_{50} \rangle$
- `min_sup=1`
- Maximal frequent itemset(s)??
- Maximal frequent itemset vs. closed frequent itemset??

From Frequent Itemsets to Association Rules

- `{chicken,cheese}` is a frequent set
- `{chicken}⇒{cheese}`??
- Or is it `{cheese}⇒{chicken}`??

Association Rules

- **A⇒B**
  - **A** and **B** are itemsets
  - **A**∩**B**=∅

Support

- The support of **A⇒B** is the percentage of the transactions that contain **A∪B**

$$\text{support}(A \Rightarrow B) = P(A \cup B) = \frac{\text{support\_count}(A \cup B)}{|D|}$$

`P(A∪B)` is the probability that a transaction contains `A∪B`
`D` is the set of the transactions

Confidence

- The confidence of **A⇒B** is the percentage of the transactions containing **A** that also contains **B** with respect to the transactions that contain only A

$$\text{confidence}(A \Rightarrow B) = P(B \mid A) = \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$$

Support and Confidence Example

- `{chicken}⇒{cheese}`??
- `{cheese}⇒{chicken}`??

Strong Association Rule

- An association rule is strong if it satisfies both a minimum support threshold (`min_sup`) and a minimum confidence threshold (`min_conf`)
- Why do we need both *support* and *confidence*??

Association Rule Mining

- Find strong association rules
  - Find all frequent itemsets
  - Generate strong association rules from the frequent itemsets

The Apriori Property

- All nonempty subsets of a frequent itemset must also be frequent
- Or, if an itemset is not frequent, its supersets cannot be frequent either

Finding Frequent Itemsets – The Apriori Algorithm

- Given `min_sup`
- Find the frequent 1-itemsets L_1
- Find the the frequent k-itemsets L_k by joining the itemsets in L_{k-1}
- Stop when L_k is empty

Apriori Algorithm Example

beef	1
chicken	2
milk	3
cheese	4
boots	5
clothes	6

TID	Transactions
1	1, 2, 3
2	1, 4
3	4, 5
4	1, 2, 4
5	1, 2, 6, 4, 3
6	2, 6, 3
7	2, 6, 3
8	1, 3

- Min Support 25%

L_1

- Scan the data once to get the count of each item
- Remove the items that do not meet `min_sup`

C_1	support_count	L_1
{1}	5	{1}
{2}	5	{2}
{3}	5	{3}
{4}	4	{4}
{5}	1	
{6}	3	{6}

L_2

- $C_2=L_1 \times L_1$
- Scan the dataset again for the support_count of C_2, then remove non-frequent itemsets from C_2, i.e. $C_2 \rightarrow L_2$

C_2	support_count	L_2
{1,2}	3	{1,2}
{1,3}	3	{1,3}
{1,4}	3	{1,4}
{1,6}	1	
{2,3}	4	{2,3}
{2,4}	2	{2,4}
{2,6}	3	{2,6}
{3,4}	1	
{3,6}	3	{3,6}
{4,6}	1	

L_3

- ??

From L_{k-1} to C_k

- Let l_i be an itemset in L_{k-1}, and $l_i[j]$ be the jth item in l_i
- Items in an itemset are sorted, i.e. $l_i[1]<l_i[2]<\ldots<l_i[k-1]$
- l_1 and l_2 are joinable if
  - Their first k-2 items are the same, and
  - $l_1[k-1]<l_2[k-1]$

From C_k to L_k

- Reduce the size of C_k using the Apriori property
  - any (k-1)-subset of an candidate must be frequent, i.e. in L_{k-1}
- Scan the dataset to get the support counts

Generate Association Rules from Frequent Itemsets

- For each frequent itemset l, generate all nonempty subset of l
- For every nonempty subset of s of l, output rule $s \Rightarrow (l-s)$ if $conf(s \Rightarrow (l-s)) \geq min\_conf$

Limitations of the Apriori Algorithm

- Multiple scans of the datasets
  - How many??
- Need to generate a large number of candidate sets

Recap

- Support and confidence
- Frequent itemset
  - Closed and maximal
- Association rules
- Apriori Algorithm

Classification

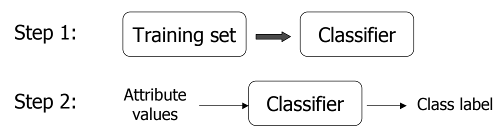
A Classification Problem

- Is a loan to a person who is 45 years old, divorced, renting an apartment, with two kids and annual income of 100K high risk or low risk?

Terminology and Concepts ...

- Record (or tuple)
  - Attributes
    - E.g. age, marital status, # of kids, owns home or not, credit score ...
  - Class label
    - E.g. high risk, low risk ...
- Classification: predict the class label with given attribute values

... Terminology and Concepts



- Classifier (or model)
- Training set: records with known class labels that are used to construct (i.e. *train*) the classifier

Classification vs. Regression

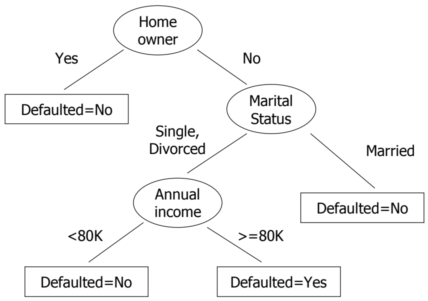
- Classification predicts categorical attribute values
- Regression predicts *continuous* numerical attribute values

SID	HW1	HW2	HW3	Final	Pass/Fail
1	40	60	70	95	Passed
2	10	15	11	65	Failed
3	30	45	40	75	Passed
4	35	50	35	?	?

A Training Set

TID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

A Decision Tree



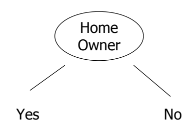
Decision Tree Induction

- Let D be the set of training record associated with current node
  - If all record in D belong to the same class C, current node is a leaf node and is labeled as C.
  - If D contains records that belong to more than one class, select an attribute to split D into subsets, and create a child node for each subset. *Apply the algorithm recursively on each child node.*

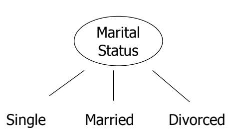
Terminating Conditions

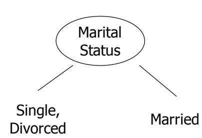
- All records in D belong to the same class
- No more attribute to split
  - *Class label??*
- No records associated with the node
  - *Class label??*

Split on Binary Attributes

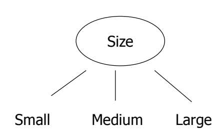


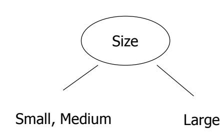
Split on Nominal Attributes



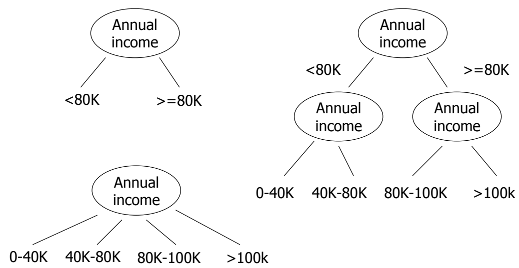


Split on Ordinal Attributes





Split on Continuous Attributes



Splitting Attribute Selection

- After a split, ideally each subset would "pure", i.e. contains only one class of records

Gender	**Age**	**Preferred color**
female	20	pink
male	20	black
female	15	pink
male	15	black

Entropy

$$Entropy(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- p_i is the fraction of records in D that belongs to class C_i
- m is the number of classes in D

Entropy Example

- Preferred color
  - 2 black and 2 pink??
  - 3 black and 1 pink??
  - 4 black??

Information Gain

- Suppose D is split into v subsets on attribute A

$$Gain(A) = Entropy(D) - \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Entropy(D_j)$$

Information Gain Example

- Preferred color
  - Gain(Gender)??
  - Gain(Age)??

Split Information

- Information gain favors attributes with lots of distinct values
- *Split information* can be used to "normalized" information gain

$$SplitInfo(A) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

Gain Ratio

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

Probabilistic Relationship between Attributes and Class

- Ten middle-aged, divorced, male borrowers have defaulted on their loans, but would the 11^{th} one default as well?

Bayes' Theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- Prior and posterior probabilities
  - P(A) and P(A|B)
  - P(B) and P(B|A)

Bayesian Classification

$$P(C_i \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C_i)P(C_i)}{P(\mathbf{X})}$$

- **x** is a given record with attribute values $(x_1, x_2, ..., x_n)$, and C_i is a class
- $P(C_i|\mathbf{X})$ is the probability of **x** belonging to class C_i given **x**'s attribute values
- We predict that **x** belong to C_i if $P(C_i|X)>P(C_j|X)$ for $j \neq i$

Calculate $P(C_i|\mathbf{X})$

- P(X) does not need to be calculated
  - *Why??*
- $P(C_i)$??
- $P(\mathbf{X}|C_i)$??

Naive Bayesian Classification

- $\mathbf{X}=(x_1,x_2,\ldots,x_n)$
- Assume the attribute values are conditionally independent of one another (the *naive* assumption)

$$P(\mathbf{X}\mid C_i)=\prod_{i=1}^{n}P(x_i\mid C_i)$$
$$=P(x_1\mid C_i)\times P(x_2\mid C_i)\times\cdots\times P(x_n\mid C_i)$$

Attribute A_k is Categorical

- $P(x_k|C_i)$ is the fraction of number of records in C_i with value x_k for attribute A_k

Attribute A_k is Continuous-valued ...

- Assume A_k follows a Gaussian distribution with a mean μ and standard deviation σ

$$\mu=\frac{1}{N}\sum_{i=1}^{N}x_i \qquad s=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2}$$

s: sample standard deviation

... Attribute A_k is Continuous-valued

$$g(x,\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(x_k\mid C_i)\rightarrow g(x_k,\mu_{c_i},\sigma_{c_i})$$

Sample Data

TID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes
11	No	Married	120K	??

Naive Bayesian Classification Example ...

- P(Default=N|HO=N,MS=M,AI=120K)
- P(Default=Y|HO=N,MS=M,AI=120K)

... Naive Bayesian Classification Example

- Annual Income, Default=No
  - μ=110, σ=54.54
  - P(AI=120K|No)=0.0072
- Annual Income, Default=Yes
  - μ=90, σ=5
  - P(AI=120K|Yes)=1.2×10^{-9}

Avoid Zero $P(x_k|C_i)$

- A zero $P(x_k|C_i)$ would make the whole $P(\mathbf{X}|C_i)$ zero
- To avoid this problem, add 1 to to each count, assuming the training set is sufficiently large that the effect of adding one is negligible
- Example
  - Low income:0
  - Medium income: 990
  - High income: 10

About Naive Bayesian Classification

- The most accurate classification *if the conditional independence assumption holds*
- In practice, some attributes may be correlated
  - E.g. education level and annual income

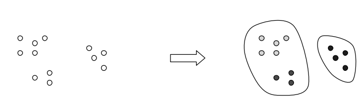
Recap

- Terminology
  - Record, attribute, class label
  - Training, training set, classifier
- Decision Tree Induction
  - Entropy, information gain, gain ratio
- Naïve Bayesian Classification
  - Probability calculation

Clustering

Clustering

- Group *similar* objects together
- Applications
  - Identify users who share similar interests
  - Automatically generate concept hierarchies
  - Reduce algorithmic complexity
  - ...



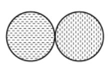
Types of Clusters

- Well separated
- Prototype based
- Contiguity based
- Density based
- Conceptual clusters

Well-separated Clusters

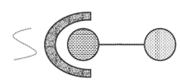
- Each point is closer to all of the points in its cluster than to any point in another cluster

Prototype-based Clusters



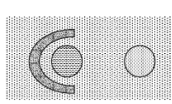
- ??

Contiguity-based Clusters



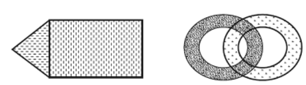
- ??
- A cluster can be considered as a *connected component* in a graph

Density-based Clusters



- A cluster is a dense region of objects surrounded by a region of low density

Conceptual Clusters



- A cluster is a set of objects that share *some property*

Types of Clustering

- Partitional vs. Hierarchical
- Exclusive vs. Overlapping vs. Fuzzy
- Complete vs. Partial

Similarity Measure

TID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No

- Is #1 more similar to #2 or #3?

Interval-Scaled Attributes

- Continuous-valued data measured with a linear scale (vs. exponential or logarithmic scale)

Distance Measures

- $\mathbf{X}=(x_1,x_2,,\ldots,x_n)$ and $\mathbf{Y}=(y_1,y_2,\ldots,y_n)$
  - E.g. `(1,2)` and `(3,5)`

Euclidean Distance:

$$dist(\mathbf{X},\mathbf{Y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Manhattan Distance:

$$dist(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{n}|x_i - y_i|$$

Minkowski Distance

$$dist(\mathbf{X},\mathbf{Y}) = \sqrt[p]{\sum_{i=1}^{n}|x_i - y_i|^p}$$

- p=1 (Manhattan Distance)
  - a.k.a. L_1 norm or L_1 distance
- p=2 (Euclidean Distance)
  - a.k.a. L_2 norm or L_2 distance

Requirements of Distance Functions

- dist($\mathbf{X}$,$\mathbf{Y}$)≥0
- dist($\mathbf{X}$,$\mathbf{X}$)=0
- dist($\mathbf{X}$,$\mathbf{Y}$)=dist($\mathbf{Y}$,$\mathbf{X}$)
- dist($\mathbf{X}$,$\mathbf{Y}$)≤dist($\mathbf{X}$,$\mathbf{Z}$)+dist($\mathbf{Z}$,$\mathbf{Y}$)
  - *Triangular Inequality*

Problem of Units

- (10m,2km) and (5m,2.1km)?
- (10m,200lb) and (5m, 210lb)?

Standardize Interval-Scaled Attributes

- Given attribute A with values $a_1, a_2, ..., a_n$

Mean: $\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$

Mean absolute deviation: $s = \frac{1}{n}\sum_{i=1}^{n} |a_i - \bar{a}|$

Standardized measurement (*z-score*): $z_i = \frac{a_i - \bar{a}}{s}$

Binary Attributes

- Symmetric
  - E.g. gender
- Asymmetric
  - E.g. HIV test result

Contingency Table for Binary Attributes

		Record **Y** 1	0
Record **X**	1	q	r
	0	s	t

- Example
  - X=(1,1,0,1,0,0,0), Y=(0,1,0,1,0,1,0)

Distance Measure for Symmetric Binary Attributes

Similarity: $sim(\mathbf{X}, \mathbf{Y}) = \frac{q+t}{q+r+s+t}$

Dissimilarity: $dsim(\mathbf{X}, \mathbf{Y}) = \frac{r+s}{q+r+s+t}$

Distance: ??

Distance Measure for Asymmetric Binary Attributes

Similarity (Jaccard Coefficient): $sim(\mathbf{X}, \mathbf{Y}) = \frac{q}{q+r+s}$

Dissimilarity: $dsim(\mathbf{X}, \mathbf{Y}) = \frac{r+s}{q+r+s}$

Distance: ??

Categorical Attributes

- Example
  - Marital status: single, married, divorced
- dist(**X**,**Y**)=(p−m)/p
  - m: number of attribute matches
  - p: total number of attributes
- Or, encode each state with a binary attribute

Ordinal Attributes

- Example
  - Grade: F, D, C, B, A
- Given an attribute with M possible values {1,2,...,M}, map value a to the range of [0.0,1.0]

$$z = \frac{a-1}{M-1}$$

Records with Mixed Types of Attributes ...

$$dist(\mathbf{X},\mathbf{Y}) = \frac{\sum_{i=1}^{n} \delta_i \times dist(x_i, y_i)}{\sum_{i=1}^{n} \delta_i}$$

- δ_i is the weight of the ith attribute a_i's contribution toward the overall distance
  - 0 if x_i or y_i is missing, or a_i is asymmetric binary and $x_i=y_i=0$
  - 1 othewise

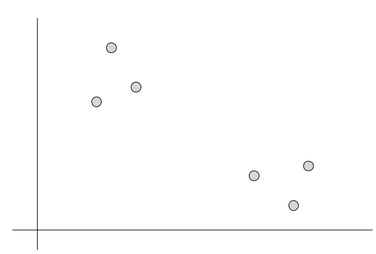
... Records with Mixed Types of Attributes

- $dist(x_i, y_i)$
  - Interval-based: $|x_i-y_i|/(max(a_i)-min(a_i))$
  - Binary or categorical: 0 if $x_i=y_i$; 1 otherwise
  - Ordinal: treat as interval-based using z_i

K-Means

- Input: dataset D and number of clusters k
- Algorithm
  1. Randomly choose k objects as cluster centers
  2. Assign each object to the closest cluster center
  3. Update each cluster center
  4. Repeat 2 until there is no reassignment occurs

K-Means Example



Limitations of K-Means

- Only handles well-separated, spherical-shaped clusters well
- Problem with outliers
- Requires the notion of *centroid*

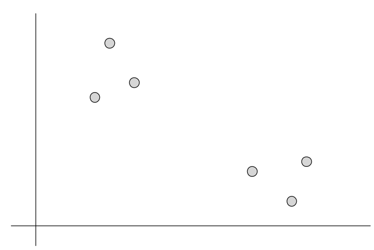
K-Medoids

- Instead of using mean/centroid, use medoid, i.e. representative object
- Objective function: sum of the distances of the objects to their medoid
- Differs from K-Means in how the medoids are updated

PAM (Partition Around Medoids)

1. Randomly choose k objects as initial medoids
2. For each non-medoid object x
   For each medoid c_i
   calculate the reduction of the total distance if c_i is replaced by x
3. Replace the c_i with x that results in maximum total distance reduction
4. Repeat Step 2 until the total distance cannot be reduced
5. Assign each object to its closest mediod

PAM Example



Recap

- Types of clusters
- Similarity/distance measures
- K-Means and K-Mediods

Reading Assignment

- Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, by Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan.
- Discovery of Significant Usage Patterns from Clusters of Clickstream Data, by Lin Lu, Margaret Dunham, Yu Meng.