Learning And Mining from DatA

http://lamda.nju.edu.cn

# Multi-Instance Multi-Label Learning
# (多示例多标记学习)


Zhi-Hua Zhou

http://cs.nju.edu.cn/zhouzh/

LAMDA Group,
National Key Laboratory for Novel Software Technology,
Nanjing University, China


Joint work with

Min-Ling Zhang, Sheng-Jun Huang, Yu-Feng Li

Thanks to

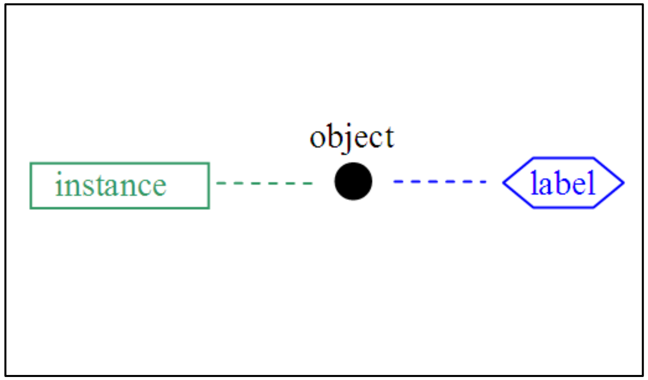
De-Chuan Zhan, James Kwok

# Traditional Setting



In traditional supervised learning:

- A real-world object is represented by an instance (feature vector)
- The instance is associated with a label which indicates the concerned characteristics (such as categorization) of the object




$\mathcal{X}$ - the instance space
$\mathcal{Y}$ - the set of class labels

**The task**:

To learn a function $f : \mathcal{X} \to \mathcal{Y}$ from a given data set $\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ where $x_i \in \mathcal{X}$ is an instance and $y_i \in \mathcal{Y}$ is the known label of $x_i$

# Ambiguous Data



*Elephant ?* *Lion ?* *Grassland?*
*Tropic ?* *Africa ?* *… …*

# Ambiguous Data


**Real Madrid Sets Off for World Tour**

Thu Jul 14, 1:00 PM ET

MADRID, Spain - Real Ma
beginning of a world tour th
Bangkok.

It will be the second trip to
six weeks for Beckham an
Michael Owen, who were h
England's national team in
Portugese midfielder Luis I
make the journey.

Real Madrid's first game w
Chicago on Saturday agair
Guadalajara, Mexico. On N
team plays the Los Angele
before moving on to Asia.

AP Photo: Soccer star David Beckham adjusts his tie at a news conference in Singapore on July...

NEWS ALERTS

*Sports ?*

*Tour ?*

*Entertainment ?*

*Economy ?*

*… …*

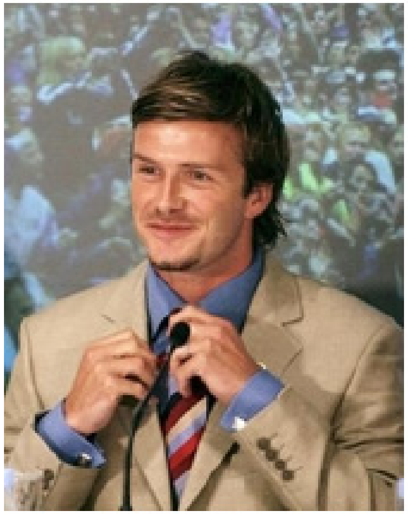
# Outline



- Previous research
  - Multi-label learning
  - Multi-instance learning
- MIML: A new framework
  - Why MIML?
  - Solving MIML - by degeneration; by regularization
  - No access to raw data – how to do?
  - Usefulness in single-label problems

# To Address the Ambiguity

**Real Madrid Sets Off for World Tour**



AP Photo: Soccer star David Beckham adjusts his tie at a news conference in Singapore on July...

NEWS ALERTS

Thu Jul 14, 1:00 PM ET

MADRID, Spain - Real Ma
beginning of a world tour th
Bangkok.

It will be the second trip to
six weeks for Beckham an
Michael Owen, who were h
England's national team in
Portugese midfielder Luis l
make the journey.

Real Madrid's first game w
Chicago on Saturday agair
Guadalajara, Mexico. On M
team plays the **Los Angele**
before moving on to Asia.

*Sports ?*

*Tour ?*

*Entertainment ?*

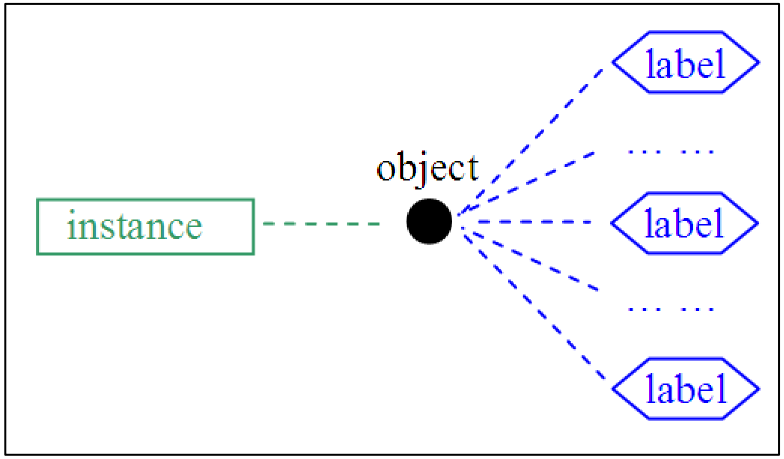
*Economy ?*

… …

Multiple labels

# Multi-Label Learning






**MLL task**:
To learn a function $f_{MLL}: \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from a given data set $\{(x_1, Y_1), (x_2, Y_2), \cdots, (x_m, Y_m)\}$, where $x_i \in \mathcal{X}$ is an instance and $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^{(i)}, y_2^{(i)}, \cdots, y_{l_i}^{(i)}\}$, $y_k^{(i)} \in \mathcal{Y}$ $(k = 1, 2, \cdots, l_i)$.

$\mathcal{X}$ - the instance space

$\mathcal{Y}$ - the set of class labels

$l_i$ - the number of labels in $Y_i$

# Multi-Label Learning Algorithms



- Decomposing the task into multiple binary classification problems each for a class
  - ✓ MLSVM [Boutell et al., PR04]
  - ✓ . . . . . .
- Considering the ranking among labels
  - ✓ BoosTexter [Schapire & Singer, MLJ00]
  - ✓ BP-MLL [Zhang & Zhou, TKDE06]
  - ✓ RankSVM [Elisseeff & Weston, NIPS'01]
  - ✓ . . . . . .
- Exploring the class correlation
  - ✓ Probabilistic generative models [McCallum, AAAI'99w; Ueda & Saito, NIPS'02]
  - ✓ Maximum entropy methods [Ghamrawi & McCallum, CIKM'05; Zhu et al., SIGIR'05]
  - ✓ . . . . . .

# MLL Evaluation Measures [Schapire & Singer, MLJ00]

Given:

$S$ : A set of multi-label examples $\{(x_1,Y_1), \ldots, (x_m,Y_m)\} \subseteq (\mathcal{X} \times \mathcal{Y})^m$

$f: \mathcal{X}\times\mathcal{Y} \rightarrow \mathbb{R}$, a ranking predictor ($h$ is the corresponding multi-label predictor)

Definitions:

Hamming Loss: $\text{hamloss}_S(f) = \frac{1}{m \times k}\sum_{i=1}^{m}\left|h(x_i)\Delta Y_i\right|$ ↓

One-error: $\text{one-err}_S(f) = \frac{1}{m}\sum_{i=1}^{m}\left|\{i \mid H(x_i) \notin Y_i\}\right|$, where $H(x) = \arg\max_{l\in\mathcal{Y}} f(x,l)$ ↓

Coverage: $\text{coverage}_S(f) = \frac{1}{m}\sum_{i=1}^{m}\max_{y\in Y_i} rank_f(x_i,y) - 1$ ↓

Ranking Loss: $\text{rankloss}_S(f) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|Y_i||\overline{Y_i}|}\left|\{(l_0,l_1)\in \overline{Y_i}\times Y_i \mid f(x_i,l_1)\le f(x_i,l_0)\}\right|$ ↓

Average Precision: $\text{avgprec}_S(f) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|Y_i|}\sum_{l\in Y_i}\frac{\left|\{l' \in Y_i \mid f(x_i,l') > f(x_i,l)\}\right|}{\left|\{j\in\{1,\ldots,k\} \mid f(x_i,j) > f(x_i,l)\}\right|}$ ↑

# Representative MLL Algorithms – MLSVM

Learning And Mining from DatA
http://lamda.nju.edu.cn

Use multi-label data more than once when training the binary SVMs

Using each example as a positive example of each of the classes to which it belongs

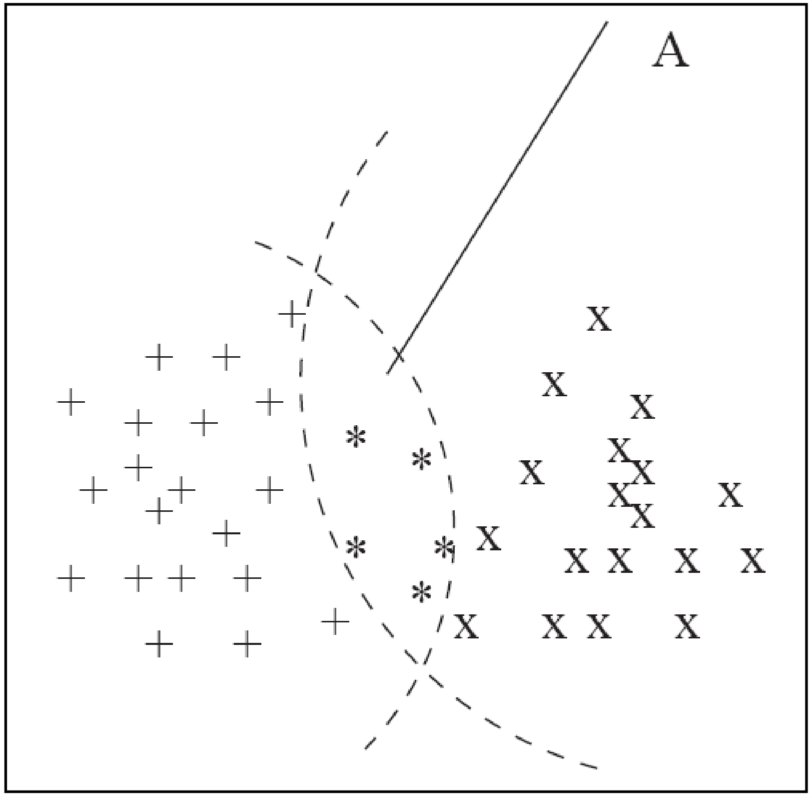



Figure reprinted from [Boutell et al., PR04]

# Representative MLL Algorithms – RankSVM



To minimize the ranking loss $\frac{1}{|\mathbf{Y}||\bar{\mathbf{Y}}|}\left|(i,j) \in \mathbf{Y} \times \bar{\mathbf{Y}} \text{ s.t. } r_i(x) \le r_j(x)\right|$ while having a large margin [Elisseeff & Weston, NIPS'01]

$$f(x) = \operatorname{sign}\left(\langle w_1, x\rangle + b_1, \ldots, \langle w_Q, x\rangle + b_Q\right)$$

The label $k$ is a "correct label" for $x$ iff $\operatorname{sign}(\langle w_k, x\rangle + b_k) > 0$

Considering the ranking loss, for $(k,l) \in \mathbf{Y} \times \bar{\mathbf{Y}}$, $\langle w_k, x\rangle + b_k$ should be bigger than $\langle w_l, x\rangle + b_l$. Thus, the margin of $(x, \mathbf{Y})$ can be expressed as

$$\min_{k \in \mathbf{Y}, l \in \bar{\mathbf{Y}}} \frac{\langle w_k - w_l, x\rangle + b_k - b_l}{\|w_k - w_l\|}$$

Thus, the optimization objective is:

$$\max_{w_j, j=1,..,Q} \quad \min_{(x,\mathbf{Y}) \in S} \min_{k \in \mathbf{Y}, l \in \bar{\mathbf{Y}}} \frac{1}{\|w_k - w_l\|^2}$$

$$\text{subject to:} \quad \langle w_k - w_l, x_i\rangle + b_k - b_l \ge 1, \ (k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$$

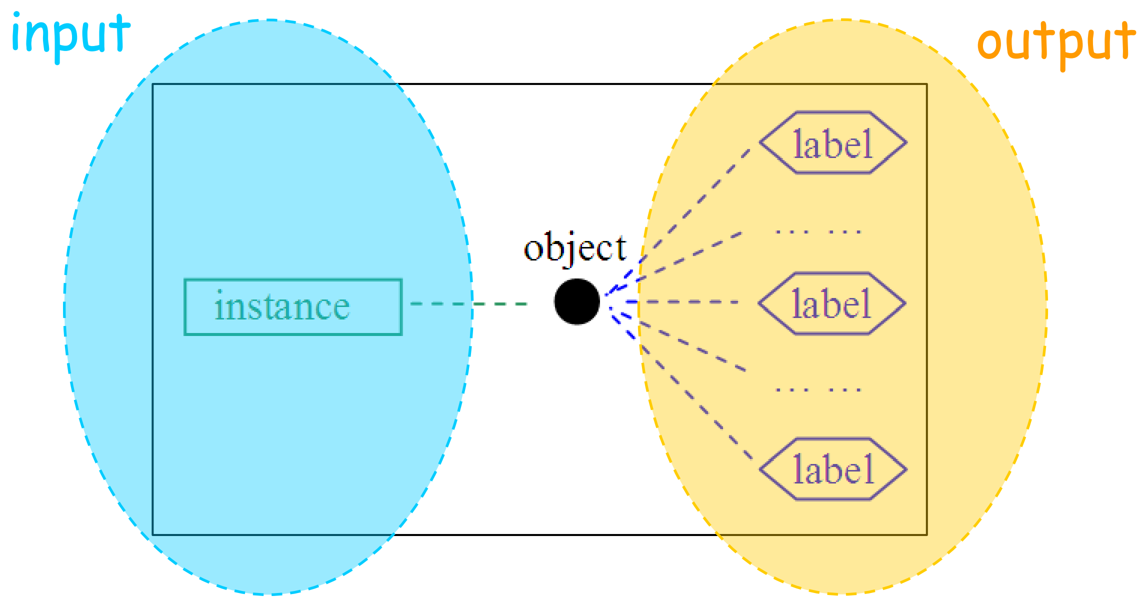
# MLL Applications



- Text categorization
  [McCallum, AAAI'99w; Schapire & Singer, MLJ00; Crammer & Singer, SIGIR'02; Ueda & Saito, NIPS'02; Cai & Hofmann, CIKM'04; Kazawa et al., NIPS'04; Rousu et al., ICML'05; Liu et al., AAAI'06; Zhang & Zhou, AAAI'07]
- Bioinformatics
  [Clare & King, PKDD01; Elisseeff & Weston, NIPS'01; Brinker et al., ECAI'06; Barutcuoglu et al., Bioinformatics06; Zhang & Zhou, AAAI'07]
- Image categorization
  [Boutell et al., PR04; Zhang & Zhou, AAAI'07]
- … …

# Input Ambiguity vs. Output Ambiguity



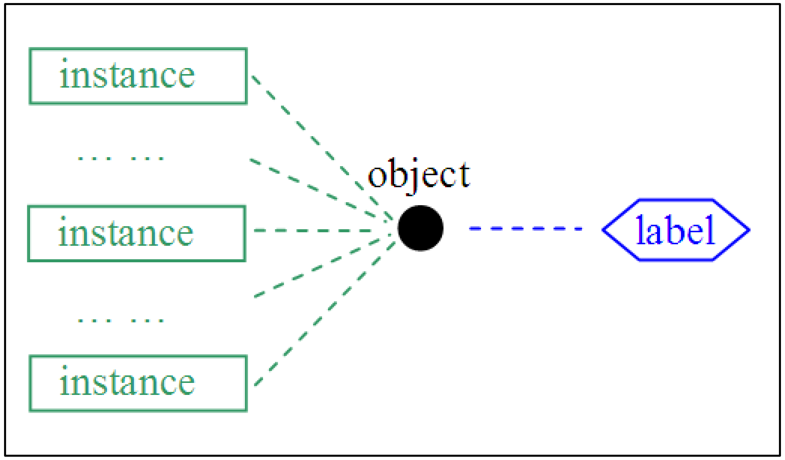



Multi-label learning only addresses the output ambiguity

How about the input ambiguity?

# Multi-Instance Learning






$\mathcal{X}$ - the instance space

$\mathcal{Y}$ - the set of class labels

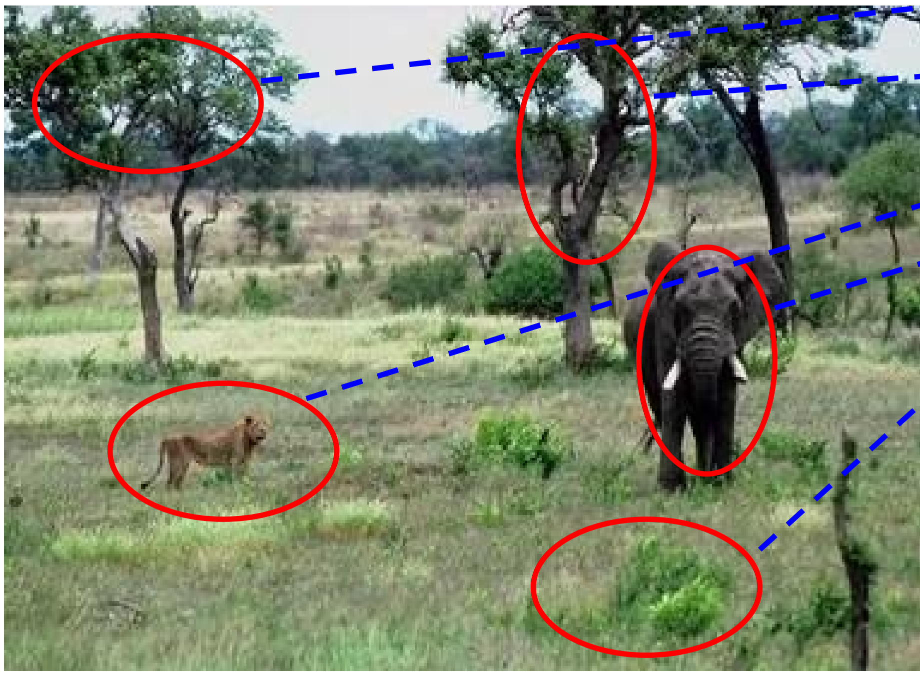
$n_i$ - the number of instances in $X_i$

**MIL task**:

To learn a function $f_{MIL} : 2^{\mathcal{X}} \to \{-1, +1\}$ from a given data set $\{(X_1, y_1), (X_2, y_2), \cdots, (X_m, y_m)\}$, where $X_i \subseteq \mathcal{X}$ is a set of instances $\{\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \cdots, \boldsymbol{x}_{n_i}^{(i)}\}$, $\boldsymbol{x}_j^{(i)} \in \mathcal{X}$ $(j = 1, 2, \cdots, n_i)$, and $y_i \in \{-1, +1\}$ is the label of $X_i$. $X_i$ is a positive bag (thus $y_i = +1$) if there exists $g \in \{1, \cdots, n_i\}$, $\boldsymbol{x}_{ig}$ is positive. Yet the value of the index $g$ is unknown.

# Why MIL is Appealing ?

Many tasks can be modeled as an MIL task



$$[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m]^\mathrm{T}$$
$$[\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m]^\mathrm{T}$$
$$\ldots$$
$$\ldots$$
$$[\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m]^\mathrm{T}$$

image ⇨ bag

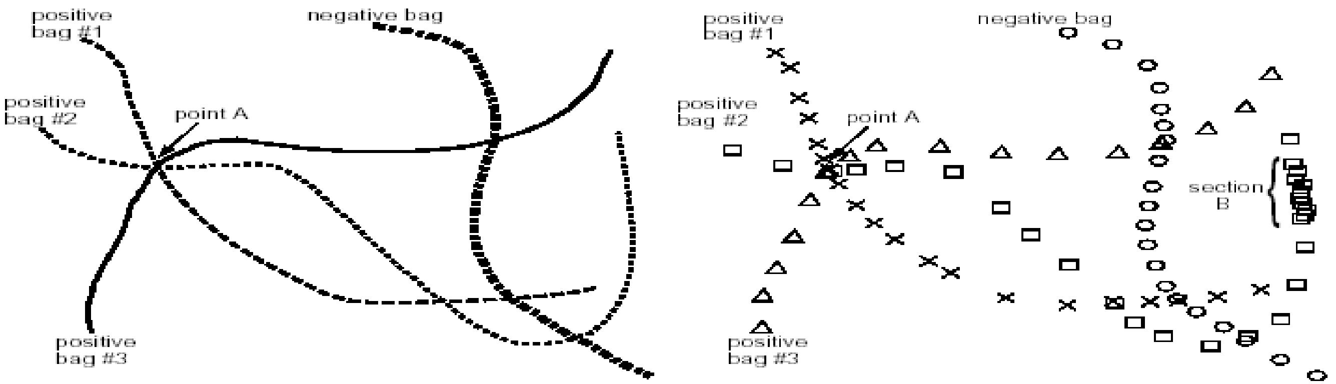
regions in the image ⇨ instances in the bag

# Multi-Instance Learning Algorithms



- ✓ Diverse Density [Maron & Lozano-Perez, NIPS'97], EM-DD [Zhou & Goldman, NIPS'01]
- ✓ kNN algorithm: Citation-kNN [Wang & Zucker, ICML'00]
- ✓ Decision tree algorithms: RELIC [Ruffo, Thesis00], ID3-MI [Chevaleyre & Zucker, CanadianAI'01]
- ✓ Rule learning algorithm: RIPPER-MI [Chevaleyre & Zucker, CanadianAI'01]
- ✓ SVM algorithms: MI-SVM [Andrews et al., NIPS'02], mi-SVM [Andrews et al., NIPS'02], DD-SVM [Chen & Wang, JMLR04]
- ✓ Ensemble algorithms: MI-Ensemble [Zhou & Zhang, ECML'03], MI-Boosting [Xu & Frank, PAKDD'04]
- ✓ Logistic regression algorithm: MI-LR [Ray & Craven, ICML'05]
- ✓ … …

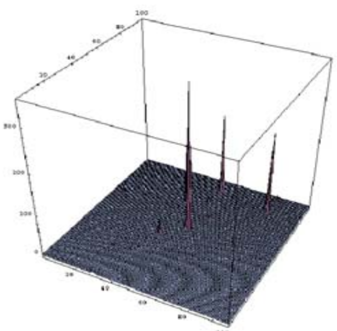
# Representative MIL Algorithms – Diverse Density




(a) The different shapes that a molecule can take on are represented as a path. The intersection point of positive paths is where they took on the same shape.

(b) Samples taken along the paths. Section B is a high density area, but point A is a high Diverse Density area.

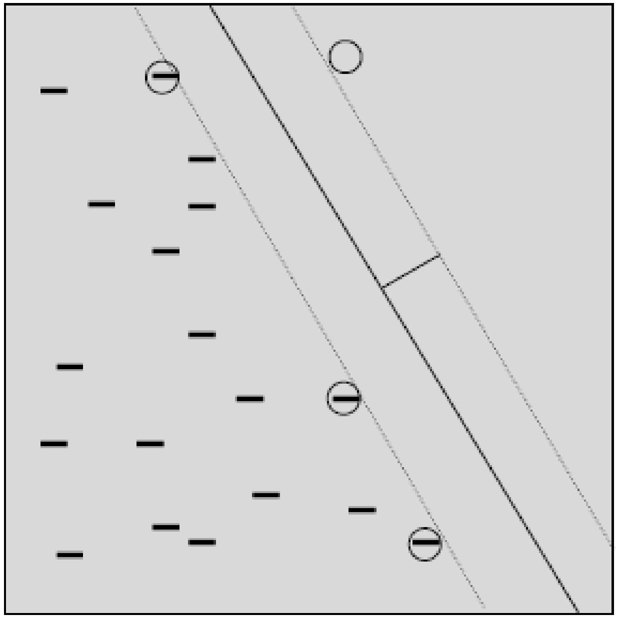
Figure 1: A motivating example for Diverse Density



To search for the point with the maximal diverse density by gradient search

every instance in positive bags is used as a start point for search

# Representative MIL Algorithms – MI-SVM



To search for the maximal margin hyperplane

the margin of a "positive bag" is the margin of its "most positive" instance

Followed by [Cheung & Kwok, ICML'06]

*MI-SVM*

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_I \xi_I$$

$$\text{s.t.} \quad \forall I:\ Y_I \max_{i \in I}(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_I,\ \xi_I \geq 0.$$

Figure reprinted from [Andrews et al., NIPS'02]

# MIL Applications



- ✓ Drug prediction [Dietterich et al., AIJ97]
- ✓ Image categorization [Maron & Ratan, ICML'98; Chen & Wang, JMLR04; Chen et al., PAMI06]
- ✓ Computer security [Ruffo, Thesis00]
- ✓ Web mining [Zhou et al., APIN05]
- ✓ Face detection [Viola et al., NIPS'05]
- ✓ … …

# Outline

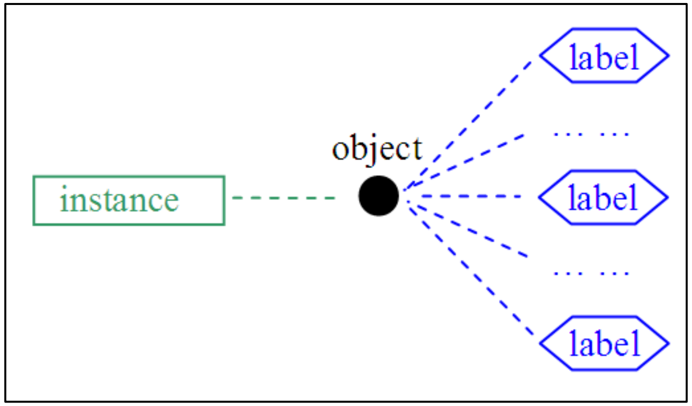


- Previous research
  - Multi-label learning
  - Multi-instance learning
- MIML: A new framework
  - Why MIML?
  - Solving MIML - by degeneration; by regularization
  - No access to raw data – how to do?
  - Usefulness in single-label problems

# A Question

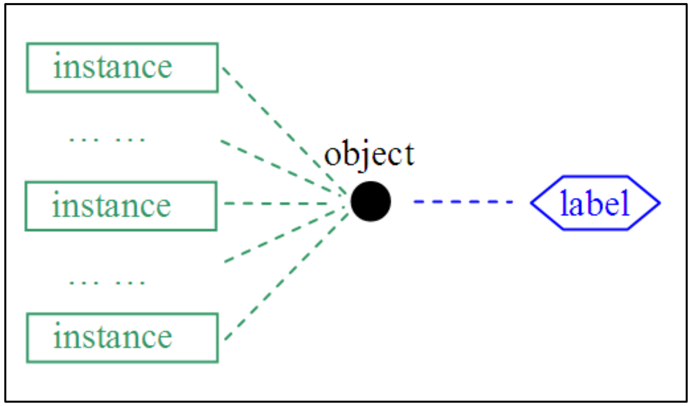


Are MLL and MIL sufficient for learning ambiguous data?




Multi-label learning

considers only the output ambiguity



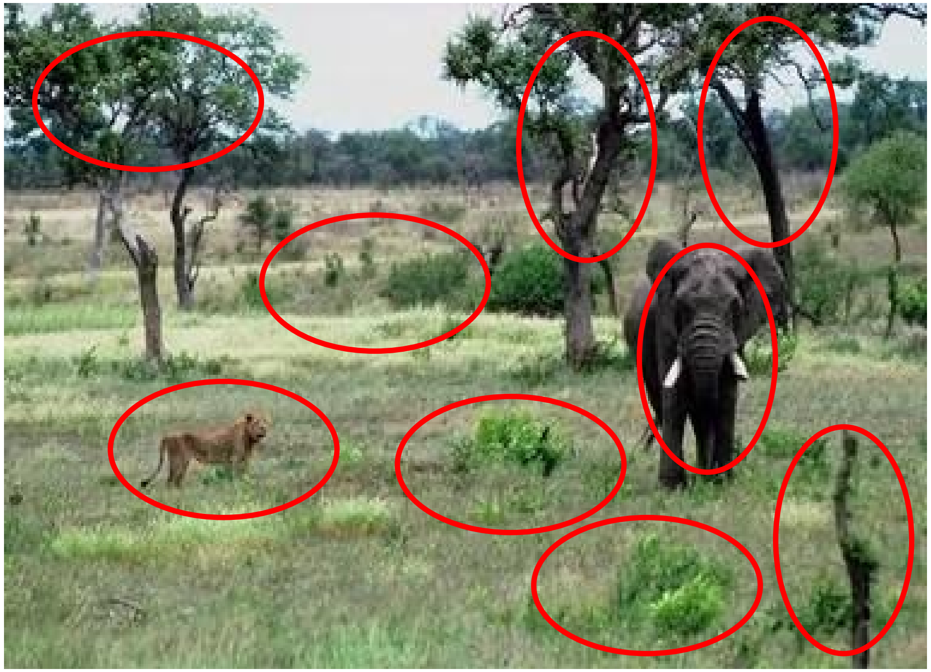

Multi-instance learning

considers only the input ambiguity

**Input and output ambiguities usually occur simultaneously !**

# For Example …

An image usually contains **multiple** regions each can be represented by an instance

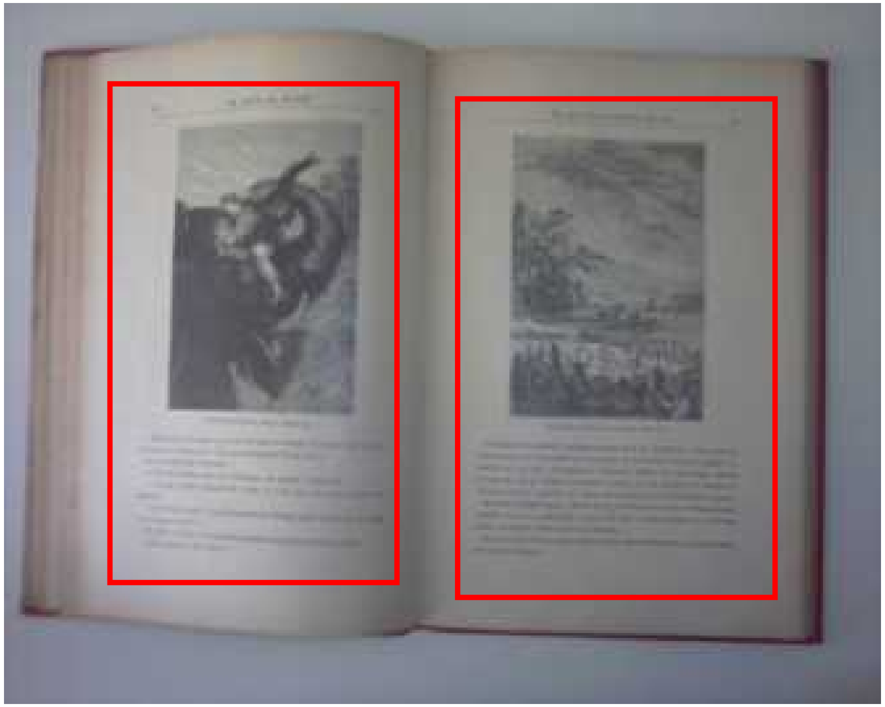


The image can simultaneously belong to **multiple** classes

*Elephant*
*Lion*
*Grassland*
*Tropic*
*Africa*
*… …*

## For Example …

A document usually contains **multiple** sections each can be represented by an instance



The document can simultaneously belong to **multiple** categories

*Scientific novel*

*Jules Verne's writing*

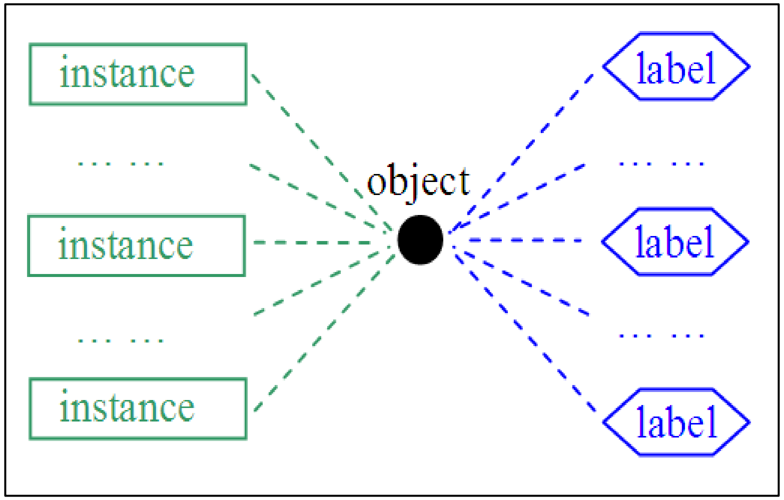
*Book on traveling*

*… …*

# Why Not ?



Why not consider the input and output ambiguities together?




**Multi-Instance Multi-Label (MIML) Learning**

# Why MIML ?



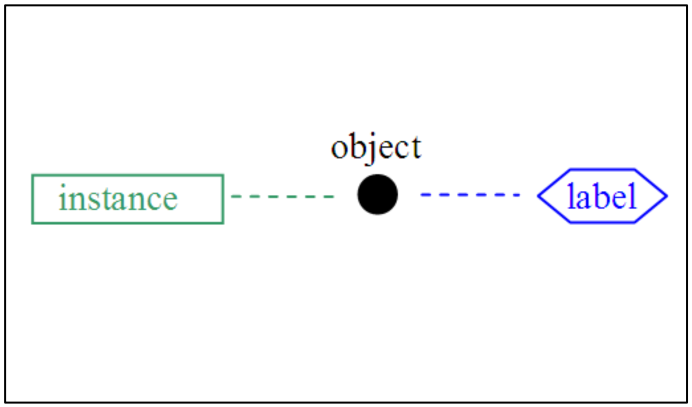
Appropriate representation is important

Having an appropriate representation is as important as having a strong learning algorithm

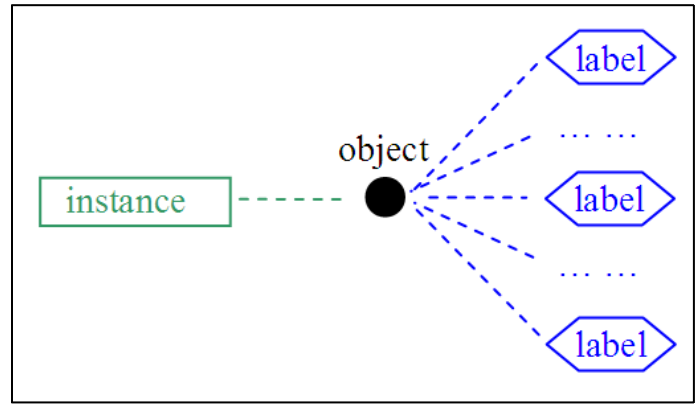
Real-world objects are usually inherited with input ambiguity as well as output ambiguity

Traditional supervised learning, multi-instance learning and multi-label learning are degenerated versions of MIML

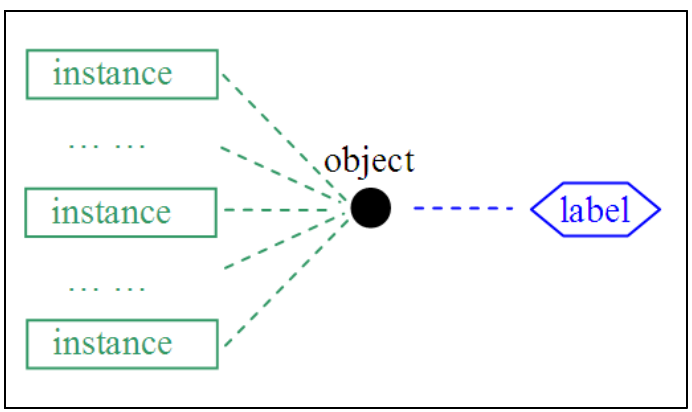
# Why MIML ? (con't)



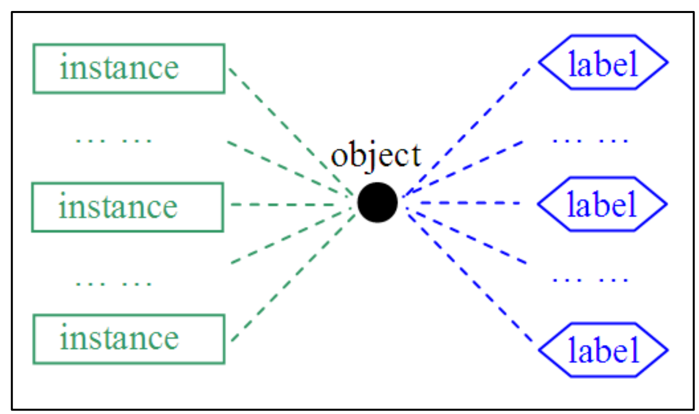

Traditional supervised learning




Multi-label learning




Multi-instance learning



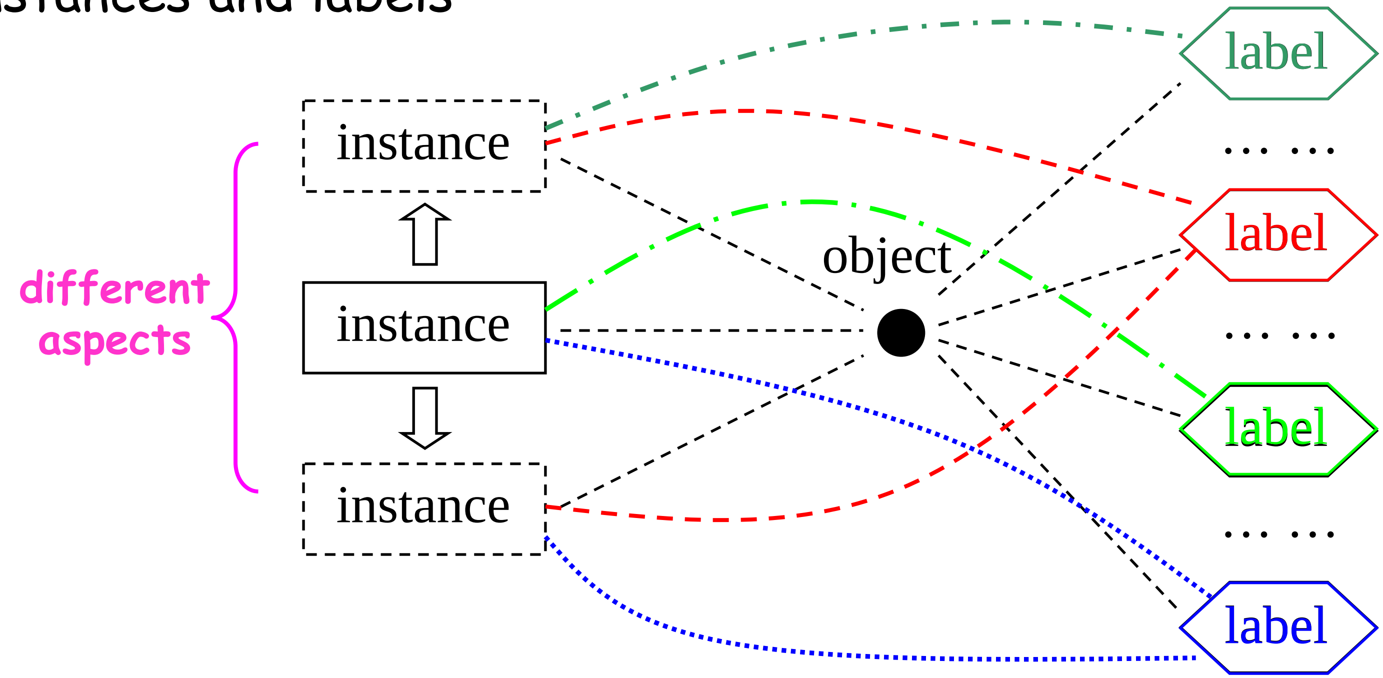

Multi-instance multi-label learning

# Why MIML ? (con't)

To learn an ***one-to-many*** mapping is an ill-posed problem

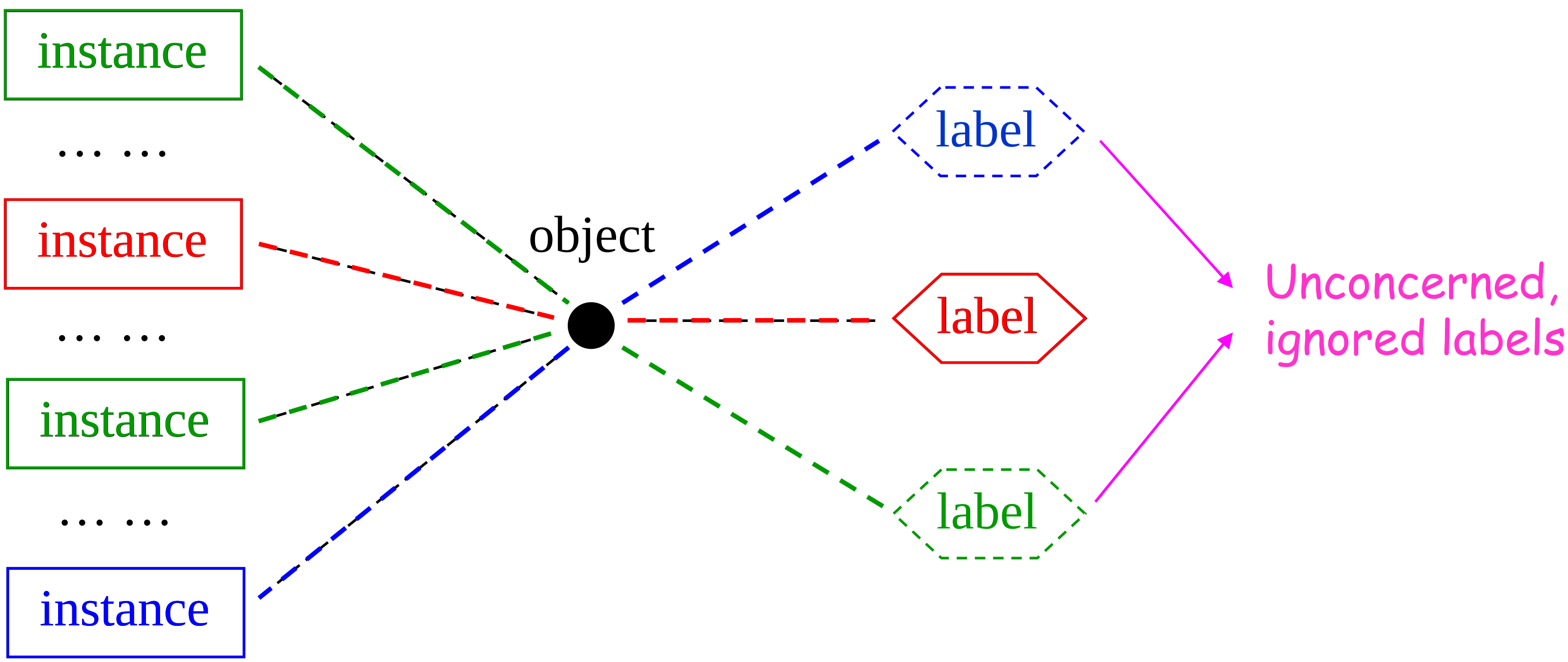
Why there are multiple labels?

***many-to-many*** mapping seems better; and moreover, MIML also offers a possibility for understanding the relationship between instances and labels



# Why MIML ? (con't)

Considering multi-instance learning, why there are multiple instances?



# Why MIML ? (con't)

Learning And Mining from DatA
http://lamda.nju.edu.cn

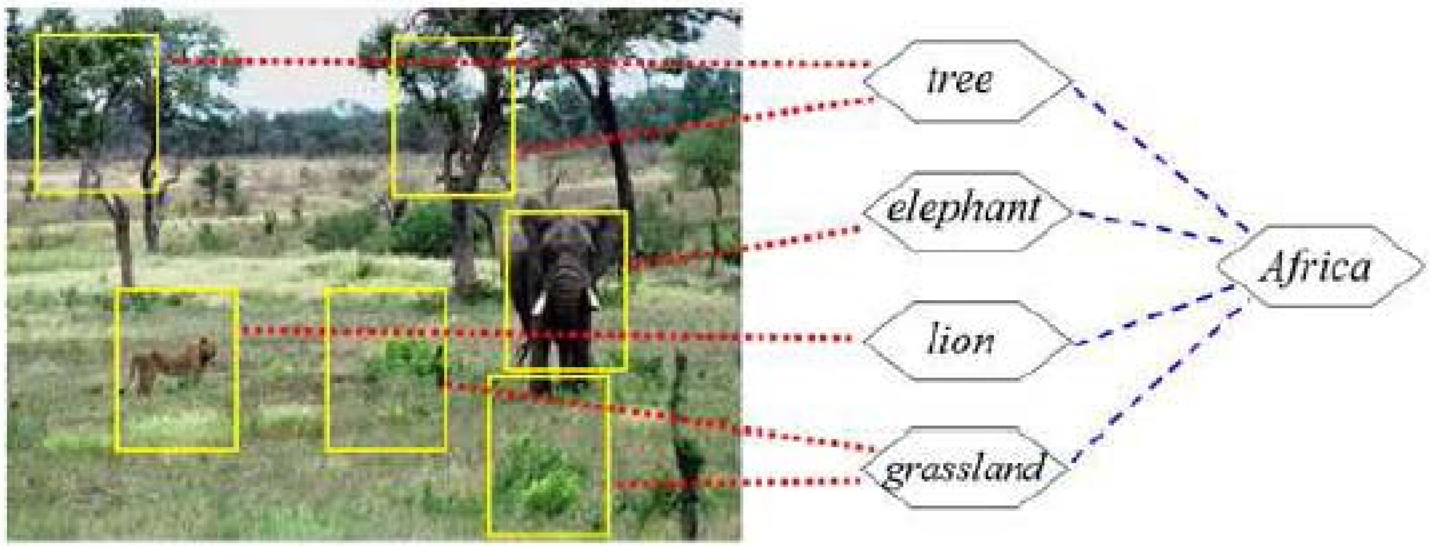
MIML can also be helpful for learning single-label examples involving complicated high-level concepts

(a) *Africa* is a complicated high-level concept

# Why MIML ? (con't)

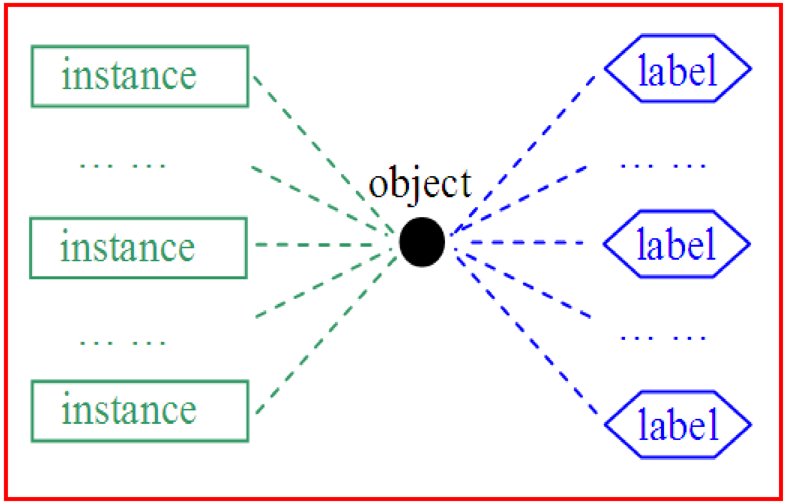


MIML can also be helpful for learning single-label examples involving complicated high-level concepts




(b) The concept *Africa* may become easier to learn through exploiting some sub-concepts

# Multi-Instance Multi-Label Learning




**MIML task**:

To learn a function $f_{MIML}: 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$ from a given data set $\{(X_1, Y_1), (X_2, Y_2), \cdots, (X_m, Y_m)\}$, where $X_i \subseteq \mathcal{X}$ is a set of instances $\{\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \cdots, \boldsymbol{x}_{n_i}^{(i)}\}$, $\boldsymbol{x}_j^{(i)} \in \mathcal{X}$ $(j = 1, 2, \cdots, n_i)$, and $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^{(i)}, y_2^{(i)}, \cdots, y_{l_i}^{(i)}\}$, $y_k^{(i)} \in \mathcal{Y}$ $(k = 1, 2, \cdots, l_i)$.

$\mathcal{X}$ - the instance space

$\mathcal{Y}$ - the set of class labels

$n_i$ - the number of instances in $X_i$

$l_i$ - the number of labels in $Y_i$

# Outline

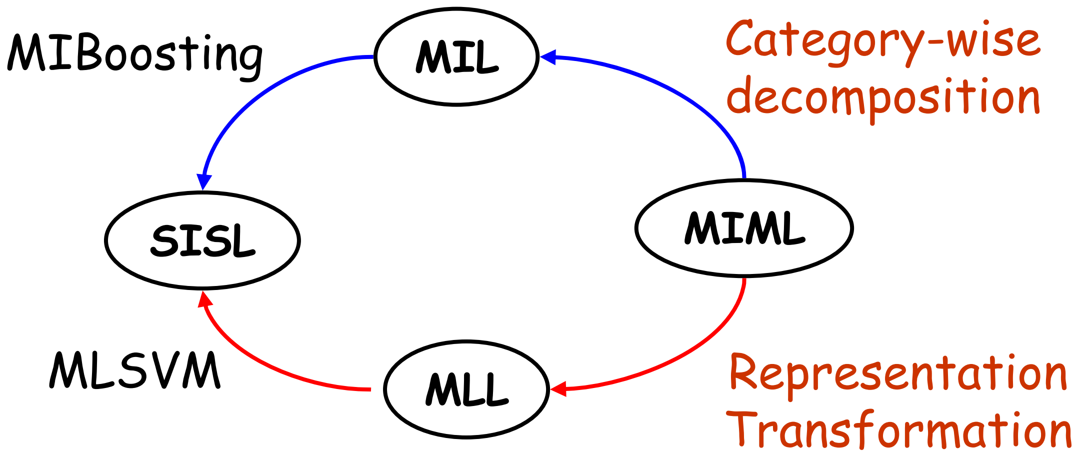


- Previous research
  - Multi-label learning
  - Multi-instance learning
- MIML: A new framework
  - Why MIML?
  - Solving MIML - by degeneration; by regularization
  - No access to raw data – how to do?
  - Usefulness in single-label problems

# MIMLBoost & MIMLSVM



MIMLBoost (an illustration of Solution 1)




MIMLSVM (an illustration of Solution 2)

# MIMLBoost

LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

Table 1: The MIMLBOOST algorithm

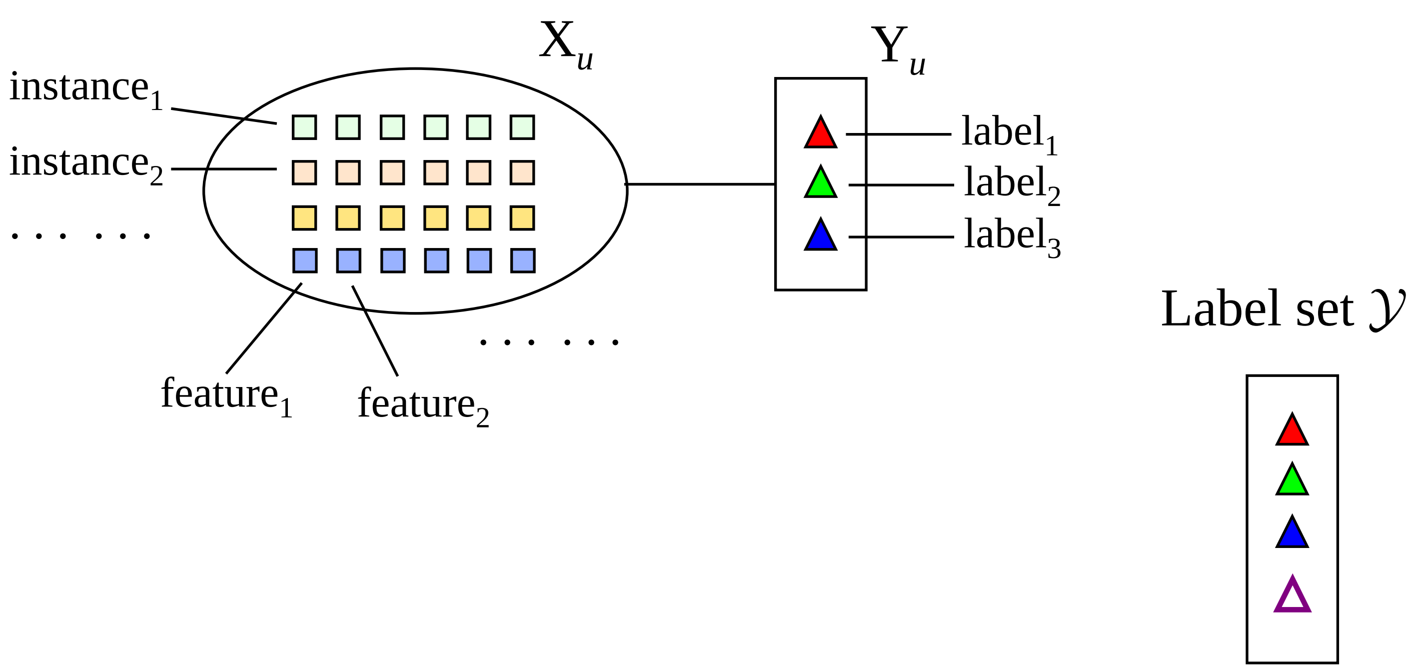
| | |
|---|---|
| 1 | Transform each MIML example $(X_u, Y_u)$ $(u = 1, 2, \cdots, m)$ into $\|\mathcal{Y}\|$ number of multi-instance bags $\{[(X_u, y_1), \Psi(X_u, y_1)], \cdots, [(X_u, y_{\|\mathcal{Y}\|}), \Psi(X_u, y_{\|\mathcal{Y}\|})]\}$. Thus, the original data set is transformed into a multi-instance data set containing $m \times \|\mathcal{Y}\|$ number of multi-instance bags, denoted by $\{[(X^{(i)}, y^{(i)}), \Psi(X^{(i)}, y^{(i)})]\}$ $(i = 1, 2, \cdots, m \times \|\mathcal{Y}\|)$. |
| 2 | Initialize weight of each bag to $W^{(i)} = \frac{1}{m \times \|\mathcal{Y}\|}$ $(i = 1, 2, \cdots, m \times \|\mathcal{Y}\|)$. |
| 3 | Repeat for $t = 1, 2, \cdots, T$ iterations: |
| 3a | Set $W_j^{(i)} = W^{(i)}/n_i$ $(i = 1, 2, \cdots, m \times \|\mathcal{Y}\|)$, assign the bag's label $\Psi(X^{(i)}, y^{(i)})$ to each of its instances $(x_j^{(i)}, y^{(i)})$ $(j = 1, 2, \cdots, n_i)$, and build an instance-level predictor $h_t[(x_j^{(i)}, y^{(i)})] \in \{-1, +1\}$. |
| 3b | For the $i$th bag, compute the error rate $e^{(i)} \in [0, 1]$ by counting the number of misclassified instances within the bag, i.e. $e^{(i)} = \frac{\sum_{j=1}^{n_i} [\![h_t[(x_j^{(i)}, y^{(i)})] \neq \Psi(X^{(i)}, y^{(i)})]\!]}{n_i}$. |
| 3c | If $e^{(i)} < 0.5$ for all $i \in \{1, 2, \cdots, m \times \|\mathcal{Y}\|\}$, go to Step 4. |
| 3d | Compute $c_t = \arg\min_{c_t} \sum_{i=1}^{m \times \|\mathcal{Y}\|} W^{(i)} \exp[(2e^{(i)} - 1)c_t]$. |
| 3e | If $c_t \leq 0$, go to Step 4. |
| 3f | Set $W^{(i)} = W^{(i)} \exp[(2e^{(i)} - 1)c_t]$ $(i = 1, 2, \cdots, m \times \|\mathcal{Y}\|)$ and re-normalize such that $0 \leq W^{(i)} \leq 1$ and $\sum_{i=1}^{m \times \|\mathcal{Y}\|} W^{(i)} = 1$. |
| 4 | Return $Y^* = \{y \mid \arg_{y \in \mathcal{Y}} sign\left(\sum_j \sum_t c_t h_t[(x_j^*, y)]\right) = +1\}$ ($x_j^*$ is $X^*$'s $j$th instance). |

# MIMLBoost (con't)



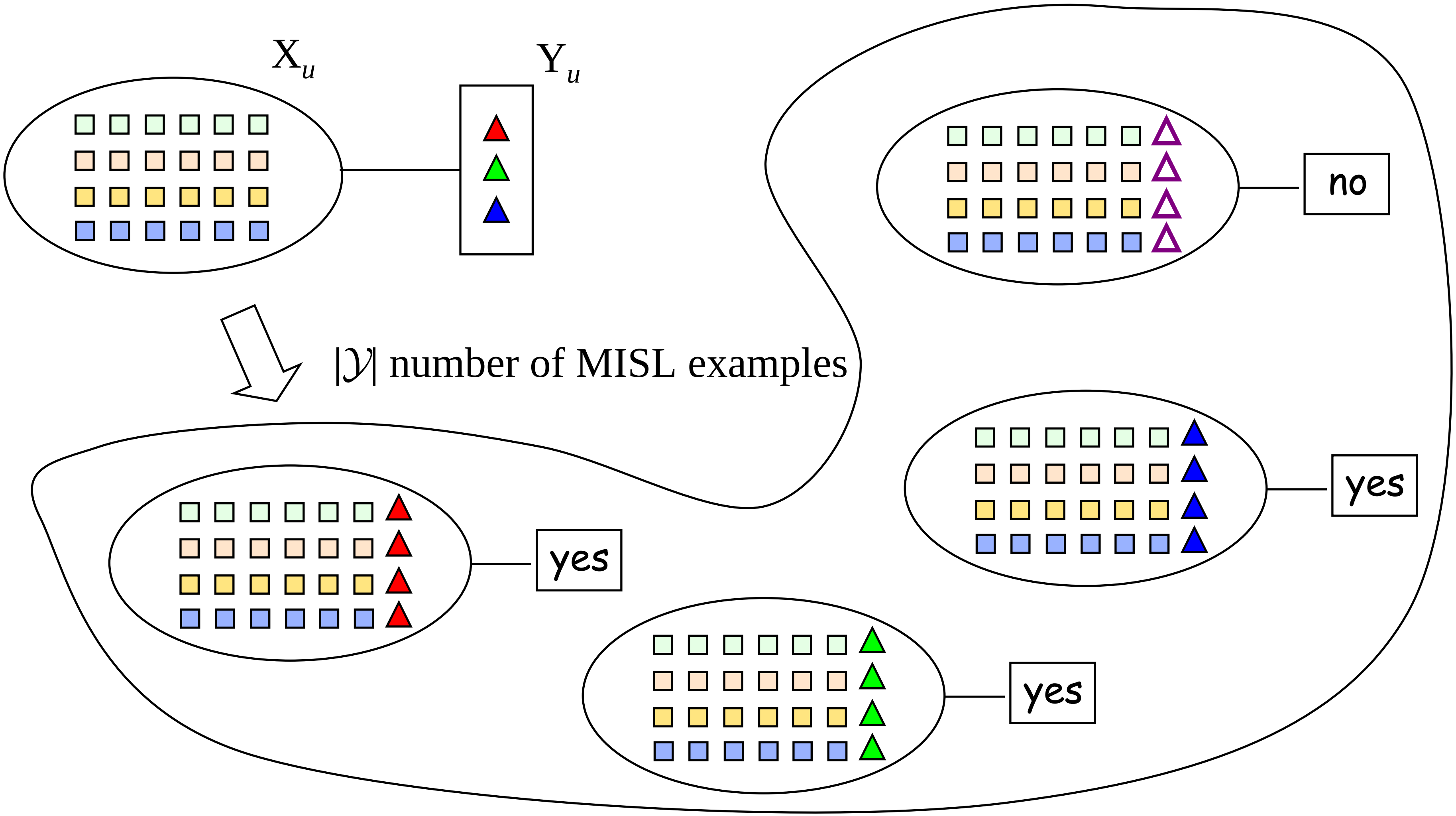
Illustration of the category-wise decomposition:

An MIML example $(X_u, Y_u)$



# MIMLBoost (con't)





# MIMLSVM

Table 2: The MIMLSVM algorithm

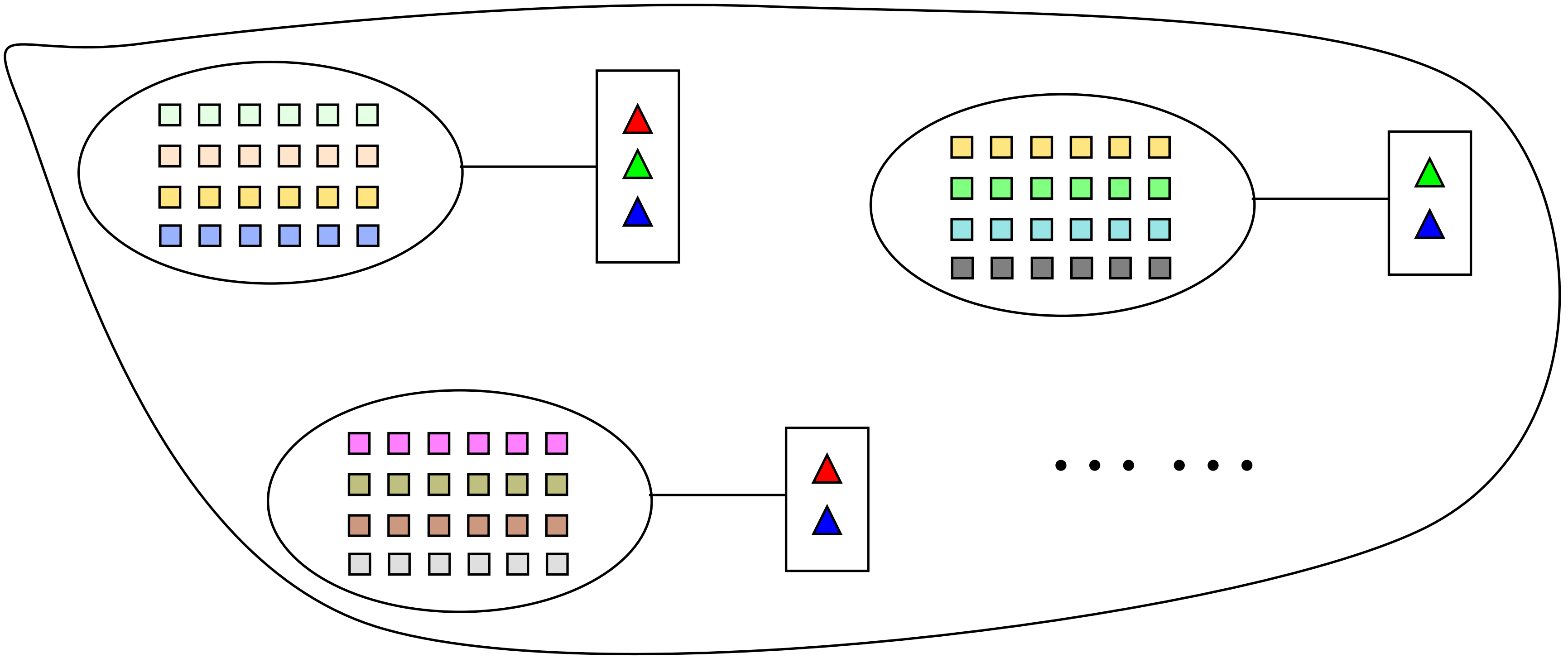
| | |
|---|---|
| 1 | For MIML examples $(X_u, Y_u)$ $(u = 1, 2, \cdots, m)$, $\Gamma = \{X_u \| u = 1, 2, \cdots, m\}$. |
| 2 | Randomly select $k$ elements from $\Gamma$ to initialize the medoids $M_t$ $(t = 1, 2, \cdots, k)$, repeat until all $M_t$ do not change: |
| | 2a $\Gamma_t = \{M_t\}$ $(t = 1, 2, \cdots, k)$. |
| | 2b Repeat for each $X_u \in (\Gamma - \{M_t \| t = 1, 2, \cdots, k\})$: $index = \arg\min_{t \in \{1, \cdots, k\}} d_H(X_u, M_t)$, $\Gamma_{index} = \Gamma_{index} \cup \{X_u\}$. |
| | 2c $M_t = \arg\min_{A \in \Gamma_t} \sum_{B \in \Gamma_t} d_H(A, B)$ $(t = 1, 2, \cdots, k)$. |
| 3 | Transform $(X_u, Y_u)$ into a multi-label example $(z_u, Y_u)$ $(u = 1, 2, \cdots, m)$, where $z_u = (z_{u1}, z_{u2}, \cdots, z_{uk}) = (d_H(X_u, M_1), d_H(X_u, M_2), \cdots, d_H(X_u, M_k))$. |
| 4 | For each $y \in \mathcal{Y}$, derive a data set $\mathcal{D}_y = \{(z_u, \Phi(z_u, y)) \| u = 1, 2, \cdots, m\}$, and then train an SVM $h_y = SVMTrain(\mathcal{D}_y)$. |
| 5 | Return $Y^* = \{\arg\max_{y \in \mathcal{Y}} h_y(z^*)\} \cup \{y \| h_y(z^*) \geq 0, y \in \mathcal{Y}\}$, where $z^* = (d_H(X^*, M_1), d_H(X^*, M_2), \cdots, d_H(X^*, M_k))$. |

# MIMLSVM (con't)

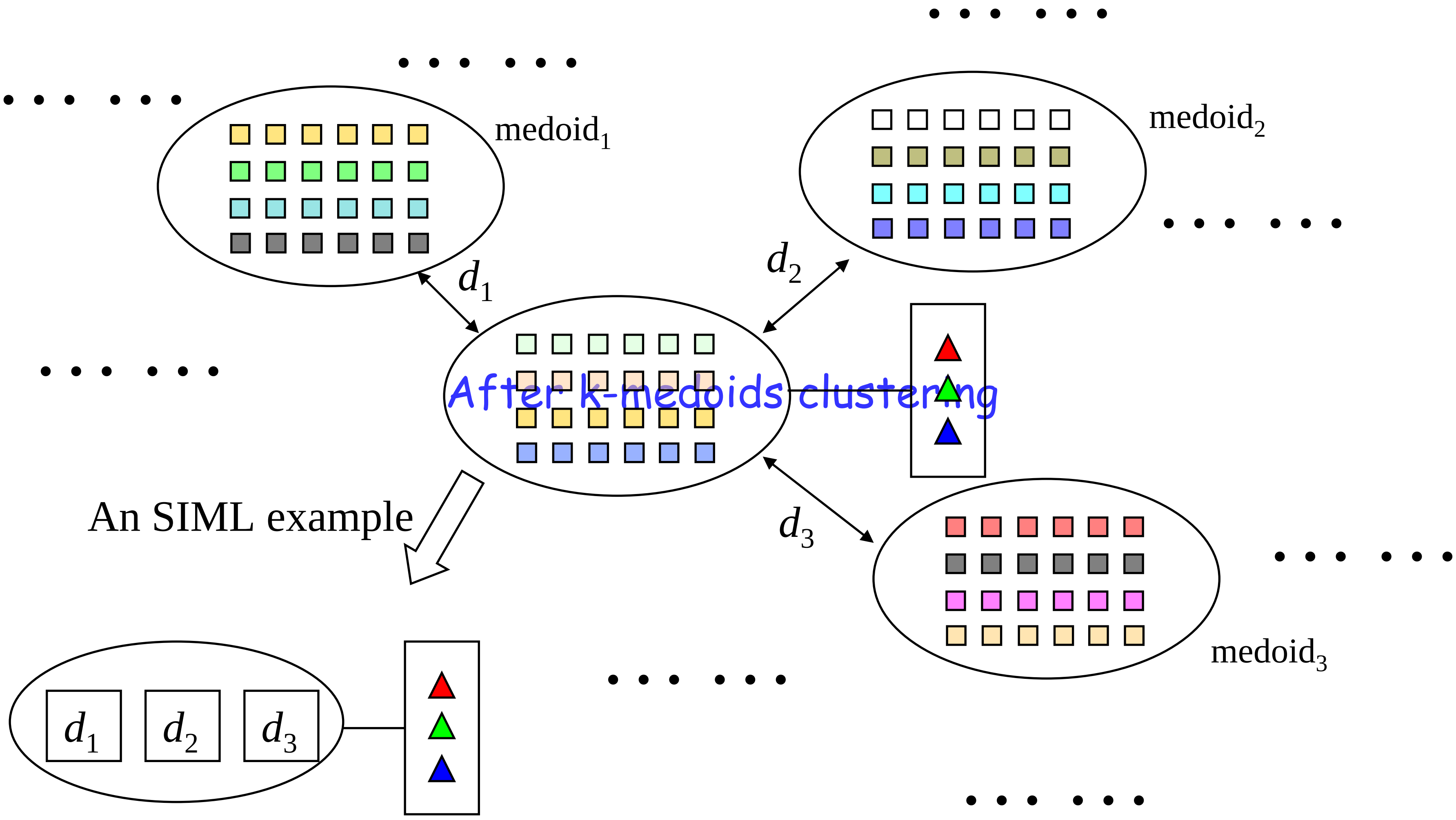
Learning And Mining from DatA
http://lamda.nju.edu.cn

Illustration of the representation transformation:

A set of MIML examples

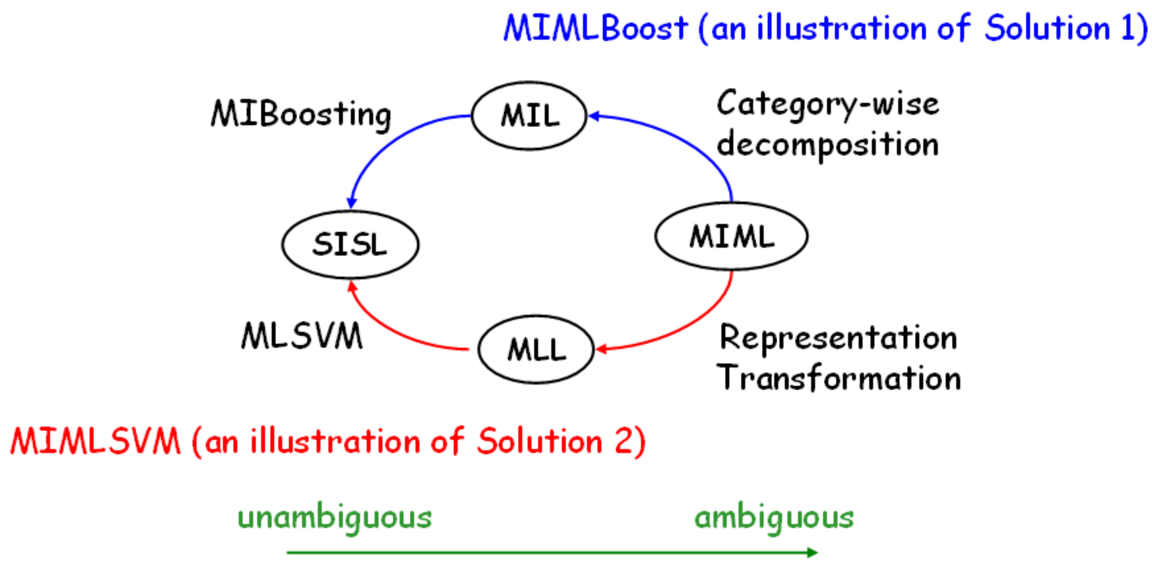


# MIMLSVM (con't)



# Again, Why MIML?






- The MIML framework incorporates more information (+)
- These solutions degenerate MIML to solve, while the degeneration loses information (-)

**If (+) > (-), then it is worth doing**

# Scene Classification: Result



Table 3
Results (mean±std.) on scene classification ('↓' indicates 'the smaller the better'; '↑' indicates 'the larger the better')

| Compared Algorithms | Evaluation Criteria | | | | |
|---|---|---|---|---|---|
| | *hloss* ↓ | *one-error* ↓ | *coverage* ↓ | *rloss* ↓ | *aveprec* ↑ |
| MimlBoost | .192±.004 | **.349±.016** | **.986±.041** | **.179±.008** | **.778±.009** |
| MimlSvm | **.190±.009** | .350±.020 | 1.083±.050 | .201±.001 | .766±.013 |
| AdtBoost.MH | .210±.006 | .436±.019 | 1.223±.049 | N/A | .718±.012 |
| RankSvm | .219±.020 | .400±.062 | 1.177±.160 | .225±.041 | .739±.040 |
| Ml-*k*NN | .191±.006 | .370±.017 | 1.085±.047 | .203±.010 | .759±.010 |

↓: the smaller, the better ↑: the larger, the better

# Text Categorization: Result



Table 4
Results (mean±std.) on text categorization ('↓' indicates 'the smaller the better'; '↑' indicates 'the larger the better')

| Compared Algorithms | Evaluation Criteria | | | | |
|---|---|---|---|---|---|
| | *hloss* ↓ | *one-error* ↓ | *coverage* ↓ | *rloss* ↓ | *aveprec* ↑ |
| MimlBoost | .054±.004 | .092±.013 | .401±.035 | .037±.004 | .937±.007 |
| MimlSvm | **.034±.003** | **.071±.009** | **.315±.029** | **.024±.003** | **.955±.006** |
| AdtBoost.MH | .055±.004 | .120±.016 | .409±.046 | N/A | .925±.010 |
| RankSvm | .093±.007 | .205±.055 | .639±.161 | .078±.027 | .867±.037 |
| Ml-*k*NN | .067±.005 | .191±.017 | .683±.052 | .085±.008 | .871±.010 |

↓: the smaller, the better ↑: the larger, the better

LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

# Outline

- Previous research
  - Multi-label learning
  - Multi-instance learning
- MIML: A new framework
  - Why MIML?
  - Solving MIML - by degeneration; by regularization
  - No access to raw data – how to do?
  - Usefulness in single-label problems

# The Loss Function



$$V(\{X_i\}_{i=1}^m, \{Y_i\}_{i=1}^m) = \frac{1}{mT} \sum_{i=1}^{m}\sum_{t=1}^{T} (1 - y_{it} f_t(X_i))_+ + \frac{\lambda}{mT} \sum_{i=1}^{m}\sum_{t=1}^{T} l\left(f_t(X_i), \max_{j=1,\cdots,n_i} f_t(x_{ij})\right)$$

The loss between the bag $X_i$'s labels and its corresponding predictions $\boldsymbol{f}(X_i)$, $\boldsymbol{f} = (f_1, f_2, \cdots, f_T)$

The loss between $\boldsymbol{f}(X_i)$ and the predictions of $X_i$'s constituent instances $\{\boldsymbol{f}(x_{ij})\}$

# The Representer Theorem



Assume $f_t(x) = \langle \boldsymbol{w}_t, \phi(\boldsymbol{x}) \rangle$ and all the $w_t$'s come from a particular Gaussian distribution, with $\boldsymbol{w}_0 = \sum_{t=1}^{T} \boldsymbol{w}_t / T$

We want to minimize $\sum_{i=1}^{T} ||\boldsymbol{w}_i||^2$ and $||\boldsymbol{w}_0||^2$ simultaneously, and thus we have

$$\min_{\boldsymbol{f} \in \mathcal{H}} \frac{1}{2T} \sum_{i=1}^{T} ||\boldsymbol{w}_i||^2 + \mu ||\boldsymbol{w}_0||^2 + \gamma V\left(\{X_i\}_{i=1}^{m}, \{Y_i\}_{i=1}^{m}, \boldsymbol{f}\right)$$

**Theorem 1** *The minimizer of the optimization problem Eq. 7 admits an expansion*

$$f_t(\boldsymbol{x}) = \sum_{i=1}^{m} \left( \alpha_{t,i0} k\left(\boldsymbol{x}, X_i\right) + \sum_{j=1}^{n_i} \alpha_{t,ij} k(\boldsymbol{x}, \boldsymbol{x}_{ij}) \right)$$

*where all* $\alpha_{t,i0}, \alpha_{t,ij} \in \mathcal{R}$.

# The Optimization Problem



Assume the bags and instances are ordered as

$$(X_1, \cdots, X_m,\ x_{11}, \cdots, x_{1,n_1}, \cdots, x_{m,1}, \cdots, x_{m,n_m})$$

Thus each object (bags or instances) can be indexed by

$$\begin{cases} \mathcal{I}(X_i) = i \\ \mathcal{I}(x_{ij}) = m + \sum_{l=1}^{i-1} n_l + j \end{cases}$$

We can obtain the $(m+n) \times (m+n)$ kernel matrix $K$ with the $i$-th column denoted by $k_i$. We have

$$f_t(X_i) = k'_{\mathcal{I}(X_i)} \alpha_t + b_t \qquad f_t(x_{ij}) = k'_{\mathcal{I}(x_{ij})} \alpha_t + b_t.$$

# The Optimization Problem (con't)



$$\min_{\boldsymbol{\alpha}_t,\boldsymbol{\xi},\boldsymbol{\delta},b} \frac{1}{2T}\sum_{t=1}^{T}\boldsymbol{\alpha}_t'\boldsymbol{K}\boldsymbol{\alpha}_t + \frac{\mu}{T^2}\mathbf{1}'\boldsymbol{A}'\boldsymbol{K}\boldsymbol{A}\mathbf{1} + \frac{\gamma}{mT}\boldsymbol{\xi}'\mathbf{1} + \frac{\gamma\lambda}{mT}\boldsymbol{\delta}'\mathbf{1}$$

$$\text{s.t. } y_{it}(\boldsymbol{k}_{\mathcal{I}(X_i)}'\boldsymbol{\alpha}_t + b_t) \geq 1 - \xi_{it},$$

$$\boldsymbol{\xi} \geq \mathbf{0},$$

$$\boldsymbol{k}_{\mathcal{I}(\boldsymbol{x}_{ij})}'\boldsymbol{\alpha}_t - \delta_{it} \leq \boldsymbol{k}_{\mathcal{I}(X_i)}'\boldsymbol{\alpha}_t,$$

$$\boldsymbol{k}_{\mathcal{I}(X_i)}'\boldsymbol{\alpha}_t - \max_{j=1,\cdots,n_i} \boldsymbol{k}_{\mathcal{I}(\boldsymbol{x}_{ij})}'\boldsymbol{\alpha}_t \leq \delta_{it},$$

where $\boldsymbol{\xi} = [\xi_{11}, \xi_{12}, \cdots, \xi_{it}, \cdots, \xi_{mT}]'$ are slack variables for the errors on the training bags for each label, $\boldsymbol{\delta} = [\delta_{11}, \delta_{12}, \cdots, \delta_{it}, \cdots, \delta_{mT}]'$ , $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_T]$. $\mathbf{0}$ and $\mathbf{1}$ are vectors of 0's and 1's respectively.

This can be solved by CCCP (concave-convex procedure)

# Considering the Class-Imbalance



Considering the class-imbalance, i.e., for any class label the number of positive instances is much fewer than the number of negative instances in MIML problems

We roughly estimate the imbalance rate for label $y$ according to

$$ibr(y) = \sum_{\substack{i=1 \\ y \in Y_i}}^{m} \frac{n_i}{|Y_i|} \times \frac{1}{\sum_{i=1}^{m} n_i} = \sum_{\substack{i=1 \\ y \in Y_i}}^{m} \frac{n_i}{n \times |Y_i|}$$

According to the *rescaling* method, we can re-write the hinge loss function into

$$\left(\frac{y_{it}+1}{2} - y_{it} \times ibr(y_{it})\right)(1 - y_{it} f_t(X_i))$$

# Considering the Class-Imbalance (con't)



Let $\boldsymbol{\tau} = [\tau_{11}, \tau_{12}, \cdots, \tau_{it}, \cdots, \tau_{mT}]$, where $\tau_{it} = \left(\frac{y_{it}+1}{2} - y_{it} \times ibr(y_{it})\right)$

We have

$$\begin{aligned}
\min_{\boldsymbol{\alpha}_t, \boldsymbol{\xi}, \boldsymbol{\delta}, b} \ & \frac{1}{2T}\sum_{t=1}^{T} \boldsymbol{\alpha}_t' \boldsymbol{K} \boldsymbol{\alpha}_t + \frac{\mu}{T^2} \mathbf{1}' \boldsymbol{A}' \boldsymbol{K} \boldsymbol{A} \mathbf{1} + \frac{\gamma}{mT} \boldsymbol{\xi}' \boldsymbol{\tau} + \frac{\gamma\lambda}{mT} \boldsymbol{\delta}' \mathbf{1} \\
\text{s.t.} \ & y_{it}(\boldsymbol{k}_{\mathcal{I}(X_i)}' \boldsymbol{\alpha}_t + b_t) \geq 1 - \xi_{it}, \\
& \boldsymbol{\xi} \geq \mathbf{0}, \\
& \boldsymbol{k}_{\mathcal{I}(\boldsymbol{x}_{ij})}' \boldsymbol{\alpha}_t - \delta_{it} \leq \boldsymbol{k}_{\mathcal{I}(X_i)}' \boldsymbol{\alpha}_t, \\
& \boldsymbol{k}_{\mathcal{I}(X_i)}' \boldsymbol{\alpha}_t - \sum_{j=1}^{n_i} \rho_{ijt} \boldsymbol{k}_{\mathcal{I}(\boldsymbol{x}_{ij})}' \boldsymbol{\alpha}_t \leq \delta_{it}.
\end{aligned}$$

This is still a standard QP problem; but, large-scale …

# An Efficient Cutting-Plane Algorithm

Efficient Algorithm for Eq. 15

---

**Input:** $K$, $\lambda$, $\mu$, $\gamma$, $\varepsilon$, $\{X_i, Y_i\}_{i=1}^m$

$\forall t$, $S_t = \emptyset$, $\boldsymbol{v}_t = (\boldsymbol{\alpha}_t^T, \xi_{t1}, \cdots, \xi_{tm}, \boldsymbol{\delta}_{t1}, \cdots, \boldsymbol{\delta}_{tm}, b_t) = \mathbf{0}$
**Repeat**
  **For** $t = 1, \cdots, T$
    Pick $p$ indexes of constraints that are not in $S_t$ randomly, denoted by $I$;
    Compute $Loss_i$ for every constraint in $I$;
    % find out the cutting plane
    $q = \arg\max_{i \in I} Loss_i$
    **If** $Loss_q > \varepsilon$
      $S_t = S_t \cup \{q\}$;
      $\boldsymbol{v}_t \leftarrow$ optimized over $S_t$;
    **End If**
  **End For**
**Until** no $S_t$ changes

---

# Compare D-MIMLSVM with MIMLSVM



Table 6
Comparing results (mean±std.) of D-MimlSvm and MimlSvm ('↓' indicates 'the smaller the better'; '↑' indicates 'the larger the better')

| Data | Compared Algorithms | Evaluation Criteria | | | | |
|---|---|---|---|---|---|---|
| | | *hloss* ↓ | *one-error* ↓ | *coverage* ↓ | *rloss* ↓ | *aveprec* ↑ |
| *Scene* | D-MimlSvm | .222±.003 | **.397±.013** | **1.154±.029** | **.221±.006** | **.742±.007** |
| | MimlSvm | **.216±.005** | .423±.014 | 1.211±.033 | .234±.007 | .724±.008 |
| *Text* | D-MimlSvm | **.041±.003** | **.091±.011** | **.354±.030** | **.030±.005** | **.943±.007** |
| | MimlSvm | .045±.003 | .102±.008 | .402±.027 | .038±.004 | .933±.005 |

↓: the smaller, the better ↑: the larger, the better

# Compare D-MIMLSVM with MIMLSVM (con't)

Learning And Mining from DatA
http://lamda.nju.edu.cn

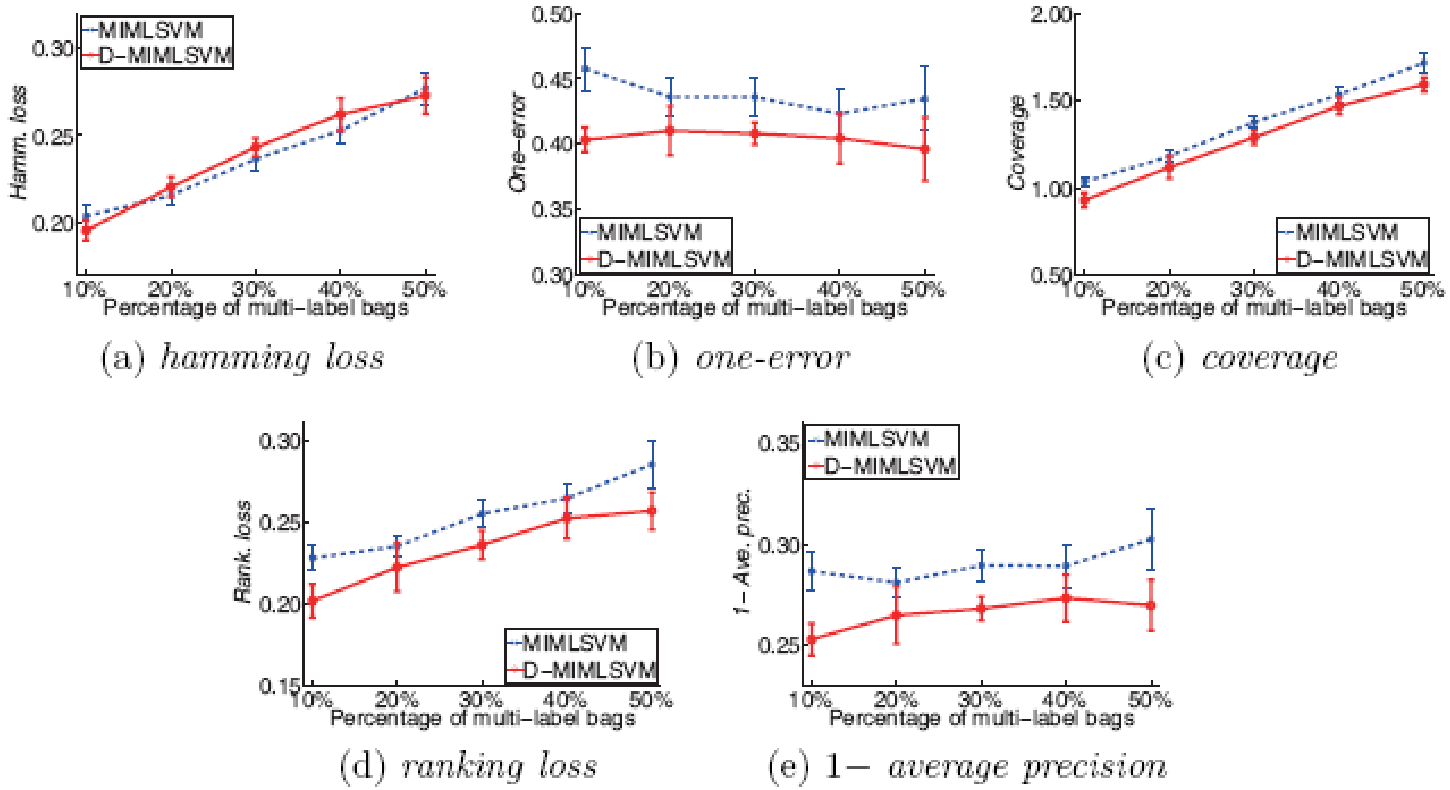
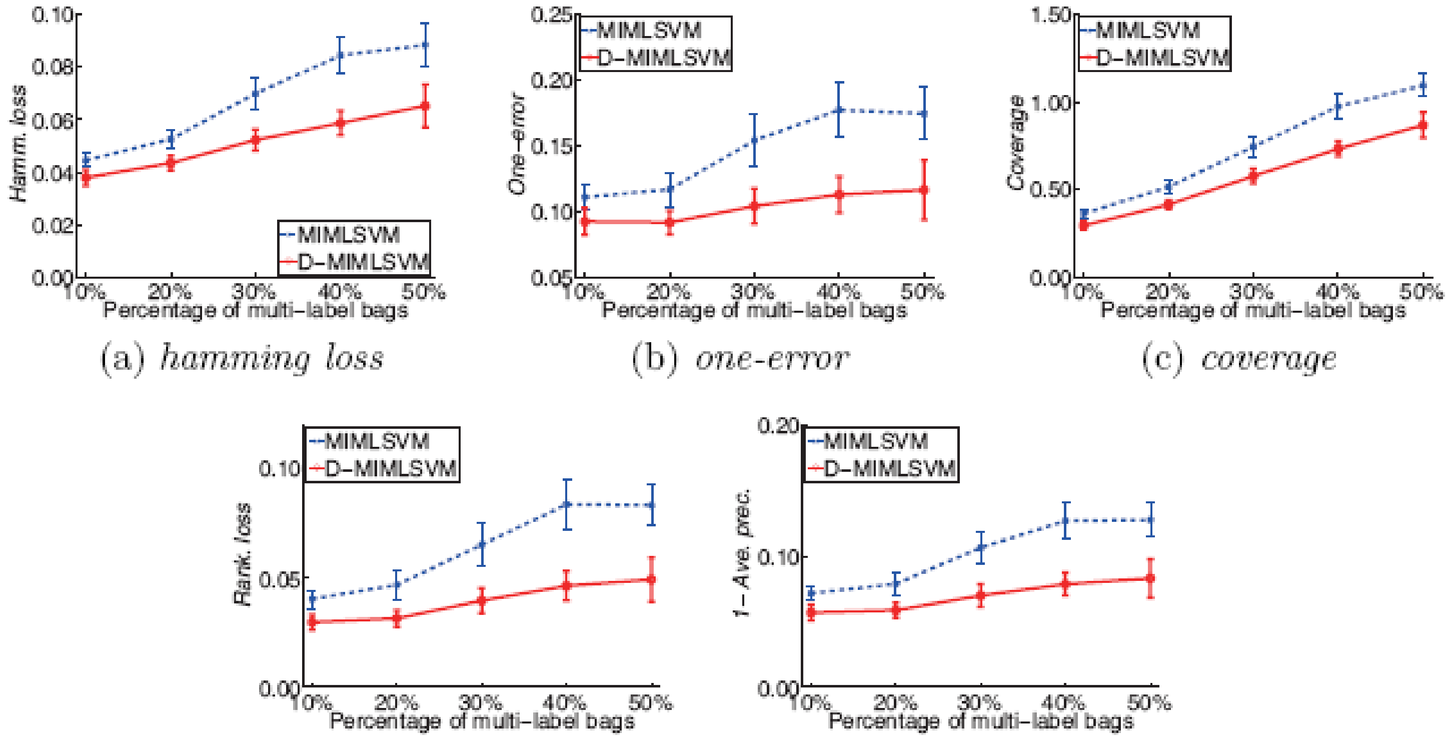



Fig. 4. Results on scene classification with different percentage of multi-label data. The lower the curve, the better the performance.

# Compare D-MIMLSVM with MIMLSVM (con't)




(a) *hamming loss* (b) *one-error* (c) *coverage*

(d) *ranking loss* (e) 1− *average precision*

Fig. 5. Results on text categorization with different percentage of multi-label data. The lower the curve, the better the performance.

# Outline



- Previous research
  - Multi-label learning
  - Multi-instance learning
- MIML: A new framework
  - Why MIML?
  - Solving MIML - by degeneration; by regularization
  - No access to raw data – how to do?
  - Usefulness in single-label problems

# A Question



If I cannot get touch with the raw data, instead, I can only access the processed data, can I make use of MIML?

Yes !

Now assume that we are given with the data set $\{(x_1, Y_1), (x_2, Y_2), \cdots, (x_m, Y_m)\}$ for which we could not access the real object, and the feature extraction task has been done by others

The InsDif approach

The key: How to transform a "single-instance" to "multi-instance"?

# Instance Differentiation



Generate a *prototype* for each class label:

$$\mathbf{v}_l = \left(\sum_{\mathbf{x}_i \in U_l} \mathbf{x}_i\right) / |U_l|, \text{ where}$$
$$U_l = \{\mathbf{x}_i | \{\mathbf{x}_i, Y_i\} \in S,\ l \in Y_i\},\ l \in \mathcal{Y}$$

For example:

$([0.1, 0.2, 0.3]^T, \{l_1, l_2\}) \rightarrow [0.1, 0.2, 0.3]^T$

$([0.2, 0.4, 0.6]^T, \{l_2\}) \rightarrow [0.2, 0.4, 0.6]^T$

$([0.3, 0.6, 0.9]^T, \{l_1, l_3\}) \rightarrow [0.3, 0.6, 0.9]^T$

$([0.4, 0.2, 0.6]^T, \{l_1, l_2, l_3\}) \rightarrow [0.4, 0.2, 0.6]^T$

$$\frac{[0.8, 1.0, 1.8]^T}{3} \quad \frac{[0.7, 0.8, 1.5]^T}{2}$$

$v_1 = [0.27, 0.33, 0.6]^T$

$v_2 = [0.23, 0.27, 0.5]^T$

$v_3 = [0.35, 0.4, 0.75]^T$

Each prototype can be approximately regarded as a profile of the concerned class

# Instance Differentiation (con't)



Transforming a single instance $x_i$ into a bag of instances:

$$B_i = \{\mathbf{x}_i - \mathbf{v}_l | l \in \mathcal{Y}\}$$

For example:

([0.1, 0.2, 0.3]$^T$, $\{l_1, l_2\}$)
([0.2, 0.4, 0.6]$^T$, $\{l_2\}$)
([0.3, 0.6, 0.9]$^T$, $\{l_1, l_3\}$)
([0.4, 0.2, 0.6]$^T$, $\{l_1, l_2, l_3\}$)

$v_1$ = [0.27, 0.33, 0.6]$^T$
$v_2$ = [0.23, 0.27, 0.5]$^T$
$v_3$ = [0.35, 0.4, 0.75]$^T$

⇩

({[-0.17, -0.13, -0.3]$^T$, [-0.13, -0.07, -0.2]$^T$, [-0.25, -0.2, -0.45]$^T$}, $\{l_1, l_2\}$)
({[-0.07, 0.07, 0]$^T$, [-0.03, 0.13, 0.1]$^T$, [-0.15, 0, -0.15]$^T$}, $\{l_2\}$)
({[0.03, 0.27, 0.3]$^T$, [0.07, 0.33, 0.4]$^T$, [-0.05, 0.2, 0.15]$^T$}, $\{l_1, l_3\}$)
({[0.13, -0.13, 0]$^T$, [0.17, -0.07, 0.1]$^T$, [0.05, -0.2, -0.15]$^T$}, $\{l_1, l_2, l_3\}$)

# Instance Differentiation (con't)

The transformation expresses some structural/distributional information of the instances and the classes

After transformation, the task becomes:

To learn from a data set $S^{new} = \{(B_1, Y_1), (B_2, Y_2), \ldots, (B_N, Y_N)\}$

which is addressed by a two-level classification structure

a new MIML algorithm

# Two-Level Classification Structure

## 1st level:

Performing k-medoids clustering on bag $B$'s using Hausdorff distance, obtaining the medoids $C_j$ ($j$=1,…, $M$):

$$C_j = \arg\min_{A \in G_j} \sum_{B \in G_j} \mathrm{H}(A, B)$$

$M$ is a parameter of InsDif

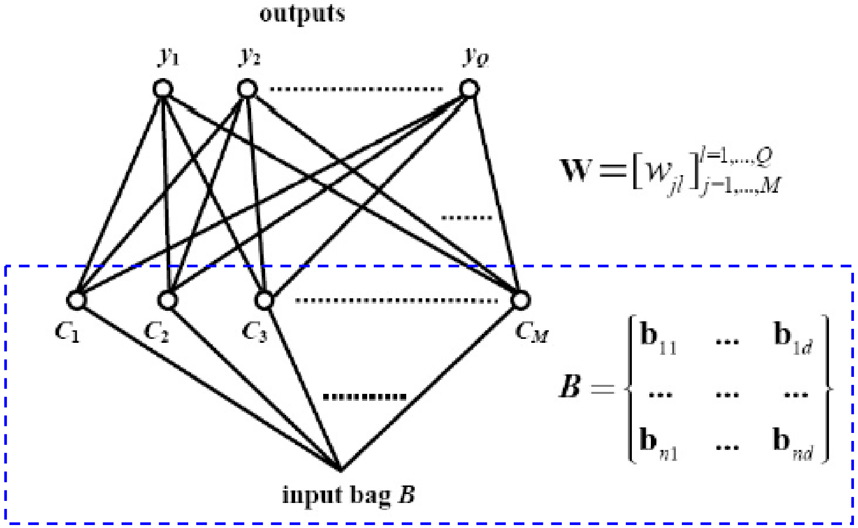



Figure 1: Two-level classification structure used by INSDIF

Then, for $B$ we can obtain $[\phi_1(B), \phi_2(B), ..., \phi_M(B)]^{\mathrm{T}}$

where $\phi_j(B) = \mathrm{H}(B, C_j)$.

Hausdorff distance between $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_{n_1}\}$ and $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_{n_2}\}$:

$$\mathrm{H}(A, B) = \max\{\max_{\mathbf{a} \in A} \min_{\mathbf{b} \in B} \|\mathbf{a} - \mathbf{b}\|, \max_{\mathbf{b} \in B} \min_{\mathbf{a} \in A} \|\mathbf{b} - \mathbf{a}\|\}$$

# Two-Level Classification Structure (con't)



## 2nd level:

Optimizing $\mathbf{W} = [w_{jl}]_{M \times Q}$ by minimizing

$$E = \frac{1}{2} \sum_{i=1}^{N} \sum_{l=1}^{Q} \{y_l(B_i) - d_l^i\}^2$$

where

$y_l(B_i) = \sum_{j=1}^{M} w_{jl} \phi_j(B_i)$ is the actual output of $B_i$ on the $l$-th class

$d_l^i$ is the desired output of $B_i$ on the $l$-th class

$d_l^i = +1$ if $l \in Y_i$ and -1 otherwise

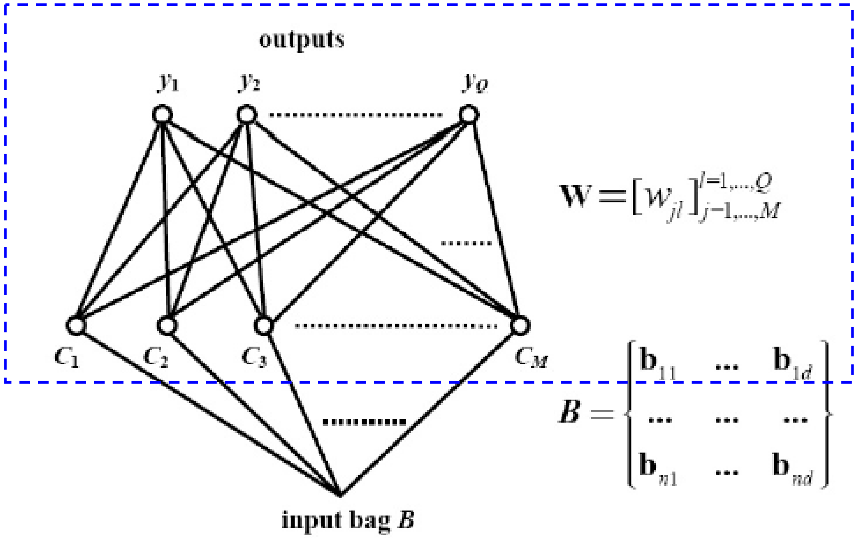



Figure 1: Two-level classification structure used by INSDIF

# Two-Level Classification Structure (con't)

## 2nd level (con't):

Differentiating the objective function

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{l=1}^{Q}\{y_l(B_i) - d_l^i\}^2$$

$$y_l(B_i) = \sum_{j=1}^{M} w_{jl}\phi_j(B_i)$$

with respect to $w_{jl}$ and setting the derivative to zero gives:

$$(\Phi^{\mathrm{T}}\Phi)\mathbf{W} = \Phi^{\mathrm{T}}\mathbf{T}$$

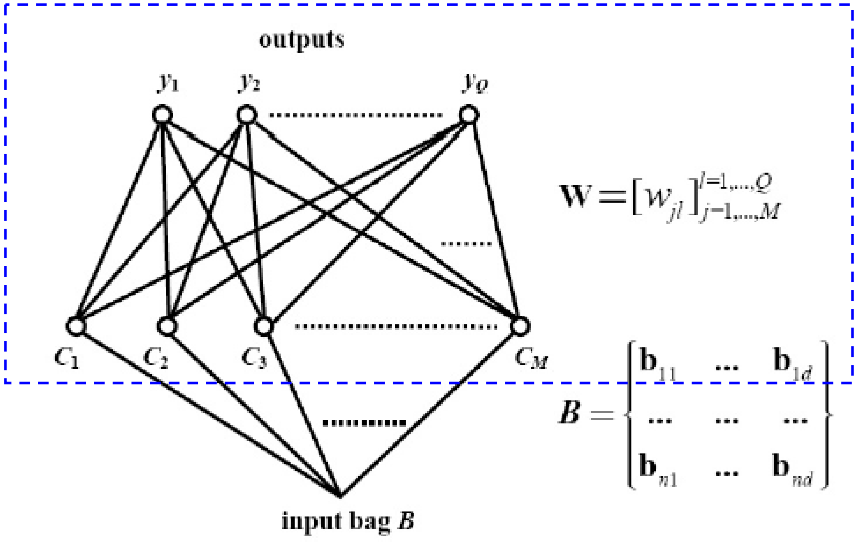



Figure 1: Two-level classification structure used by INSDIF

where $\Phi = [\phi_{ij}]_{N\times M}$ is with elements $\phi_{ij} = \phi_j(B_i)$

$\mathbf{T} = [t_{il}]_{N\times Q}$ is with elements $t_{il} = d_l^i$

$\mathbf{W}$ can be solved by singular value decomposition

# The INSDIF Algorithm



The InsDif algorithm

instance differentiation

1 For single-instance multi-label examples $(\boldsymbol{x}_u, Y_u)$ $(u = 1, 2, \cdots, m)$, compute the prototype vectors $\boldsymbol{v}_l$ $(l \in \mathcal{Y})$ using Eq. 16.

2 Derive the new training set $S^*$ by transforming each $\boldsymbol{x}_i$ into a bag of instances $B_i$ using Eq. 17.

two level classification structure

3 Divide $\{B_1, B_2, \cdots, B_m\}$ into $M$ partitions using $k$-medoids algorithm employing Hausdorff distance.

1st level

4 Determine the medoids $C_j$ $(j = 1, 2, \cdots, M)$ using Eq. 18.

2nd level

5 Compute the weights $\mathbf{W}$ by solving Eq. 20 using singular value decomposition.

6 Return $Y^* = \{l | y_l(B^*) = \sum_{j=1}^{M} w_{jl}\phi_j(B^*) > 0,\ l \in \mathcal{Y}\}$, where $B^* = \{\boldsymbol{x}^* - \boldsymbol{v}_l | l \in \mathcal{Y}\}$.

prediction

# Yeast Gene Functional Analysis: Result

Learning And Mining from DatA
http://lamda.nju.edu.cn

Table 8
Results (mean±std.) on yeast gene data set ('↓' indicates 'the smaller the better'; '↑' indicates 'the larger the better')

| Compared Algorithms | Evaluation Criteria | | | | |
|---|---|---|---|---|---|
| | *hloss* ↓ | *one-error* ↓ | *coverage* ↓ | *rloss* ↓ | *aveprec* ↑ |
| InsDif | **.189±.010** | **.214±.030** | 6.288±0.240 | **.163±.017** | **.774±.019** |
| AdtBoost.MH | .207±.010 | .244±.035 | 6.390±0.203 | N/A | .744±.025 |
| RankSvm | .207±.013 | .243±.039 | 7.090±0.503 | .195±.021 | .749±.026 |
| Ml-$k$nn | .194±.010 | .230±.030 | **6.275±0.240** | .167±.016 | .765±.021 |
| Cnmf | N/A | .354±.184 | 7.930±1.089 | .268±.062 | .668±.093 |

↓: the smaller, the better ↑: the larger, the better

Learning And Mining from DatA
http://lamda.nju.edu.cn

# Web Page Categorization: Result

Table 10
Results (mean±std.) on eleven web page categorization data sets ('↓' indicates 'the smaller the better'; '↑' indicates 'the larger the better')

| Compared Algorithms | Evaluation Criteria | | | | |
|---|---|---|---|---|---|
| | *hloss* ↓ | *one-error* ↓ | *coverage* ↓ | *rloss* ↓ | *aveprec* ↑ |
| InsDif | **.039±.013** | **.381±.118** | 4.545±1.285 | **.102±.037** | **.686±.091** |
| AdtBoost.MH | .043±.013 | .461±.137 | **4.083±1.191** | N/A | .632±.105 |
| RankSvm | .043±.014 | .440±.143 | 7.508±2.396 | .193±.065 | .605±.117 |
| Ml-$k$nn | .043±.014 | .471±.157 | 4.097±1.236 | .102±.045 | .625±.116 |
| Cnmf | N/A | .509±.142 | 6.717±1.588 | .171±.058 | .561±.114 |

↓: the smaller, the better ↑: the larger, the better

# Outline

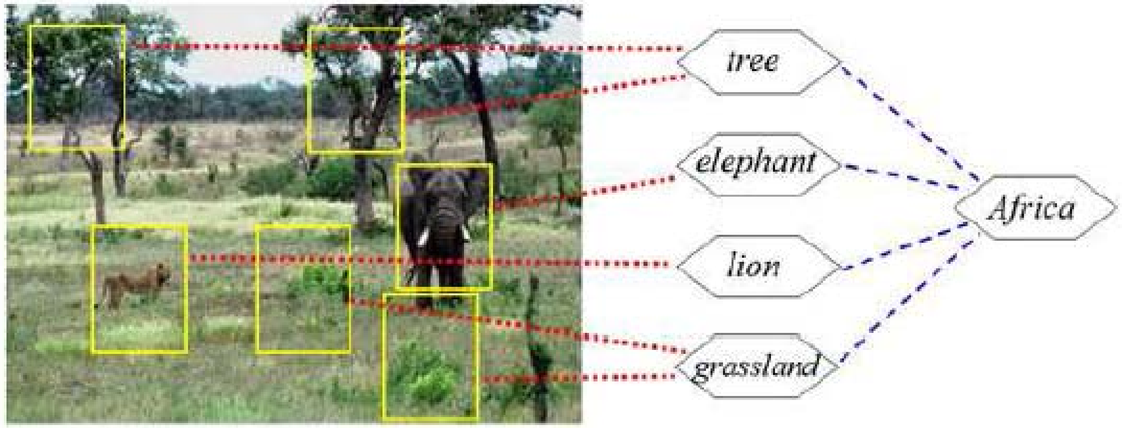


- Previous research
  - Multi-label learning
  - Multi-instance learning
- MIML: A new framework
  - Why MIML?
  - Solving MIML - by degeneration; by regularization
  - No access to raw data – how to do?
  - Usefulness in single-label problems

# Remind …

MIML can be helpful for learning single-label examples involving complicated high-level concepts

(a) *Africa* is a complicated high-level concept




(b) The concept *Africa* may become easier to learn through exploiting some sub-concepts

# Sub-Concept Discovery



We collect all instances from all bags to compose data set $D = \{x_{11}, \cdots, x_{1,n_1}, x_{21}, \cdots, x_{2,n_2}, \cdots, x_{m1}, \cdots, x_{m,n_m}\}$, and re-index it as $\{x_1, x_2, \cdots, x_N\}$

A Gaussian mixture model with $M$ mixture components is to be learned from $D$ by the EM algorithm, and the mixture components are regarded as *sub-concepts*

# Sub-Concept Discovery (con't)



E-step

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{M} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \Sigma_j)} \qquad (i = 1, 2, \cdots, N)$$

M-step

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{i=1}^{N} \gamma_{ik} \boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_{ik}} \qquad \Sigma_k^{new} = \frac{\sum_{i=1}^{N} \gamma_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})^{\mathrm{T}}}{\sum_{i=1}^{N} \gamma_{ik}} \qquad \pi_k^{new} = \frac{\sum_{i=1}^{N} \gamma_{ik}}{N}$$

The log-likelihood is evaluated according to

$$\ln p(D|\boldsymbol{\mu}, \Sigma, \pi) = \sum_{i=1}^{N} \ln \left( \sum_{k=1}^{M} \pi_k^{new} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k^{new}, \Sigma_k^{new}) \right)$$

# Sub-Concept Discovery (con't)



After the convergence of the EM process, we can estimate the associated sub-concept for every instance $x_i$ as

$$sc(\boldsymbol{x}_i) = \arg\max_k \gamma_{ik} \qquad (k = 1, 2, \cdots, M)$$

Then, we can derive the multi-label for every instance by considering the sub-concept belongingness

In making prediction on new data, we use MIML learner to predict its "multi-label" at first, and then use a traditional classifier to map the "multi-label" to original "single-label"

# The Process

*Africa* ← A traditional learner

MIML

(a) *Africa* is a complicated high-level concept

[1 1 1 1 0 0 … 0 0 0 0 0]

[0 0 0 0 0 0 … 0 0 1 1 0]

[0 0 0 0 0 0 … 1 0 0 0 0]

[1 0 0 0 0 0 … 1 0 0 0 0]

[0 0 0 0 0 0 … 0 0 0 1 1]

[1 0 1 1 0 1 … 0 0 0 0 0]

# The SUBCOD Algorithm



The SubCod algorithm

---

1 For multi-instance single-label examples $(X_u, y_u)$ $(u = 1, 2, \cdots, m)$, collect all the instances $\boldsymbol{x} \in X_u$ together and identify the Gaussian mixture components through the EM process detailed in Eqs. 21 to 25.

2 Determine the sub-concept for every instance $\boldsymbol{x} \in X_u$ according to Eq. 26, and then derive the label vector $\boldsymbol{c}_u$ for $X_u$.

3 Make corrections to $\boldsymbol{c}_u$ by optimizing Eq. 27, which results in $\tilde{\boldsymbol{c}}_u$ for $X_u$, and then train a MIML learner $h_t(X)$ on $\{(X_u, \tilde{\boldsymbol{c}}_u)\}$ $(u = 1, 2, \cdots, m)$.

4 Train a classifier $h_y(\tilde{\boldsymbol{c}})$ on $\{(\tilde{\boldsymbol{c}}_u, y_u)\}$ $(u = 1, 2, \cdots, m)$, which maps the derived multi-labels to the original single-labels.

5 Return $y^* = h_y\left(h_t\left(X^*\right)\right)$.

---

# Experiments: Results



Table 12
Predictive accuracy on five multi-instance benchmark data sets

| Compared Algorithms | Data sets | | | | |
|---|---|---|---|---|---|
| | *Musk1* | *Musk2* | *Elephant* | *Tiger* | *Fox* |
| SubCod | 85.0% | **92.1%** | **83.6%** | 80.8% | **61.6%** |
| Dd | 88.0% | 84.0% | N/A | N/A | N/A |
| Em-dd | 84.8% | 84.9% | 78.3% | 72.1% | 56.1% |
| mi-Svm | 87.4% | 83.6% | 82.0% | 78.9% | 58.2% |
| Mi-Svm | 77.9% | 84.3% | 81.4% | **84.0%** | 59.4% |
| Ch-Fd | **88.8%** | 85.7% | 82.4% | 82.2% | 60.4% |

# Take-Home Message

- ✓ Real-world learning tasks are complex; previous simple frameworks may lose useful information
- ✓ MIML is a good framework for learning with ambiguous data

# Multi-Instance Multi-Label Learning




- ✓ Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, Y.-F. Li. MIML: A Framework for Learning with Ambiguous Objects. CORR abs/0808.3231, 2008.
- ✓ Z.-H. Zhou. Mining ambiguous data with multi-instance multi-label learning. ADMA'07 invited talk
- ✓ M.-L. Zhang, Z.-H. Zhou. Multi-label learning by instance differentiation. AAAI'07, pp.669-674
- ✓ Z.-H. Zhou, M.-L. Zhang. Multi-instance multi-label learning with application to scene classification. NIPS'06, pp.1609-1616.


Codes:

- MIMLBoost & MIMLSVM: http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/annex/MIMLBoost&MIMLSVM.htm
- InsDif: http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/annex/InsDif.htm

Data: http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/annex/miml-image-data.htm

# Thanks!