Label Space Coding for Multi-label Classification

Mathematical Machine Learning for Modern Artificial Intelligence

Hsuan-Tien Lin

National Taiwan University

7th 6th 3rd TWSIAM Annual Meeting, 05/25/2019
From Intelligence to Artificial Intelligence

**intelligence**: thinking and acting *smartly*
- humanly
- rationally

**artificial intelligence**: *computers* thinking and acting *smartly*
- humanly
- rationally

*humanly* $\approx$ *smartly* $\approx$ *rationally*
—*are humans rational? :-)*
Traditional vs. Modern [My] Definition of AI

Traditional Definition
humanly $\approx$ intelligently $\approx$ rationally

My Definition
intelligently $\approx$ easily
is your smart phone ‘smart’? :-)

modern artificial intelligence
$=$ application intelligence
Examples of Application Intelligence

**Siri**
By Bernard Goldbach [CC BY 2.0]

**iRobot**
By Yuan-Chou Lo [CC BY-NC-ND 2.0]

**Amazon Recommendations**
By Kelly Sims [CC BY 2.0]

**Vivino**
from nordic.businessinsider.com
Machine Learning and AI

Easy-to-Use

Acting Humanly

Acting Rationally

Machine Learning

*machine learning*: core behind modern (data-driven) AI
ML Connects Big Data and AI

From Big Data to Artificial Intelligence

big data → ML → artificial intelligence

ingredient → tools/steps → dish

“cooking” needs many possible tools & procedures

(Photos Licensed under CC BY 2.0 from Andrea Goh on Flickr)
Bigger Data Towards Better AI

best route by shortest path

best route by current traffic

best route by predicted travel time

big data can make machine look smarter
• human sometimes faster learner on initial (smaller) data
• industry: black plum is as sweet as white

often important to leverage human learning, especially in the beginning
Application: Tropical Cyclone Intensity Estimation

Meteorologists can ‘feel’ & estimate TC intensity from image.

- TC images
- Human learning/analysis
- CNN
- Polar rotation invariance
- ML
- Domain knowledge (HumanI)

Better than current system & ‘trial-ready’

(Chen et al., KDD 2018)
(Chen et al., Weather & Forecasting 2019)
Cost-Sensitive Multiclass Classification
### Patient Status Prediction

**Error Measure = Society Cost**

<table>
<thead>
<tr>
<th></th>
<th>H7N9</th>
<th>cold</th>
<th>healthy</th>
</tr>
</thead>
<tbody>
<tr>
<td>H7N9</td>
<td>0</td>
<td>1000</td>
<td>100000</td>
</tr>
<tr>
<td>cold</td>
<td>100</td>
<td>0</td>
<td>3000</td>
</tr>
<tr>
<td>healthy</td>
<td>100</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

- H7N9 mis-predicted as healthy: **very high cost**
- cold mis-predicted as healthy: **high cost**
- cold correctly predicted as cold: **no cost**

**Human doctors consider costs of decision; how about computer-aided diagnosis?**
Setup: Cost-Sensitive Classification

**Given**

$N$ classification examples (input $\mathbf{x}_n$, label $y_n$) $\in \mathcal{X} \times \{1, 2, \ldots, K\}$

and a ‘proper’ cost matrix $C \in \mathbb{R}^{K \times K}$

**Goal**

a classifier $g(\mathbf{x})$ that pays a small cost $C(y, g(\mathbf{x}))$ on future unseen example $(\mathbf{x}, y)$

Cost-sensitive classification:

more **application-realistic**

than traditional classification
Key Idea: Cost Estimator (Tu and Lin, ICML 2010)

Goal

a classifier $g(x)$ that pays a small cost $C(y, g(x))$ on future unseen example $(x, y)$

consider expected conditional costs $c_x[k] = \sum_{y=1}^{K} C(y, k)P(y|x)$

if $c_x$ known

optimal

$g^*(x) = \arg\min_{1 \leq k \leq K} c_x[k]$

if $r_k(x) \approx c_x[k]$ well

approximately good

$g_r(x) = \arg\min_{1 \leq k \leq K} r_k(x)$

how to get cost estimator $r_k$? regression
Cost Estimator by Per-class Regression

Given

\( N \) examples, each \((\text{input} \, x_n, \, \text{label} \, y_n) \in \mathcal{X} \times \{1, 2, \ldots, K\}\)

- take \( c_n \) as \( y_n \)-th row of \( C \): \( c_n[k] = C(y_n, k) \)

<table>
<thead>
<tr>
<th>input</th>
<th>( c_n[1] )</th>
<th>input</th>
<th>( c_n[2] )</th>
<th>( \ldots )</th>
<th>input</th>
<th>( c_n[K] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>0,</td>
<td>( x_1 )</td>
<td>2,</td>
<td>( \ldots )</td>
<td>( x_1 )</td>
<td>1,</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>1,</td>
<td>( x_2 )</td>
<td>3,</td>
<td></td>
<td>( x_2 )</td>
<td>5,</td>
</tr>
<tr>
<td>( \ldots )</td>
<td></td>
<td>( \ldots )</td>
<td></td>
<td></td>
<td>( \ldots )</td>
<td></td>
</tr>
<tr>
<td>( x_N )</td>
<td>6,</td>
<td>( x_N )</td>
<td>1,</td>
<td></td>
<td>( x_N )</td>
<td>0,</td>
</tr>
</tbody>
</table>

\[ r_1 \]

\[ r_2 \]

\[ r_K \]

want: \( r_k(x) \approx c_x[k] \) for all future \( x \) and \( k \)
The Reduction Framework

cost-sensitive example \((x_n, y_n, c_n)\) \(\Rightarrow\) regression examples \((X_{n,k}, Y_{n,k})\)

\(k = 1, \cdots, K\) \(\Rightarrow\) regression algorithm \(\Rightarrow\) regressors \(r_k(x)\)

\(k \in 1, \cdots, K\) \(\Rightarrow\) cost-sensitive classifier \(g_r(x)\)

1. transform classification examples \((x_n, y_n)\) to regression examples \((x_{n,k}, Y_{n,k}) = (x_n, C(y_n, k))\)

2. use your favorite algorithm on the regression examples and get estimators \(r_k(x)\)

3. for each new input \(x\), predict its class using \(g_r(x) = \arg\min_{1 \leq k \leq K} r_k(x)\)

the reduction-to-regression framework: systematic & easy to implement
A Simple Theoretical Guarantee

\[ g_r(\mathbf{x}) = \arg\min_{1 \leq k \leq K} r_k(\mathbf{x}) \]

**Theorem (Absolute Loss Bound)**

For any set of estimators (cost estimators) \( \{ r_k \}_{k=1}^K \) and for any tuple \((\mathbf{x}, y, \mathbf{c})\) with \( c[y] = 0 = \min_{1 \leq k \leq K} c[k] \),

\[
    c[g_r(\mathbf{x})] \leq \sum_{k=1}^{K} \left| r_k(\mathbf{x}) - c[k] \right|.
\]

**low-cost classifier \( \iff \) accurate estimator**
Our Contributions

In 2010 (Tu and Lin, ICML 2010)

- **tighten** the simple guarantee (+math)
- **propose** loss function (+math) from tighter bound
- **derive** SVM-based model (+math) from loss function

—eventually reaching **superior experimental results**

Six Years Later (Chung et al., IJCAI 2016)

- **propose smoother** loss function (+math) from tighter bound
- **derive** world’s first cost-sensitive deep learning model (+math) from loss function

—eventually reaching **even superior experimental results**

**why are people not using those cool ML works for their AI? :-)**
### Issue 1: Where Do Costs Come From?

#### A Real Medical Application: Classifying Bacteria

- by human doctors: **different treatments** ⇐⇒ **serious costs**
- cost matrix averaged from two doctors:

<table>
<thead>
<tr>
<th></th>
<th>Ab</th>
<th>Ecoli</th>
<th>HI</th>
<th>KP</th>
<th>LM</th>
<th>Nm</th>
<th>Psa</th>
<th>Spn</th>
<th>Sa</th>
<th>GBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ab</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Ecoli</td>
<td>3</td>
<td>0</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>HI</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>KP</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>LM</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Nm</td>
<td>3</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Psa</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>8</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>Spn</td>
<td>6</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Sa</td>
<td>7</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>GBS</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

**issue 2: is cost-sensitive classification really useful?**
Cost-Sensitive vs. Traditional on Bacteria Data

Jan et al. (Academic Sinica) Cost-Sensitive Classification on SERS October 31, 2011 15 / 19

Are cost-sensitive algorithms great?

RBF kernel

0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

OVOSVM

csOSRSVM

csOVOSVM

csFTSVM

algorithms

cost

Cost-sensitive algorithms perform better than regular algorithm (Jan et al., BIBM 2011)

cost-sensitive better than traditional; but why are people still not using those cool ML works for their AI? :-)


18/40
The Problem

- cost-sensitive classifier: low cost but high error rate
- traditional classifier: low error rate but high cost
- how can we get the blue classifiers?: low error rate and low cost
  — **math++** on **multi-objective** optimization (Jan et al., KDD 2012)

now, are people using those cool ML works for their AI? :-)

Lessons Learned from Research on Cost-Sensitive Multiclass Classification

1. more realistic (generic) in academia
   ≠ more realistic (feasible) in application
   e.g. the ‘cost’ of inputting a cost matrix? :-)

2. cross-domain collaboration important
   e.g. getting the ‘cost matrix’ from domain experts

3. not easy to win human trust
   —humans are somewhat multi-objective

4. many battlefields for math towards application intelligence
   e.g. abstraction of goals and needs
Label Space Coding for Multilabel Classification
What Tags?

?: \{ machine learning, data structure, data mining, object oriented programming, artificial intelligence, compiler, architecture, chemistry, textbook, children book, \ldots \text{etc.} \}

a multilabel classification problem: tagging input to multiple categories
Binary Relevance: Multilabel Classification via Yes/No

<table>
<thead>
<tr>
<th>Binary Classification</th>
<th>multilabel w/ L classes: L Y/N questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>{yes, no}</td>
<td>machine learning (Y), data structure (N), data mining (Y), OOP (N), AI (Y), compiler (N), architecture (N), chemistry (N), textbook (Y), children book (N), etc.</td>
</tr>
</tbody>
</table>

- **Binary Relevance** approach: transformation to **multiple isolated binary classification**
- **disadvantages:**
  - **isolation**—hidden relations not exploited (e.g. ML and DM highly correlated, ML subset of AI, textbook & children book disjoint)
  - **unbalanced**—few yes, many no

**Binary Relevance**: simple (& good) benchmark with known disadvantages
## From Label-set to Coding View

<table>
<thead>
<tr>
<th>label set</th>
<th>apple</th>
<th>orange</th>
<th>strawberry</th>
<th>binary code</th>
</tr>
</thead>
<tbody>
<tr>
<td>{o}</td>
<td>0 (N)</td>
<td>1 (Y)</td>
<td>0 (N)</td>
<td>[0, 1, 0]</td>
</tr>
<tr>
<td>{a, o}</td>
<td>1 (Y)</td>
<td>1 (Y)</td>
<td>0 (N)</td>
<td>[1, 1, 0]</td>
</tr>
<tr>
<td>{a, s}</td>
<td>1 (Y)</td>
<td>0 (N)</td>
<td>1 (Y)</td>
<td>[1, 0, 1]</td>
</tr>
<tr>
<td>{o}</td>
<td>0 (N)</td>
<td>1 (Y)</td>
<td>0 (N)</td>
<td>[0, 1, 0]</td>
</tr>
<tr>
<td>{}</td>
<td>0 (N)</td>
<td>0 (N)</td>
<td>0 (N)</td>
<td>[0, 0, 0]</td>
</tr>
</tbody>
</table>

Subset of $2^{\{1, 2, \ldots, L\}} \Leftrightarrow$ length-$L$ binary code
General Compressive Sensing

Sparse (many 0) binary vectors $y \in \{0, 1\}^L$ can be robustly compressed by projecting to $M \ll L$ basis vectors $\{p_1, p_2, \ldots, p_M\}$

Comp. Sensing for Multilabel Classification (Hsu et al., NeurIPS 2009)

1. **compress**: encode original data by **compressive sensing**
2. **learn**: get regression function from compressed data
3. **decode**: decode regression predictions to sparse vector by **compressive sensing**

**Compressive Sensing**: seemly strong competitor from related theoretical analysis
Our Proposed Approach:
Compressive Sensing $\Rightarrow$ PCA

Principal Label Space Transformation (PLST), i.e. PCA for Multilabel Classification (Tai and Lin, NC Journal 2012)

1. **compress**: encode original data by **PCA**
2. **learn**: get regression function from compressed data
3. **decode**: decode regression predictions to label vector by reverse **PCA + quantization**

**does PLST perform better than CS?**
Hamming Loss Comparison: PLST vs. CS

- **PLST** better than **CS**: faster, **better** performance
- similar findings across **data sets** and **regression algorithms**

Why? **CS** creates **harder-to-learn** regression tasks
Our Works Continued from PLST

1. **Compression Coding** (Tai & Lin, NC Journal 2012 with 216 citations)
   - condense for efficiency: better (than CS) approach PLST
   - key tool: PCA from Statistics/Signal Processing

2. **Learnable-Compression Coding** (Chen & Lin, NeurIPS 2012 with 157 citations)
   - condense learnably for better efficiency: better (than PLST) approach CPLST
   - key tool: Ridge Regression from Statistics (+ PCA)

3. **Cost-Sensitive Coding** (Huang & Lin, ECML Journal Track 2017)
   - condense cost-sensitively towards application needs: better (than CPLST) approach CLEMS
   - key tool: Multidimensional Scaling from Statistics

cannot thank statisticians enough for those tools!
Lessons Learned from Label Space Coding for Multilabel Classification

?: \{ machine learning, data structure, data mining, object-oriented programming, artificial intelligence, compiler, architecture, chemistry, textbook, children book, \ldots etc. \}

1. Is Statistics the same as ML? Is Statistics the same as AI?
   - does it really matter?
   - Modern AI should embrace every useful tool from every field & any necessary math

2. ‘application intelligence’ tools not necessarily most sophisticated ones
   e.g. PCA possibly more useful than CS for label space coding

3. more-cited paper \neq more-useful AI solution
   —citation count not the only impact measure

4. are people using those cool ML works for their AI?
   —we wish!
Active Learning by Learning
Active Learning: Learning by ‘Asking’

labeling is expensive:

active learning ‘question asking’
—query $y_n$ of chosen $x_n$

unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$

labeled training examples

(🍎, +1), (🍎, +1), (🍎, +1)
(香蕉, -1), (香蕉, -1), (香蕉, -1)

learning algorithm $\mathcal{A}$

final hypothesis $g \approx f$

active: improve hypothesis with fewer labels (hopefully) by asking questions strategically
Pool-Based Active Learning Problem

**Given**

- labeled pool $\mathcal{D}_l = \{(\text{feature } x_n, \text{label } y_n \text{ (e.g. IsApple?)})\}_{n=1}^{N}$
- unlabeled pool $\mathcal{D}_u = \{\tilde{x}_s\}_{s=1}^{S}$

**Goal**

design an algorithm that iteratively

1. **strategically query** some $\tilde{x}_s$ to get associated $\tilde{y}_s$
2. move $(\tilde{x}_s, \tilde{y}_s)$ from $\mathcal{D}_u$ to $\mathcal{D}_l$
3. learn classifier $g^{(t)}$ from $\mathcal{D}_l$

and improve **test accuracy of** $g^{(t)}$ w.r.t #queries

**how to query strategically?**
How to Query Strategically?

Strategy 1
ask **most confused** question

Strategy 2
ask **most frequent** question

Strategy 3
ask **most debateful** question

- **choosing** one single strategy is **non-trivial**:

  ![Graphs showing accuracy vs. percentage of unlabelled data for different strategies.](image)

**application intelligence: how to choose strategy smartly?**
Idea: Trial-and-Reward Like Human

when do humans trial-and-reward? gambling

K strategies: \( A_1, A_2, \ldots, A_K \)

try one strategy

“goodness” of strategy as reward

K bandit machines: \( B_1, B_2, \ldots, B_K \)

try one bandit machine

“luckiness” of machine as reward

intelligent choice of strategy \( \implies \) intelligent choice of bandit machine
Active Learning by Learning (Hsu and Lin, AAAI 2015)

Given: $K$ existing active learning strategies

for $t = 1, 2, \ldots, T$

1. let some bandit model decide strategy $A_k$ to try
2. query the $\tilde{x}_s$ suggested by $A_k$, and compute $g^{(t)}$
3. evaluate goodness of $g^{(t)}$ as reward of trial to update model

only remaining problem: what reward?
Design of Reward

**ideal reward** after updating classifier \( g(t) \) by the query \((x_{nt}, y_{nt})\):

accuracy of \( g(t) \) on **test set** \( \{(x'_m, y'_m)\}_{m=1}^M \)

—test accuracy **infeasible** in practice because labeling **expensive**

**more feasible reward**: training accuracy on the fly

accuracy of \( g(t) \) on **labeled pool** \( \{(x_{n\tau}, y_{n\tau})\}_{\tau=1}^t \)

—but **biased** towards **easier** queries

**weighted training accuracy** as a better reward:

acc. of \( g(t) \) on **inv.-prob. weighted labeled pool** \( \{(x_{n\tau}, y_{n\tau}, \frac{1}{p_\tau})\}_{\tau=1}^t \)

—‘bias correction’ from querying probability within bandit model

**Active Learning by Learning (ALBL):**

bandit + **weighted training acc. as reward**
Comparison with Single Strategies

- no single best strategy for every data set
  —choosing needed
- proposed ALBL consistently matches the best
  —similar findings across other data sets

ALBL: effective in making intelligent choices
**weighted training accuracy**\[ \frac{1}{t} \sum_{\tau=1}^{t} \frac{1}{p_\tau} \left[ y_{n_\tau} = g^{(t)}(x_{n_\tau}) \right] \] as reward

- is reward **unbiased estimator** of test performance?  
  no for learned $g^{(t)}$ (yes for fixed $g$)

- is reward fixed before playing?  
  no because $g^{(t)}$ learned from $(x_{n_\tau}, y_{n_\tau})$

- is reward **independent** of each other?  
  no because past history all in reward

---

ALBL: tools from statistics + **wild/unintended usage**

---

‘application intelligence’ outcome:  
**open-source tool** released  
(https://github.com/ntucllab/libact)
scalability bottleneck of ‘application intelligence’: choice of methods/models/parameter/. . .

think outside of the math box: ‘unintended’ usage may be good enough

important to be brave yet patient —idea: 2012 —paper (Hsu and Lin, AAAI 2015); software (Yang et al., 2017)
Summary

- **ML for (Modern) AI:**
  tools + human knowledge
  \[\Rightarrow\text{easy-to-use application intelligence}\]

- **ML Research for Modern AI:**
  need to be **more open-minded**
  —in methodology, in collaboration, in KPI

- **Math** in ML Research for Modern AI:
  —**new setup/need/goal** & wider usage of tools

Thank you! Questions?