



Big data, Big innovation

次のイノベーションは、ビッグデータから

Multi-label Classification via Feature-aware Implicit Label Space Encoding

AS 池田裕章

2014.9.3



Introduction



マルチラベル分類

- マルチクラス分類（多クラス分類）とは異なる
- マルチクラス分類では、一つの事例が一つのクラスに分類されるが、
マルチラベル分類では、一つの事例が複数のクラスに同時に分類されうる

マルチクラス分類

ドキュメント1 ⇒ 政治
ドキュメント2 ⇒ 経済
ドキュメント3 ⇒ 科学

マルチラベル分類

ドキュメント1 ⇒ 政治
ドキュメント2 ⇒ 政治、経済
ドキュメント3 ⇒ 科学、物理

構造

X_1	X_2	X_3	X_4	X_5	C_1	C_2	C_3	C_4	$Y \subseteq \mathcal{Y}$
3.2	1.4	4.7	7.5	3.7	1	0	0	1	$\{\lambda_1, \lambda_4\}$
2.8	6.3	1.6	4.7	2.7	0	0	1	1	$\{\lambda_3, \lambda_4\}$
7.7	6.2	4.1	3.3	7.7	1	0	0	1	$\{\lambda_1, \lambda_4\}$
9.2	0.4	2.8	0.5	3.9	0	1	0	0	$\{\lambda_2\}$
5.5	5.3	4.9	0.6	6.6	1	1	1	0	$\{\lambda_1, \lambda_2, \lambda_3\}$

- マルチラベル分類はマルチクラス分類に比べて困難
 - ラベル関連の問題
「政治」と「経済」が同時に表れやすい場合、「政治」だけ、「経済」だけの特徴を学習するのは難しい
 - ラベルの不均一性 (label-imbalance)
個々のラベルで見た場合、出現頻度が非常に少なくなる
 - 分類パターンの増大
ラベル数が増えるに従って、分類パターンは指数的に増加
(ラベル数をKとすると 2^K 通りの分類があり得る)

既存手法

■ マルチラベル分類の手法として、下記のようなものが提案されている

- Binary relevance (BR)
あるラベルがあるかないかの複数の2値問題に変換
- Label power set (LP)
ラベルのあり・なしをビット列で表し、 2^n 個 (n はラベル数) のラベル集合に変換
- Copy and copy weight
2個ラベルがついた事例があったら、二つにコピーして、それぞれの重みを1/2ずつにする
- Ranking by pairwise comparison
二つのラベルを考えて、一方だけが付く場合に正負のラベルを各事例について考える。
- Calibrated label ranking
Pairwise comparisonに加え、ラベル付けの順位を扱う目的で、二値の仮想ラベルを加える。
- Prune problem transformation
label power set で低頻度のラベルの組み合わせを無視する改良
- Random k-labelsets
大きなラベル集合を複数の集合にわけ、それぞれをlabel power setで解く

(参考) Mining Multi-label Data

<http://lpis.csd.auth.gr/publications/tsoumakas09-dmkdh.pdf>

Large label space problem

- 近年、ユーザが記事や画像、動画に自由にタグを付けるようなサービスが増加
- 上記のようなサービスでは、タグの数が数万～数百万といったオーダー
- 多くの既存手法は、ラベル数個以上の分類モデルを組み合わせるため、計算コストが膨大になり、現実的には計算困難になる

Label Space Dimension Reduction

- ラベル数が膨大な場合の問題を解決する方法として、近年Label Space Dimension Reduction(LSDR)という方法が提案されている

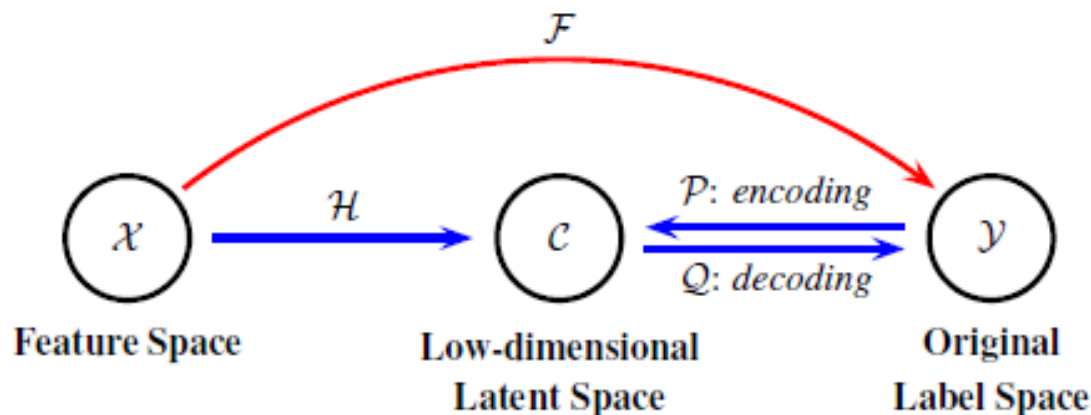


Figure 1. An illustration of the principles of traditional multi-label classification methods (red) and those with LSDR (blue).



Label Space Dimension Reduction

- エンコードされた低次元空間において予測モデルを学習し、デコード関数によって元のラベル空間に復元
- 予測モデルの精度が十分高く、デコードプロセスの再現性が十分に高ければ、予測精度をほとんど減少させることなく計算コストを大幅に削減可能



BrainPad

論文要旨

- 本論文では、予測精度と復元性を保ちつつ低次元コード行列Cを構成する方法として、
Feature-aware Implicit label space Encoding (FaIE)
という手法を提案
- ポイントは2点
 - Feature-aware
入力データの特徴を反映するように低次元コードベクトルを構築
 - Implicit
エンコード関数に、ラベル空間に対して線形といった仮説を置かない
(エンコード関数をexplicitに求めない)

■ LSDRに関する既存研究

- Compressed sensing (CS) (Hsu et al. , 2009)
- BML-CS (Kapoor et al. , 2012)
- PLST (Tai and Lin, 2010)
- CPLST (Chen and Lin, 2012)
- Linear Gaussian random projection (Zhou et al. 2012)
- MLC-BMaD (Wicker et al., 2012)
- ML-CSSP (Bi & Kwok, 2013)

■ MLC-BMaD以外、explicitにエンコード関数を要求

■ MLC-BMaDはfeature-unaware



Feature-aware implicit label space encoding



BrainPad

準備

■ 表記は以下の通り

- データ数 N
- 特徴数 F
- ラベル数 K
- データ集合

$\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^F$ and $\mathbf{y}_i \in \mathcal{Y} \subset \{0, 1\}^K$

- 通常のマルチラベル分類のアプローチ

$\{\mathcal{F}_i | \mathcal{F}_i : \mathcal{X} \mapsto \mathcal{Y}_i, i = 1, 2, \dots, K\}$, where $\mathcal{Y}_i \subset \{0, 1\}$

- LSDRのアプローチ

Encoding process $\mathcal{P} : \mathcal{Y} \mapsto \mathcal{C}$

Decoding process $\mathcal{Q} : \mathcal{C} \mapsto \mathcal{Y}$

Learning process $\{\mathcal{H}_i | \mathcal{H}_i : \mathcal{X} \mapsto \mathcal{C}_i, i = 1, 2, \dots, L\}$

$\mathcal{C} \subset \mathbb{R}^L (L \ll K)$

FaIEのアプローチ

- ラベル行列**Y**がコード行列**C**とデコード行列**D**に分解できるとし、エンコードプロセス**P**に制約を置かずに直接**C**と**D**を学習する

$$Y \sim CD$$

- 一般にLSDRの精度はコード行列**C**がデータの特徴と強く相関している方が良く、デコードプロセスにおいて元のラベルを正しく復元できた方が良い
- そこで、FaIEでは予測能力(Predictivity)と復元能力(Recoverability)を結合させた次の関数を最大化するようにコード行列**C**とデコード行列**D**を決める

$$\Omega = \max_{C,D} \underbrace{\Phi(Y, C, D)}_{\text{Recoverability}} + \alpha \underbrace{\Psi(X, C)}_{\text{Predictivity}}$$

Recoverability、Predictivity
の定義については後述

- α は予測能力と復元能力の重要度を調整するパラメータ
- 後で示すように、**D**は**C**が決まれば一意に決まるので、実際は以下の様に**C**に関する最大化のみを考えればよい

$$\Omega = \max_C \Phi(Y, C) + \alpha \Psi(X, C)$$

復元能力(Recoverability)

- ラベル行列の復元に関しては、以下の式を最小化すれば良い

$$\mathcal{E}(Y, C, D) = \|Y - CD\|_{fro}^2$$

- **D**に関して上記の式の最小問題を解くと、以下の式が導かれる

$$D = (C^T C)^{-1} C^T Y$$

- 冗長性を排除し、表現をコンパクトにするため、**C**に対して直交性を仮定すると、**D**は以下のようなになる

$$C^T C = I_{L \times L}$$

$$D = C^T Y$$

復元能力(Recoverability)

- ラベル行列の復元性の式にDを代入

$$\mathcal{E}(Y, C, D) = \text{Tr}[Y^T Y - Y^T C C^T Y]$$

- $\text{Tr}[Y^T Y]$ は定数であるため、FaIEでは以下の式をRecoverabilityとして定義する

$$\begin{aligned} \Phi(Y, C) = & \text{Tr}[Y^T C C^T Y] = \text{Tr}[C^T Y Y^T C] \\ \text{s.t.} \quad & C^T C = I \end{aligned}$$

予測能力(Predictivity)

- 低次元隠れ空間の予測能力を改善するには、コード行列 \mathbf{C} が特徴空間 \mathbf{X} と強く相関していると良い(Zhang & Schneider, 2011)
- そこで特徴空間 \mathbf{X} の線形写像ベクトル \mathbf{r} を導入し、コード行列の列要素 \mathbf{c} と特徴空間の相関を考える（カーネルを用いた非線形への拡張は後述）
- 特徴空間とコードベクトルの相関

$$\rho(X, \mathbf{c}) = \frac{(\mathbf{X}\mathbf{r})^T \mathbf{c}}{\sqrt{(\mathbf{X}\mathbf{r})^T (\mathbf{X}\mathbf{r})} \sqrt{\mathbf{c}^T \mathbf{c}}}$$

- \mathbf{C} の直交性より $\mathbf{c}^T \mathbf{c} = 1$ であるため、以下の最適化問題を解くことで \mathbf{c} に対して相関を最大にする写像ベクトル \mathbf{r} が求まる

$$\max (\mathbf{X}\mathbf{r})^T \mathbf{c} \quad s.t. \quad (\mathbf{X}\mathbf{r})^T \mathbf{X}\mathbf{r} = 1$$

$$\mathbf{r}^* = \frac{(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{c}}}$$

予測能力(Predictivity)

- \mathbf{r}^* を相関係数の式に代入したものを、 \mathbf{C} の列要素 \mathbf{c} に対するPredictivityと定義する

$$\psi(X, \mathbf{c}) = (X\mathbf{r}^*)^T \mathbf{c} = \sqrt{\mathbf{c}^T \Delta \mathbf{c}}$$

$$\text{where } \Delta = X(X^T X)^{-1} X^T$$

- コード行列 \mathbf{C} 全体に対するPredictivityは以下の様に定義される

$$\Psi(X, C) = \sum_i C_{\cdot, i}^T \Delta C_{\cdot, i} = \text{Tr}[C^T \Delta C]$$

$$s.t. \quad C^T C = I$$

予測能力(Predictivity)

- 前ページで定義されたPredictivityに対して、以下の定理が成立する

Lemma 1. *For any given matrix $C_{N \times L}$ satisfying $C^T C = I$, $\text{Tr}[C^T \Delta C]$ has an upper bound being $\min(L, \text{rank}(\Delta))$.*

- これはPredictivityに対して低次元隠れ空間の次元に依存する上限が存在することを意味しており、低次元に圧縮するほど予測性能が減少することを示唆している
- また、 α のチューニングにおいて、上限以上に α を増加させる必要がないことを意味している

コード行列の導出

- RecoverabilityとPredictivityを代入したFaIEの最適化関数は、以下となる

$$\begin{aligned}\Omega &= \max_C \text{Tr}[C^T Y Y^T C] + \alpha \text{Tr}[C^T \Delta C] \\ &= \max_C \text{Tr}[C^T (Y Y^T + \alpha \Delta) C] \\ s.t. \quad &C^T C = I \\ \text{where } \Delta &= X(X^T X)^{-1} X^T\end{aligned}$$

- このとき、上記最適化問題は**C**の個々の列要素に対して以下のsub-problemと同値になる

$$\begin{aligned}\Omega_i &= \max_{C_{:,i}} C_{:,i}^T (Y Y^T + \alpha \Delta) C_{:,i} \\ s.t. \quad &C_{:,i}^T C_{:,i} = 1, C_{:,i}^T C_{:,j} = 0 \ (\forall j < i)\end{aligned}$$

- ラグランジュ未定乗数法を用いれば、以下の固有値問題に帰着するので、固有値上位 **L** 個に関する固有ベクトルから**C**を構成すれば良い

$$(Y Y^T + \alpha \Delta) C_{:,i} = \underline{\lambda_i} C_{:,i}$$

ラグランジュ未定乗数



BrainPad

アルゴリズム

Algorithm 1 Implementation of FaIE

Input: Feature matrix X_{tr} and tagging matrix Y_{tr} of training set, feature matrix X_{ts} of test set, predefined parameter α and target dimension L of the latent space

Output: Predicted tagging matrix Y_{ts} of test set

- 1: $\Delta = X_{tr}(X_{tr}^T X_{tr})^{-1} X_{tr}^T$
 - 2: $\Omega = Y_{tr} Y_{tr}^T + \alpha \Delta$
 - 3: $C_{tr} = \text{eigenvector}(\Omega, L)$ {normalized eigenvectors of Ω corresponding to the top L largest eigenvalues}
 - 4: $D = C_{tr}^T Y_{tr}$ {decoding matrix}
 - 5: learn predictive models: $\mathcal{H}(X_{tr}) \rightarrow C_{tr}$
 - 6: $C_{ts} = \mathcal{H}(X_{ts})$ {predicted code vectors of test set}
 - 7: $Y_{ts} = \text{round}(C_{ts} D)$
-

Linear Encoding Case

- FaIEではエンコード関数を陽に求めないが、特殊な場合として線形のエンコード行列 P を用いる場合を考える

$$C = YP$$

- この場合、FaIEの最適化関数は下記のようになる

$$\begin{aligned} \Omega = \max_P \quad & \text{Tr}[P^T(Y^T Y Y^T Y + \alpha Y^T \Delta Y)P] \\ \text{s.t.} \quad & P^T Y^T Y P = I \end{aligned}$$

- Implicitな場合と同様に P の列要素に関する最適化問題は下記の固有値問題に帰着する

$$(Y^T Y Y^T Y + \alpha Y^T \Delta Y)P_{:,i} = \lambda_i(Y^T Y)P_{:,i}$$

- 上記の式において、固有値上位 L 個の固有ベクトルより P を構成すれば良い。このとき、デコード行列は以下となる

$$D = (YP)^T Y$$

Kernel version

- kernel-FaIEとして、相関を求める際に線形写像ではなく、カーネルを用いた非線形写像を考える
- 特徴ベクトル \mathbf{x}_i の再生核ヒルベルト空間への写像を θ_i とし、その写像からなるカーネル特徴行列を Θ とする
- このとき、写像ベクトル \mathbf{r}' が以下の様に構成されるとする

$$\mathbf{r}' = \Theta^T \beta$$

- ここで、 β はN次元の重みベクトル
- 相関係数

$$\begin{aligned} \rho(\Theta, \mathbf{c}) &= \frac{(\Theta \Theta^T \beta)^T \mathbf{c}}{\sqrt{(\Theta \Theta^T \beta)^T (\Theta \Theta^T \beta)} \sqrt{\mathbf{c}^T \mathbf{c}}} && \text{カーネル行列} \\ &= \frac{(K \beta)^T \mathbf{c}}{\sqrt{(K \beta)^T (K \beta)} \sqrt{\mathbf{c}^T \mathbf{c}}} && \text{where } K = \Theta \Theta^T \end{aligned}$$

- 以下同様の計算より、kernel-FaIEの最適化関数は下記となる

$$\begin{aligned} \Omega &= \max_C \text{Tr}[C^T (Y Y^T + \alpha \Gamma) C] \\ s.t. \quad & C^T C = I \end{aligned} \quad \Gamma = K (K^T K)^{-1} K^T$$

他の研究との関係

■ PLST

FaIEにおいて $\alpha=0$ かつ線形のエンコード行列 P を用いた場合と等価

$$\Omega = \max_C \text{Tr}[C^T Y Y^T C], \quad s.t. \quad C^T C = I$$

■ CPLST

FaIEにおいてPredictivityのみを考え、線形のエンコード行列 P を用いた場合に類似

$$\text{FaIE : } \Omega = \max_P \text{Tr}[P^T Y^T \Delta Y P], \quad s.t. \quad P^T Y^T Y P = I$$

$$\begin{aligned} \text{CPLST : } \Omega_1 &= \max_P \text{Tr}[P^T Y^T H Y P], \quad s.t. \quad P^T P = I \\ H &= X(X^T X)^{-1} X^T = \Delta \end{aligned}$$



Experiments

Experiments

- 以下の4種類のデータに対してテスト

Table 1. Statistics of datasets

	domain	instances	labels	features
<i>delicious</i>	text	5,000	983	500
<i>CAL500</i>	music	502	174	68
<i>mediamill</i>	video	5,000	101	120
<i>ESPGame</i>	image	5,000	1,932	516

- ベンチマーク

- Binary Relevance (BR) (Furnkranz et al., 2008)
- Compressed sensing (CS) (Hsu et al., 2009)
- PLST (Tai & Lin, 2010)
- CPLST (Chen & Lin, 2012)
- kernel-CPLST (Chen & Lin, 2012)
- MLC-BMaD (Wicker et al., 2012)
- ML-CSSP (Bi & Kwok, 2013)

Experiments

- 5-fold cross validationを用いてパラメータをチューニング

- 評価指標

- Label based macro-F1

$$macro_F_1 = \frac{1}{Q} \sum_{j=1}^Q \frac{2 \times p_j \times r_j}{p_j + r_j}$$

- Example based accuracy

$$accuracy(h) = \frac{1}{N} \sum_{i=1}^N \frac{|h(\mathbf{x}_i) \cap \mathcal{Y}_i|}{|h(\mathbf{x}_i) \cup \mathcal{Y}_i|}$$

- 隠れ空間における予測モデルはリッジ回帰を利用

Table 2. Experimental results on *delicious*

<i>L/K</i>		<i>label-based macroF1</i>					<i>example-based Accuracy</i>				
		10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
BR	L-SVM	0.0790					0.1419				
	L-RR	0.0308					0.0810				
CS		0.0057	0.0172	0.0341	0.0366	0.0320	0.0260	0.0534	0.0854	0.0932	0.0927
PLST		0.0196	0.0214	0.0217	0.0217	0.0217	0.0734	0.0762	0.0768	0.0769	0.0768
CPLST		0.0240	0.0244	0.0244	0.0244	0.0244	0.0787	0.0787	0.0788	0.0788	0.0788
MLC-BMaD		0.0180	0.0214	0.0265	0.0277	0.0288	0.0525	0.0682	0.0690	0.0703	0.0705
ML-CSSP		0.0139	0.0197	0.0230	0.0253	0.0266	0.0555	0.0659	0.0724	0.0729	0.0760
R-FaIE (\sim PLST)		0.0197	0.0213	0.0216	0.0215	0.0216	0.0736	0.0764	0.0771	0.0772	0.0771
P-LinearFaIE (\sim CPLST)		0.0369	0.0407	0.0412	0.0412	0.0412	0.0892	0.0971	0.0987	0.0987	0.0987
OP-FaIE		0.0472	0.0515	0.0526	0.0527	0.0514	0.1073	0.1083	0.1084	0.1082	0.1083
LinearFaIE		0.0411	0.0431	0.0434	0.0435	0.0435	0.0984	0.1058	0.1061	0.1061	0.1061
FaIE		0.0544	0.0591	0.0602	0.0603	0.0586	0.1198	0.1207	0.1203	0.1202	0.1116
<i>kernel-CPLST</i>		0.0341	0.0341	0.0341	0.0341	0.0341	0.1048	0.1048	0.1048	0.1048	0.1048
<i>kernel-FaIE</i>		0.0566	0.0688	0.0726	0.0744	0.0750	0.1448	0.1496	0.1506	0.1507	0.1508

Table 3. Experimental results on *CAL500*

<i>L/K</i>		<i>label-based macroF1</i>					<i>example-based Accuracy</i>				
		10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
BR	L-SVM	0.1397					0.2436				
	L-RR	0.0569					0.1995				
CS		0.0677	0.0820	0.0906	0.0976	0.1142	0.1130	0.1299	0.1626	0.1904	0.1835
PLST		0.0604	0.0605	0.0606	0.0609	0.0608	0.2099	0.2103	0.2103	0.2106	0.2104
CPLST		0.0640	0.0643	0.0644	0.0645	0.0645	0.2003	0.2007	0.2009	0.2010	0.2010
MLC-BMaD		0.0485	0.0444	0.0420	0.0472	0.0468	0.1286	0.1215	0.1194	0.1244	0.1255
ML-CSSP		0.0453	0.0498	0.0507	0.0528	0.0543	0.1806	0.1880	0.1913	0.1958	0.1966
R-FaIE (\sim PLST)		0.0596	0.0600	0.0600	0.0601	0.0600	0.2099	0.2109	0.2109	0.2109	0.2106
P-LinearFaIE (\sim CPLST)		0.0800	0.0976	0.1000	0.0998	0.0998	0.2093	0.2210	0.2205	0.2206	0.2206
OP-FaIE		0.1034	0.1080	0.1088	-	-	0.2283	0.2262	0.2251	-	-
LinearFaIE		0.1062	0.1107	0.1105	0.1105	0.1105	0.2329	0.2300	0.2301	0.2299	0.2300
FaIE		0.1199	0.1245	0.1260	0.1249	0.1245	0.2413	0.2414	0.2417	0.2384	0.2379
<i>kernel-CPLST</i>		0.0754	0.0774	0.0774	0.0774	0.0774	0.2139	0.2148	0.2148	0.2148	0.2148
<i>kernel-FaIE</i>		0.1178	0.1243	0.1250	0.1290	0.1291	0.2429	0.2443	0.2422	0.2430	0.2419

Table 4. Experimental results on *mediamill*

<i>L/K</i>		<i>label-based macroF1</i>					<i>example-based Accuracy</i>				
		10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
BR	L-SVM	0.0554					0.3130				
	L-RR	0.0454					0.4173				
CS		0.0056	0.0145	0.0134	0.0311	0.0274	0.0103	0.0296	0.0343	0.1403	0.1357
PLST		0.0419	0.0435	0.0436	0.0436	0.0436	0.4117	0.4144	0.4142	0.4141	0.4143
CPLST		0.0433	0.0440	0.0440	0.0440	0.0440	0.4142	0.4148	0.4149	0.4149	0.4148
MLC-BMaD		0.0412	0.0426	0.0425	0.0423	0.0425	0.4027	0.4037	0.4027	0.4027	0.4027
ML-CSSP		0.0343	0.0406	0.0428	0.0416	0.0437	0.3378	0.3954	0.4058	0.4045	0.4148
R-FaIE (\sim PLST)		0.0419	0.0438	0.0441	0.0440	0.0440	0.4121	0.4150	0.4150	0.4150	0.4151
P-LinearFaIE (\sim CPLST)		0.0425	0.0444	0.0453	0.0452	0.0452	0.4135	0.4145	0.4151	0.4155	0.4156
OP-FaIE		0.0450	0.0469	0.0472	0.0474	0.0471	0.4163	0.4185	0.4182	0.4177	0.4179
LinearFaIE		0.0444	0.0463	0.0464	0.0461	0.0461	0.4154	0.4172	0.4160	0.4153	0.4154
FaIE		0.0570	0.0595	0.0609	0.0602	0.0607	0.4327	0.4339	0.4338	0.4334	0.4336
<i>kernel-CPLST</i>		0.0492	0.0521	0.0521	0.0521	0.0521	0.4228	0.4234	0.4240	0.4240	0.4240
<i>kernel-FaIE</i>		0.0682	0.0805	0.0852	0.0857	0.0855	0.4401	0.4440	0.4447	0.4423	0.4385

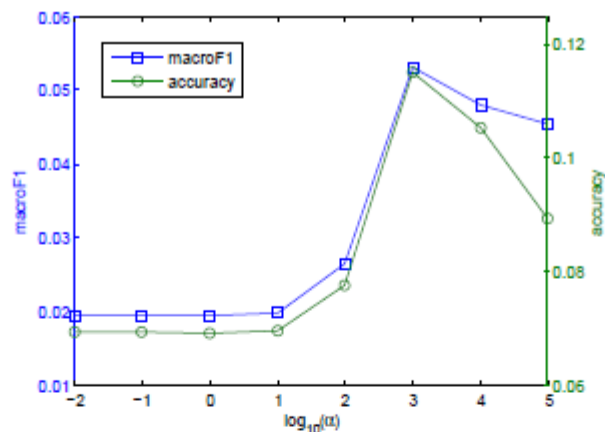
Table 5. Experimental results on *ESPGame*

<i>L/K</i>		<i>label-based macroF1</i>					<i>example-based Accuracy</i>				
		5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
BR	L-SVM	0.0213					0.0726				
	L-RR	0.0014					0.0452				
CS		0.0004	0.0006	0.0013	0.0013	0.0018	0.0166	0.0213	0.0445	0.0467	0.0505
PLST		0.0008	0.0008	0.0008	0.0008	0.0008	0.0457	0.0457	0.0457	0.0457	0.0457
CPLST		0.0008	0.0008	0.0008	0.0008	0.0008	0.0469	0.0470	0.0470	0.0470	0.0470
MLC-BMaD		0.0009	0.0009	0.0010	0.0009	0.0010	0.0450	0.0449	0.0458	0.0446	0.0446
ML-CSSP		0.0008	0.0005	0.0007	0.0007	0.0008	0.0432	0.0280	0.0357	0.0321	0.0397
R-FaIE (\sim PLST)		0.0008	0.0008	0.0008	0.0008	0.0008	0.0455	0.0455	0.0453	0.0455	0.0453
P-LinearFaIE (\sim CPLST)		0.0015	0.0018	0.0020	0.0021	0.0021	0.0394	0.0491	0.0554	0.0581	0.0581
OP-FaIE		0.0019	0.0024	0.0024	0.0024	0.0026	0.0593	0.0596	0.0593	0.0591	0.0595
LinearFaIE		0.0021	0.0022	0.0023	0.0024	0.0024	0.0556	0.0638	0.0639	0.0659	0.0670
FaIE		0.0023	0.0028	0.0028	0.0029	0.0028	0.0641	0.0640	0.0666	0.0669	0.0690
<i>kernel-CPLST</i>		0.0009	0.0009	0.0009	0.0009	0.0009	0.0488	0.0483	0.0488	0.0483	0.0488
<i>kernel-FaIE</i>		0.0038	0.0045	0.0054	0.0058	0.0062	0.0831	0.0834	0.0837	0.0840	0.0841

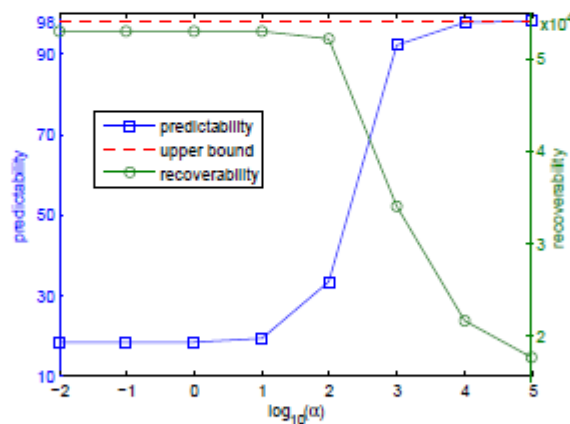
Table 6. Average training costs (in seconds) of algorithms (“performing LSDR + training predictive models”) with $L/K = 10\%$.

		<i>delicious</i>		<i>ESPGame</i>	
BR	L-SVM	28.168	(0.000 + 28.168)	377.035	(0.000 + 377.035)
	L-RR	49.993	(0.000 + 49.993)	101.940	(0.000 + 101.940)
CS		5.057	(0.064 + 4.993)	10.364	(0.203 + 10.161)
PLST		5.387	(0.396 + 4.991)	12.783	(2.581 + 10.201)
CPLST		7.080	(2.132 + 4.948)	16.684	(6.486 + 10.198)
MLC-BMaD		21.267	(16.276 + 4.990)	77.251	(67.058 + 10.193)
ML-CSSP		5.346	(0.371 + 4.975)	12.819	(2.640 + 10.180)
FaIE		10.012	(5.032 + 4.980)	20.110	(9.909 + 10.201)

Matlab on a server with an Intel Xeon E5620 CPU and 24G RAM, except that BR with L-SVM is conducted using LIBLINEAR



(a) Label-based macroF1 and example-based accuracy



(b) Values of recoverability and predictability

Figure 2. Effects of α in FaIE on the performance of multi-label classification (sub-figure 2(a)) and the values of recoverability and predictability (sub-figure 2(b)) on *delicious*, with $L/K = 10\%$ and the theoretical upper bound $\min(L, \text{rank}(\Delta)) = 98$.



BrainPad

まとめ

- マルチラベル分類の手法としてFaIEを紹介
- ラベル数が多い場合でも比較的少ない計算コストで精度をほとんど落とさずに計算可能
- アルゴリズムが簡単な割に精度が良さそう