

素性空間再構成による Word-Sense Disambiguation

高村 大也 山田寛康 工藤拓

山本薫 松本 裕治

奈良先端科学技術大学院大学
情報科学研究科自然言語処理学講座
〒630-0101 奈良県生駒市高山町 8916-5
0743-72-5246, 5248

{hiroya-t,hiroya-y,taku-ku,kaoru-ya,matsu}@is.aist-nara.ac.jp

あらまし

語義の曖昧性解消のための新しい手法を提案する。提案手法では、独立成分分析 (ICA) や主成分分析 (PCA) を用いて複数の素性空間を再構成し、各々の空間における曖昧性解消結果を重み付き多数決する。また、手法の有効性を SENSEVAL のデータを用いて実験的に示した。本手法は幅広いタスクに応用可能な多数決方法である。

キーワード
素性空間

語義, 曖昧性解消, 独立成分分析, サポートベクターマシン, ラベル無しデータ,

Word Sense Disambiguation based on Feature Space Restructuring

Hiroya Takamura Hiroyasu Yamada Taku Kudoh
Kaoru Yamamoto Yuji Matsumoto

Nara Institute of Science and Technology
Graduate School of Information Science
8916-5 Takamaya, Ikoma, Nara 630-0101, JAPAN
+81-743-72-5246, 5248

Abstract

We propose a new method for Word Sense Disambiguation. In the proposed method, weighted majority voting method is applied for the several restructured feature space computed by Independent Component Analysis (ICA) or Principal Component Analysis (PCA). We also conducted some experiments with SENSEVAL data. This voting method is not specific to WSD and can be applied to various tasks.

key words

word sense, disambiguation, Independent Component Analysis, Support Vector Machines, unlabeled data, feature space

1 序論

語義の曖昧性解消 (WSD) は、多義語に対し、その語が使われているコンテキストからその語義を決定するタスクである。本来ならばそこには世界知識の導入が不可欠である。しかし、文脈や世界知識を正確に記述することは既存の技術では不可能であり、なんらかの近似が必要とされる。特に計算機を用いて曖昧性の解消を行う場合は、大幅な近似が避けられない。

文脈の近似としては、多義語の周辺における他の語の出現などがよく使われる。もちろん、語そのものだけでなくある種の一般化が行われる場合も多い。例えば、意味カテゴリーや品詞などの出現を扱うという方法がある。世界知識の記述の近似はさらに実現しにくく、辞書の活用という形で主に近似される。

さて、本稿では文脈に注目する。one sense per discourse という言葉が (Yarowsky, 1995) などにおいて使われたが、この言葉は文脈のある側面に着目しているといえる。つまり、「同じ話題においては多義語はある決まった意味を持つ」という主張であり、これは話題という文脈と多義語の意味の関係を言い表している。このような側面から捉えると、WSD というタスクは文書分類とかなり近い性質を持つことが窺える。そこで我々は (Takamura and Matsumoto, 2001) で提案された文書分類のための素性空間再構成による方法を採用する。

しかし、WSD はそれほど単純ではない。bank のような明らかに異なる場面で使われる多義語ならば文書分類と同様に解決すると考えられるが、one sense per collocation と言い表されるケースのように高度な曖昧性を持つ多義語に関してはそうはいかない。そこで、我々はローカルな文脈にも注目することにした。ローカルな文脈としては、直前直後に出現する単語や品詞を用いた。ローカル文脈に対し、上で述べた話題に対応する文脈をグローバル文脈と呼ぶことにする。

我々のアプローチでは、これら二種類の文脈を、独立成分分析 (Independent Component Analysis, ICA) や主成分分析 (Principal Component Analysis, PCA) を用いた素性空間再構成により結び付け、複数の素性空間を構築する。さらに、それらの重み付き多数決を採ることにより分類を行う。(Kudoh and Matsumoto, 2001) において、サポートベク

ターマシン (Support Vector Machine, SVM) の重み付き多数決がチャンキングに用いられた。ここでは、異なるチャンク表現を用いることにより個別の分類器が構築された。しかし、様々なタスクにおいてそのように適切な分類器が構築できるとは限らない。我々の提案手法は、タスクそのものに個別の分類器を構築するための複数の素性が存在しなくても、重み付き多数決に必要な分類器を独立成分分析や主成分分析を通して構成できるという汎用性を持つ。

また、提案手法の有効性を数値実験により示した。用いたデータは SENSEVAL-1998 の英語版である。我々の手法は精度において、SENSEVAL の最高値や (Pedersen, 2001) による最近の報告を大幅に上回った。

2 関連研究

SVM は WSD の世界にはまだそれほど応用されていないが、(福本, 2001) がある。これは、各単語の各語義について Cross-Validation により最適な素性セットを選択するという手法である。そこで使用された素性セットは統語的に分けられており、我々の使用する素性セットとは異なる。また、多数決の WSD への応用としては、(Pedersen, 2000) がある。ここでは、Naive Bayes Classifier をベース分類器として、異なる ウィンドウサイズを取るにより複数の分類器を構築し多数決を行った。(Agirre et al, 2000) では、複数のヒュースティックスのアンサンブルを WSD に適用している。

また、本稿では (Pedersen, 2001) を比較対照に使用した。これは、統計的に重要と考えられる bigram を使って決定木により分類を行っている。

3 基盤となる手法

3.1 サポートベクターマシン

サポートベクターマシン (SVM) は、Large-Margin Classifier の一種である。与えられた素性ベクトルとラベルのペアの集合、

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n) \quad (1) \\ \forall i, \mathbf{x}_i \in \mathbf{R}^d, y_i \in \{-1, 1\}$$

に対して、SVM はマージン (分離平面とベクトルとの距離) が最大になるような分離平面を構成する (図 1):

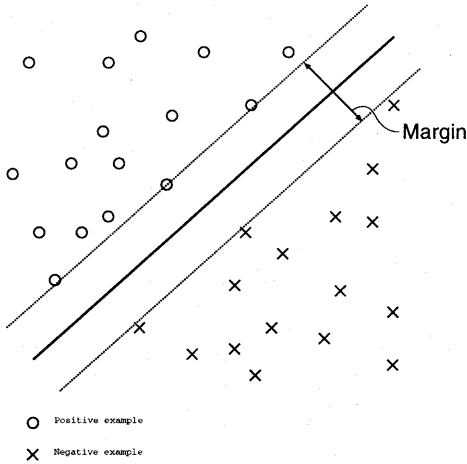


図 1: Support Vector Machine (実線が最適分離平面を表す).

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b. \quad (2)$$

最大マージンを達成することはノルム $\|\mathbf{w}\|$ を最小にすることと同値になる. この問題は次のように書ける.

$$\begin{aligned} \min. \quad & \frac{1}{2} \|\mathbf{w}\|^2, \\ \text{s.t.} \quad & \forall i, y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0. \end{aligned} \quad (3)$$

この問題の解は次の双対問題を解くことによって得られる:

$$\begin{aligned} \max. \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0, \\ & \forall i, \alpha_i \geq 0. \end{aligned} \quad (4)$$

ここで α_i 's はラグランジュ乗数である. (4) を最大にする α_i 's を用いて, 最適な $\mathbf{w}$ は,

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i. \quad (5)$$

と表される. (5) を (2) に代入することにより,

$$f(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b. \quad (6)$$

を得る.

テスト事例は (6) の符合に従って分類される.

SVM は二値分類器であるので, 直接的に多値分類へ応用することができない. このような場合, SVM に限らず一般的に one-versus-rest 方式, または pairwise 方式が用いられる. 前者は, 各クラスについてそのクラスかそれ以外かを分類することにより, 後者は任意の 2 クラスの組合せについて分類を行うことにより, それぞれ多値へ拡張する方法である.

pairwise 方式ではクラス数の増加に伴って構築すべき二値分類器がクラス数の二乗のオーダーで増えるのに対し, one-versus-rest 方式では一乗のオーダーである. WSD ではクラス数が数十に及ぶことがあるという事実を踏まえ, 我々は前者の one-versus-rest 方式を採用した. 各クラスについて分離平面が求められるので, 各事例に関して, それら分離平面との距離が最も大きくなるようなクラスを予測するものとする.

3.2 独立成分分析

独立成分分析 (ICA) は, 混合信号から独立信号源を抽出する方法である. ICA は PCA と類似した手法で, PCA においては変換行列が直行行列であったのに対し, ICA ではそのような制限はない.

計算は独立性を測る目的関数を用い, 最急勾配法によって行われる. 独立性の尺度としていくつもの関数が提案されているが, ここでは情報量最大化基準を用いる (Bell and Sejnowski, 1995). 学習規則は自然勾配 (Amari, 1998) を考慮して算出されたものを使用する.

3.3 素性空間再構成

まず, PCA もしくは ICA を用いて各ベクトル $\mathbf{d}$ の次元を圧縮する. どちらの場合も,

$$\mathbf{d} = \mathbf{A}\mathbf{s}. \quad (7)$$

と表現することができる. ここで $\mathbf{A}$ は変換行列, $\mathbf{s}$ は圧縮されたベクトルである.

次に元のベクトル $\mathbf{d}$ と圧縮されたベクトル $\mathbf{s}$ を連結 (concatenate) する:

$$\hat{\mathbf{d}} = \begin{bmatrix} \mathbf{d} \\ \mathbf{s} \end{bmatrix}. \quad (8)$$

$\hat{\mathbf{d}}$ を入力として SVM により分類を行う. このような変形は, 元のベクトルによる情報を失わずに, 圧縮空間に重みを与えることを可能にする.

4 素性空間再構成を基にした重み付き多数決

4.1 基本素性

まず、基本となる素性について述べる。我々が使用した素性はローカルな素性とグローバルな素性に分けられる。具体的には以下の通りである。

- ローカルな素性
 1. 多義語の直前2単語
 2. 多義語の直前2単語の品詞
 3. 多義語の直後2単語
 4. 多義語の直後2単語の品詞
 5. 多義語自身の品詞
- グローバルな素性
 1. 多義語の直前50単語
 2. 多義語の直後50単語

4.2 素性空間の選択

WSDに効果的な素性空間は各多義語によって異なる((福本, 2001)によれば各語義について異なるが、ここではそれは考慮しない)。この相違を超越する素性空間を作るのは非常に困難である。そこで我々は複数の素性空間を構成し、各々の素性による分類の重み付き多数決を採ることとする。しかし、用いた素性セットによっては分類精度が非常に低くなる可能性があり、それらも含めて多数決を行うことは分類精度の低下につながる。よって各々の分類器の分類精度をある程度高いものに保つ必要がある。この要請と、ローカルの素性単独の分類精度が高いという実験的事実に基づいて、我々は以下の素性を用いてそれぞれの分類器を構築した。

1. 基本素性(ローカル, local)
2. local + globalの圧縮空間(PCA)
3. local + globalの圧縮空間(ICA)
4. local + globalの再構成空間(PCA)
5. local + globalの再構成空間(ICA)

それぞれの分類器の重みは、(Vapnik and Chapelle, 2000)で提案され、(Kudoh and Matsumoto, 2001)によって多数決における重みとしての有効性が示された LOO-bound を用いる。LOO-bound は SVM に対し Leave-One-Out Estimation を行ったときの

エラーの期待値の上限であり、以下のように定義される：

$$\text{LOO-bound} = \frac{\text{サポートベクターの数}}{\text{訓練事例の数}}. \quad (9)$$

各語義について LOO-bound が求められるので、その平均を用いて、

$$(\text{重み}) = 1 - (\text{LOO-bound の平均}), \quad (10)$$

と重みを決定する。

実は、(Kudoh and Matsumoto, 2001)において LOO-bound より VC-bound の方が重みとしての有効性が高いことが示されているが、ICA や PCA を行うことにより VC-bound の計算誤差が大きくなることが予想されるので、今回は LOO-bound を使用することにした。

5 実験

使用したデータは SENSEVAL-1998 の英語版¹である。SENSEVAL は WSD のコンペティションであり、(Kilgarriff and Rosenzweig, 2000)によると辞書編集の専門家によってタグ付けされており、しかもチェック体制が整備されているため、高度な一貫性を持つ優れたデータが作成されていると言われている。そのような理由から、我々は SENSEVAL の公開データを使用した。ターゲットとなる単語は 36 語で、名詞(n)、動詞(v)、形容詞(a)、さらに複数の異なる品詞を取り得る単語(p)の三つに分かれる。それぞれの単語の語義数や、訓練事例とテスト事例の数などの詳細は結果と共に表1に示した。評価方法は fine-grained²で行った。また、答えのラベルが disjunction になっているものについては、そのどちらを予測しても正解に数えた。

前処理では、TreeTagger (Schmid, 1994)を用いて stemming 及び品詞解析を行った。素性として品詞を入れる場合はこの解析結果を用いた。SVM の計算には TinySVM³を使用した。用いたカーネル関数は線形カーネルで

¹<http://www.itri.brighton.ac.uk/events/senseval/>より入手可能。

²意味ラベルが階層的になっているが、完全に一致した場合のみ正解とする。詳しくは、<http://www.itri.brighton.ac.uk/events/senseval/ARCHIVE-RESULTS/senseval.html>を参照して頂きたい。

³<http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/>より入手可能。

ある。抽出する独立成分あるいは主成分の数は50に固定した。

5.1 他手法との比較

ここでは、他の手法と分類精度を比較した。結果は表1に示した。比較したのは、以下の分類器である。提案手法以外の精度は、(Pedersen, 2001)より抜粋した。

- maj: 訓練事例で最も出現頻度が大きい語義を予測するもの。ベースライン。
- best: SENSEVALで最も値がよかったもの。最高一致率と最高復元率を元に算出されており、両者が違う参加者の値でありうるため、実際の値はこれより低い可能性がある。
- avg: SENSEVALにおける精度の平均。
- PJ: (Pedersen, 2001)による最新の報告。bigramのうち、統計的に重要と思われるものを利用したモデル。
- NB: (Pedersen, 2000)で提案されたモデル。Naive Bayes 分類器のアンサンブルによるもの。
- FSR: 提案手法。ここでは節4.2に示した素性セット1, 3, 5の多数決である。

また、表1において、test はテスト事例の数、# は取り得る語義の数、tr は訓練事例の数を示す。平均 Avg では、各単語の精度をテスト事例の数を重みにして計算した。表1中で最も下に位置するセルでは、提案手法が他の手法より高い精度だった単語数 (win)、同じ精度だった単語数 (tie)、低い精度だった単語数 (lose) をそれぞれ示してある。

5.2 多数決の効果

ここでは、重み付き多数決による効果がどれくらいあったのかを見てみる。結果は表2に示した。列1,2,3,4,5はそれぞれ節4.2で示した素性によるベース分類器単独の精度である。また右側の1-5などは、対応するベース分類器による多数決の結果である。各行は各単語に相当するが、スペースの関係上、単語そのものは記載していない。単語の順番は表1と同じなのでそちらを参照して頂きたい。一番下の行にはそれぞれの平均値を記載した。

5.3 考察

他の手法との比較(表1)では、平均において他の手法より10%以上良い精度を出している。win-tie-loseにおいても、全ての他手法

に対して9割近くが提案手法の win という結果になっている。訓練事例数が小さいときは、best の精度が上回ることがしばしばあるが、これはそのようなケースでは精度の分散が大きくなるので最も良い数値が選ばれている best が有利であるからであろう。

しかし、表2によると、精度の良さのほとんどはベース分類器であるSVMの性能に依るものであることがわかる。重み付き多数決による精度向上は0.5%に留まる。それでも多数決したものはその組合せに関わらず、僅かながら平均精度が向上している。また、組合せ1,3,5が最もよかったという結果は、独立成分分析のWSDにおける可能性を示唆するものである。もちろん現段階では精度向上が有為であるとは言いきれないので、さらなる実験が必要である。

6 結論

語義の曖昧性解消問題のための新しい手法を提案した。提案手法では、ICA や PCA を用いて複数の素性空間を再構成し、各々の空間における曖昧性解消結果を重み付き多数決を採った。ベース分類器としてSVMを用い、手法の有効性をSENSEVALのデータを用いて実験的に示した。精度向上はSVMに依っている部分が大いとはいえ、重み付き多数決による精度向上も確認された。

また、提案した複数の分類器の構成方法は、タスクに関わらず適用可能である。

今後の発展としては、さらなる実験による有効性の裏付けや、別のタスクへの適用などがある。また、辞書知識などの有効利用などが課題である。実際、今回我々が使用した情報以外に辞書の知識や複合語情報を利用した(Veenstra et al, 1998) などでは我々のものより高い平均精度を報告している。

理論的な面では、多数決の際の重みの決定、より適切な素性空間の構成などの問題がある。

References

- Agirre, E., Atserias, J., Padró, L. and Rigau, G. 2000. Combining Supervised and Unsupervised Lexical Knowledge Methods for Word Sense, Disambiguation Computers and the Humanities. *Special Double Issue on SensEval*, Eds. Martha Palmer and Adam Kilgarrieff. 34:1,2.

- Amari, S. 1998. Natural Gradient Works Efficiently in Learning. *Neural Computation*, vol. 10-2, pp. 251-276.
- Bell, A. J. and Sejnowski, T. J. 1995. An Information Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7, 1129-1159.
- 福本 文代. 2001. 語義の曖昧性解消のための最適な属性選択. 情報処理学会自然言語処理研究報告, 2001-NL-141, pp. 69-76.
- Kilgariff, A. and Rosenzweig, J. 2000. English Senseval: Report and Results. Second International Conference on Language Resources and Evaluation (LREC-2000).
- Kudoh, T. and Matsumoto, Y. 2001. Chunking with Support Vector Machines *Proceedings of the Second Meeting of North American Chapter of Association for Computational Linguistics (NAACL)*, pp.192-199, Pittsburgh.
- Pedersen, T. 2000. A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 63-69, Seattle.
- Pedersen, T. 2001. A Decision Tree of Bigrams is an Accurate Predictor of Word Sense. *Proceedings of the Second Meeting of North American Chapter of Association for Computational Linguistics (NAACL)*, pp.79-86, Pittsburgh.
- Schmid, H. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*, pp. 44-49, Manchester.
- Takamura, H. and Matsumoto, Y. 2001. *Proceedings of 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 51-57, Pittsburgh.
- Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Vapnik, V and Chapelle, O. 2000. Bounds on error expectation for SVM. *Advances in Large Margin Classifiers*. pages 261-280, Cambridge, MA, MIT Press.
- Veenstra, J., Van den Bosch, A., Buchholz, S., Daelemans, W. and Zavrel, J. 1998. Memory-based word sense disambiguation. Computing and the Humanities, Special issue on SENSEVAL.
- Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, MA, pp. 189-196.

表 1: 他手法との精度の比較

word	test	#	tr	maj	best	avg	PJ	NB	FSR
accident-n	267	8	1234	75.3	87.1	79.6	85.0	83.1	92.5
behaviour-n	279	3	994	94.3	92.9	90.2	95.7	93.2	96.1
bet-n	274	15	106	18.2	50.7	39.6	41.8	39.3	70.0
excess-n	186	8	251	1.1	75.9	63.7	65.1	64.5	88.1
float-n	75	12	61	45.3	66.1	45.0	52.0	56.0	60.0
giant-n	118	7	355	49.2	67.6	56.6	68.6	70.3	70.3
knee-n	251	22	435	48.2	67.4	56.0	71.3	64.1	83.7
onion-n	214	4	26	82.7	84.8	75.7	82.7	82.2	84.6
promise-n	113	8	845	62.8	75.2	56.9	48.7	78.0	84.1
sack-n	82	7	97	50.0	77.1	59.3	80.5	74.4	90.2
scrap-n	156	14	27	41.7	51.6	35.1	26.3	26.7	41.7
shirt-n	184	8	533	43.5	77.4	59.8	46.7	60.9	62.5
amaze-v	70	1	316	0.0	100	92.4	58.6	71.4	100
bet-v	117	9	60	43.2	60.5	44.0	50.8	58.5	74.4
bother-v	209	8	294	75.0	59.2	50.7	69.9	62.2	82.3
bury-v	201	14	272	38.3	32.7	22.9	48.8	42.3	47.8
calculate-v	218	5	249	83.9	85.0	75.5	90.8	80.7	91.3
consume-v	185	6	67	39.8	25.2	20.2	36.0	31.7	45.4
derive-v	217	6	259	47.9	44.1	36.0	82.5	72.4	65.9
float-v	228	16	183	33.2	30.8	22.5	30.1	56.3	50.0
invade-v	207	6	64	40.1	30.9	25.5	28.0	31.0	51.7
promise-v	224	6	1160	85.7	82.1	74.6	85.7	85.3	92.0
sack-v	178	3	185	97.8	95.6	95.6	97.8	97.2	98.3
scrap-v	186	3	30	85.5	80.6	68.6	85.5	82.3	87.1
seize-v	259	11	291	21.2	51.0	42.1	52.9	51.7	63.7
brilliant-a	229	10	442	45.9	31.7	26.5	55.9	58.1	53.7
floating-a	47	5	41	57.4	49.3	27.4	57.4	55.3	61.7
generous-a	227	6	307	28.2	37.5	30.9	44.9	48.9	52.0
giant-a	97	5	302	94.8	98.0	93.5	95.9	94.8	97.9
modest-a	270	9	374	61.5	49.6	44.9	72.2	68.1	66.3
slight-a	218	6	385	91.3	92.7	81.4	91.3	91.3	95.0
wooden-a	196	4	362	93.9	81.7	71.3	96.9	93.9	94.4
band-p	302	29	1326	77.2	81.7	75.9	86.1	83.1	90.4
bitter-p	372	14	144	27.0	44.6	39.8	36.4	32.6	55.4
sanction-p	431	7	96	57.5	74.8	62.4	57.5	56.8	80.5
shake-p	356	36	963	23.6	56.7	47.1	52.2	46.6	73.3
Avrg				54.4	63.9	54.9	64.4	63.9	74.5
win				35	30	36	31	31	-
tie				1	1	0	0	1	-
lose				0	5	0	5	4	-

表 2: 多数決による精度の変化

1	2	3	4	5	1-5	1,2,3	1,4,5	1,3,5
92.5(0.14)	92.8(0.14)	92.5(0.14)	89.5(0.27)	89.5(0.27)	92.5	92.8	92.5	92.5
95.6(0.15)	95.3(0.14)	95.3(0.14)	94.6(0.22)	94.6(0.22)	95.3	95.3	96.1	96.1
68.2(0.37)	68.6(0.38)	68.2(0.39)	59.1(0.60)	59.4(0.60)	70.8	68.9	70.1	70.0
87.0(0.23)	87.0(0.25)	87.0(0.26)	85.4(0.54)	84.9(0.54)	89.2	87.0	87.6	88.1
53.3(0.57)	54.6(0.59)	56.0(0.63)	70.6(0.71)	72.0(0.71)	61.3	54.6	58.7	60.0
70.3(0.29)	73.7(0.29)	71.1(0.28)	74.5(0.45)	74.5(0.45)	69.4	72.8	70.3	70.3
82.4(0.13)	80.8(0.14)	80.0(0.14)	76.8(0.27)	76.8(0.26)	81.2	82.4	83.3	83.7
84.5(0.72)	84.5(0.68)	84.5(0.68)	84.5(0.86)	84.5(0.86)	84.5	84.5	84.6	84.6
79.6(0.24)	78.7(0.24)	79.6(0.23)	77.8(0.38)	77.8(0.39)	84.9	81.4	85.0	84.1
89.0(0.33)	87.8(0.36)	89.0(0.38)	80.4(0.60)	80.4(0.60)	90.2	89.0	90.2	90.2
47.4(0.63)	47.4(0.61)	48.0(0.63)	38.4(0.80)	38.4(0.80)	40.3	48.0	44.9	41.7
58.1(0.29)	59.7(0.29)	56.5(0.29)	59.2(0.40)	58.6(0.40)	59.7	58.1	60.9	62.5
100(0.31)	100(0.33)	100(0.35)	100(0.67)	100(0.67)	100	100	100	100
76.0(0.38)	76.0(0.40)	75.2(0.42)	70.0(0.58)	69.2(0.58)	75.2	75.2	75.2	74.4
81.3(0.21)	81.3(0.21)	81.8(0.21)	74.1(0.39)	74.1(0.39)	81.8	80.8	82.8	82.3
46.7(0.29)	47.2(0.30)	48.2(0.30)	49.7(0.44)	50.2(0.44)	48.7	47.7	47.3	47.8
89.4(0.24)	90.3(0.26)	90.3(0.26)	89.9(0.40)	89.9(0.40)	91.7	90.3	90.8	91.3
48.6(0.60)	49.1(0.62)	45.9(0.60)	42.1(0.75)	42.1(0.75)	45.9	48.1	44.3	45.4
62.2(0.41)	61.2(0.40)	65.4(0.39)	65.4(0.59)	65.4(0.59)	66.3	61.2	64.5	65.9
47.3(0.33)	50.8(0.34)	49.1(0.34)	47.8(0.48)	48.2(0.47)	48.6	50.0	50.9	50.0
48.3(0.56)	53.1(0.61)	53.1(0.63)	48.7(0.85)	48.7(0.85)	50.2	50.7	50.7	51.7
93.3(0.15)	91.9(0.14)	92.4(0.14)	90.1(0.23)	90.1(0.23)	91.5	92.8	92.0	92.0
98.3(0.17)	98.3(0.16)	98.3(0.16)	97.7(0.17)	97.7(0.17)	97.7	98.3	98.3	98.3
83.8(0.53)	85.4(0.53)	85.4(0.57)	87.6(0.80)	87.6(0.80)	87.0	85.4	86.6	87.1
62.5(0.30)	60.6(0.30)	58.6(0.31)	56.7(0.47)	56.7(0.47)	64.0	61.0	62.9	63.7
53.2(0.34)	52.4(0.34)	47.5(0.33)	47.5(0.44)	47.1(0.44)	50.6	52.4	53.7	53.7
63.8(0.68)	65.9(0.72)	63.8(0.72)	59.5(0.84)	59.5(0.84)	63.8	65.9	66.0	61.7
47.5(0.55)	50.2(0.54)	51.5(0.53)	51.9(0.66)	52.8(0.66)	53.7	50.2	51.5	52.0
97.9(0.24)	97.9(0.23)	97.9(0.24)	98.9(0.37)	98.9(0.37)	98.9	97.9	97.9	97.9
67.7(0.30)	65.1(0.30)	64.8(0.29)	66.6(0.41)	66.2(0.41)	68.8	67.4	65.6	66.3
94.4(0.18)	95.4(0.18)	94.9(0.18)	94.0(0.36)	94.0(0.36)	94.0	94.9	95.0	95.0
95.4(0.14)	95.4(0.14)	94.8(0.14)	93.8(0.23)	93.8(0.23)	94.3	95.4	94.4	94.4
91.7(0.09)	91.0(0.09)	89.7(0.08)	86.0(0.15)	86.0(0.15)	89.4	91.3	90.1	90.4
55.6(0.37)	56.4(0.38)	54.3(0.38)	46.2(0.50)	46.2(0.50)	54.8	56.7	56.5	55.4
79.1(0.48)	79.8(0.50)	80.0(0.52)	74.2(0.71)	74.2(0.71)	80.0	79.3	80.3	80.5
73.0(0.09)	73.0(0.09)	72.4(0.09)	69.3(0.17)	69.3(0.17)	73.0	72.7	74.2	73.3
73.7	74.0	73.5	71.2	71.2	74.2	74.1	74.4	74.5