ファジィc-平均識別器の訓練時間の改善

1　はじめに

本論文では，識別器のハイパーパラメータを最適化するまでの時間を訓練時間とし，訓練時間の高速化を図ったファジィc-平均識別器 (FCMC) [1, 2, 3, 4, 5, 6] とサポートベクターマシン（SVM) [8, 9, 10, 11] との数値実験による比較結果を報告する．

FCMC はガウス混合モデル [2, 7] による識別器のように，判別のクラス毎にデータをファジィクラスターに分け，それらのメンバシップ関数の足し合わせで識別しようとするもので，比較的少量のデータの場合は，SVM にも劣らない識別性能が報告されている [3, 12]. FCMC は駐車場の車両検知へ実際に応用され [4]，すでに複数の駐車場で稼働している．そこで明らかになったのは訓練データが多ければ多いほどテスト性能が改善されることである．そのためには，大量データを扱えることが重要である．この駐車場管理システムは，現場でデータを追加して計算（再訓練）する必要があり，大量データでの訓練時間を短縮することの意味は大きい．そこで，本論文では大量訓練データに対応できるように改良した FCMC の訓練時間を SVM と比較する．

SVM は，カーネル関数を用いてパターンを有限もしくは無限次元の特徴空間へ写像し，特徴空間上で線形分離を行うことで，高精度な非線型判別を可能にした識別器である．しかし，SVM の計算オーダーは訓練データ数の 2 乗から 3 乗で，大量データでの計算時間が長いことが欠点である．SVM に SMO（Sequential Minimum Optimization) 法 [13] を実装した LibSVM [14, 15] は訓練データが大量にある場合にも高速に計算できるように改良されているが，それでも，100 万件を超えるような大量データでは実用的な時間内に訓練を終了させることができない場合が多い．

訓練データが大量で，テストデータも大量にある場合の識別器の性能は一般にテストデータの識別精度で評価され，SVM のカーネルパラメータ g や正則化パラメータ C などのハイパーパラメータはテストデータの識別精度が良くなるように選ばれる [16, 17, 18, 19, 20, 21]. 交差確認法（CV 法）の評価用データでパラメータを最適化するには，その分割数倍の時間がかかる．データが大量であれば計算時間がかかりすぎるのと，十分な数のテストデータがあればその精度を一般的な性能評価値として用いることができるためである．したがって，SVM の訓練時間は 2 次計画法などでの訓練データに対する識別率の最適化のための計算時間とされているのが一般的である．しかし，テストデータに対して最適なハイパーパラメータを選ぶためには，テストデータの識別率も計算する必要があり，パラメータ値を変更しながら訓練とテストを何度も繰り返す必要がある．そこで，本論文では FCMC と LibSVM の訓練時間をハイパーパラメータ最適化のための繰り返し計算を含めて比較する．ハイパーパラメータは，テストデータまたは評価用データに対して最適化される少数のパラメータで，訓練データに対して最適化される変数ではない．

SVM のパラメータ最適化にはグリッドサーチが良く用いられ，FCMC では粒子群最適化法 (PSO)[22, 23, 24] を用いている場合もあるが，パラメータ探索の方法によって計算時間は変わり，またグリッドサーチの刻み幅や PSO のランダムネスによって最適値が変動するので公平な比較は困難である．また，100 万件以上の大量訓練データを対象とする場合は，SVM では実行可能な時間内に終了しない場合が多い．最適化手法によっても時間は異なるので，LibSVM の総訓練時間は少ない反復（評価）回数での時間から推定した 50 回の訓練時間とする．LibSVM のパラメータ（C, g）を自動で最適化するのに 50 回では少ないと考えられるが，訓練時間の比較を明確にするために LibSVM では 50 回で最適化できるものとする．

FCMC の訓練時間の改善として，次の四つのことを行った．1) PCA と 2 分木を用いるクラスタリングの初期化（初期分割）でのクラスター中心と共分散行列の計算には，訓練データを $s = 2^{\log_2 c - i} - 1$ 個飛びに用いる．ただし，$i = 1, ..., \log_2 c$ で，c はクラスター数，$\log_2 c$ は 2 分木の深さである．2) データを複数個のブロックに分割して読み込み，更新されたメンバシップの中間結果もそのブロック毎にハードディスクに書き出す．3) クラスタリングの反復（更新）回数は 1 回のみとする，4) サンプリングした少ないテストデータでランダムサーチを行い，得られた解の近傍で全テストデータでの探索を行う．

3 章で報告する FCMC の結果は，本論文の提案アル

ゴリズムでの最適化の結果であるが，LibSVM の最適パラメータでの識別精度は，文献で報告されている最適値を参考に多くの試行を繰り返して知り得た，テストデータでの精度がもっとも良かった場合である．

文献 [25, 26] は，訓練データ数が変化したときの性能比較を報告したもので，FCMC の多数のパラメータを訓練データに対して最適化した場合のテストデータでの識別精度と訓練時間を報告している．本論文では，クラスター数をさらに多くして，テストデータに対して最適化するハイパーパラメータを，FCMC は四つ，LibSVM は二つとする．パラメータ数は多くするほど識別精度を良くすることができるが，最適化のための時間がかかる．たとえば，LibSVM ではカーネル関数のパラメータを増やしたり，カーネル関数そのものを幾通りか用いれば精度は改善される．しかし，最適化の時間は非常に長くなり大量データの場合は不可能である．FCMC では訓練時間が短いので，クラスター数と共分散行列の低階数近似に用いる基底ベクトル数もパラメータとする．そして，これらのパラメータ値は二三通りから選べば良いことを，それぞれの計算結果から示す．

FCMC の訓練時間は，大量データの場合には LibSVM に比べて百倍から千倍短く，実用的な時間内にパラメータの最適化までを含んだ訓練が終了する．

2 FCM 識別器

FCM 識別器 [3, 4, 5, 6] は 2 つのフェーズから成り，第 1 フェーズではクラス毎に c 個のクラスターにクラスタリングし，第 2 フェーズではハイパーパラメータの最適化を行う．ハイパーパラメータは，テストデータまたは評価用データに対して最適化され，本論文では以下で述べる四つのパラメータ (m, γ, c, r) とする．

第一フェーズのクラスタリングには，マハラノビス距離を用いて一般化したハードクラスタリング法 [2, 3, 27] を用いる．ただし，クラスタリングの更新アルゴリズムはその反復回数を 1 回のみとする．訓練データのクラスター中心までの距離の計算はクラスタリングのアルゴリズムに含まれているが，テストデータについての距離はクラスタリング終了時点に一度だけ計算される．

$$D(x_k, v_i; S_i) = (x_k - v_i)^\top S_i^{-1}(x_k - v_i) \tag{1}$$

はデータベクトル $x_k \in \mathcal{R}^p$ とクラスター中心ベクトル $v_i \in \mathcal{R}^p$ のマハラノビス距離で，N をデータ件数とすると，次の S_i はクラスター i の共分散行列である．

$$S_i = \frac{\sum_{k=1}^N u_{ki}(x_k - v_i)(x_k - v_i)^\top}{\sum_{k=1}^N u_{ki}} \tag{2}$$

$$v_i = \frac{\sum_{k=1}^N u_{ki} x_k}{\sum_{k=1}^N u_{ki}} \tag{3}$$

はクラスター中心で，クラスターの混合比率 (mixing proportion) は

$$\alpha_i = \frac{\sum_{k=1}^N u_{ki}}{\sum_{j=1}^c \sum_{k=1}^N u_{kj}} = \frac{1}{N}\sum_{k=1}^N u_{ki} \tag{4}$$

となる．S_i が特異行列になりにくいように確率的主成分分析における共分散行列の低階数近似法 [28, 29] を用いる．クラスタリング結果が初期値に依存しないように，クラス毎のデータの主成分スコアーの符号（+, -）によってクラス毎のデータを二つのクラスターに分割する．この操作をクラス毎に，指定したクラスター数になるまで繰り返して初期クラスターを決定する（完全 2 分木）．クラスター数はクラス毎に指定可能であるが，本論文では同数とする．初期化にランダムネスはないので，クラスタリングは決定論的である．また，クラスター数は 2 のべき乗個である．クラスタリングの反復回数は 1 とし，高次元データは 50 次元に圧縮して用いるので，共分散行列，クラスター中心，中心からデータ点への距離の計算オーダーは，全て $O(c \times N)$ である．

第 2 フェーズでは，テストデータのクラスター中心までの距離を，第 1 フェーズで求めたクラスター中心とクラスター毎の共分散行列を用いて式 (1) から求める．そして，式 (6) のメンバシップ値の大小から誤識別率を求め，それを最小化するようにパラメータ m と γ の値を探索する．したがって，式 (1) の距離の計算は一度行うだけであるが，誤識別率はパラメータをランダムに与えて何度も計算することになる．パラメータの最適化には PSO などを用いることができるが [3, 4, 5, 6]，本論文ではランダムサーチとする．

π_q をクラス q の混合比率, すなわちクラス q の事前確率とする．x_k のクラス q へのメンバシップを次のように定める．

$$u^*_{qjk} = \alpha_{qj}|S_{qj}|^{-\frac{1}{\gamma}}(D(x_k, v_{qj}; S_{qj}) + \nu)^{-\frac{1}{m}} \tag{5}$$

$$\tilde{u}_{qk} = \frac{\pi_q \sum_{j=1}^c u^*_{qjk}}{\sum_{s=1}^Q \pi_s \sum_{j=1}^c u^*_{sjk}} \tag{6}$$

c はクラス毎のクラスター数であり, Q はクラス数である．ハイパーパラメータ m と γ を事前に指定した区間内の一様乱数で与えテストデータの誤識別率を最小化する値を選択する．クラスタリングアルゴリズムの導出やメンバシップ関数の設定についての詳細は文献

[3, 12] を参照いただきたい．第 2 フェーズでの主な計算は，各テストデータの各クラスターへのメンバシップの計算であり，計算オーダーは探索回数が一定であるので，テストデータ数を n とすると $O(c \times n)$ である．従って，テストデータを減らせば計算時間を減らすことができる．下記の 3) はこのことから行った改善である．

大量データに対応した改善点は以下の四つである．
1) クラスタリングの初期分割（2 分木）でのクラスター中心と共分散行列の計算には，訓練データを $s = 2^{\log_2 c - i} - 1$ 個飛びに $i = 1, ..., \log_2 c$ として用いる．ただし，c はクラスター数で，$\log_2 c$ は 2 分木の深さである．従って，$i = 1$ のときは，$2/c$ の訓練データを用いて，最後の $i = \log_2 c$ のときは，全ての訓練データを用いる．
2) 訓練データをクラス毎に小さなブロックに分けて読み込むようにし，メンバシップなどの中間的な計算結果をブロック毎にハードディスクに書きだす．このことで，計算機のメモリー不足を解消し，かつ仮想メモリーを用いずに計算可能とする．
3) クラスタリングの更新式の反復回数は 1 回のみとする．すなわち，初期分割の後にクラスターの中心と共分散行列を求めておき，クラスタリングは，最も近いクラスターへのメンバシップを 1 とし，その他は 0 と更新することと，式 (3) と式 (2) の更新を順に 1 回ずつだけ行う．
4) パラメータ最適化はランダムに選んだ 100 件のテストデータを用いて，200 回のランダムサーチ（探索範囲は $m \in$[0, 2], $\gamma \in$[0, 20], $\nu = 5$）を行い，求めた m の候補とその近傍 ($m \pm 0.005$) の 3 か所を全テストデータを用いて探索する．ν は全てのベンチマークデータで 5 に固定した．以上の探索を 5 回反復しその中の誤識別率が最小となる m と γ を最適パラメータとする．従って，総探索回数は (200 回+3 回) × 5 回である．

LibSVM のハイパーパラメータは正則化のための C とガウシアンカーネルの g の二つである．FCMC は ν を 5 に固定し自動探索の対象は m, γ の二つのみとする．クラスター数 c と共分散行列の低階数近似のための基底ベクトル数 r もパラメータとするが，これらは二三通り試せば十分であることを次章で示す．

3 LibSVM との性能比較

本章では，いくつかの大量ベンチマークデータを用いてクラスター数に応じた FCMC の訓練時間と識別精度を LibSVM と比較する．表 1 に用いたベンチマークデータを示す．

全て連続の実数値データでカメラ画像と手書き数字の画像データ (USPS, MNIST) を典型的な対象とした．KDD データはカテゴリカルな特徴量を含むが，データ圧縮が非常に有効なケースでかつ大量データであるので取り上げた．文書データ [30] などの大量であるが線形判別が有効とされるデータやカテゴリカルデータは対象としなかった．

表の features は特徴量の次元数で，括弧で元の次元数，その前に PCA で圧縮したデータの次元数を示している．Parking データは，駐車場の車両検知システムのためのカメラ画像データ [6] である．その他のデータは http://www.cse.ust. hk/~ivor/cvm.html [19] からダウンロードしたものを用いた．

FCMC の計算機プログラムは MATLAB で作成した．LibSVM のプログラム [15] は http://www.csie.ntu.edu.tw/~cjlin/libsvm からダウンロードして用いた．カーネル関数はガウシアン $k(x, y) = \exp(-g|x - y|^2)$ を用いた．カーネルパラメータの g と正則化パラメータ C は，大量データでの SVM の性能評価が報告されている多くの文献 [16, 17, 18, 19, 20, 21] と同様にテストデータに対して最適な値を選択した．駐車場の画像データ以外は文献で最適な値が報告されているので，それを参考にして，マニュアルや乱数でパラメータ値を与えてテストを繰り返した結果，誤識別率が最小となったものを最適なパラメータ値とした．

FCMC のパラメータ探索の範囲は，少量データの場合の文献 [3] での $m \in$[0, 2], $\gamma \in$[0, 20] とし一様乱数で与えた．ν は全てのデータで 5 に固定した．LibSVM では最適値から外れたパラメータでは極端に時間がかかり終了しないと思われる場合があるので，データ毎に最適値の近辺を区間として指定し，ランダムに選んだパラメータ値での 10 回の計算時間を 5 倍して 50 回の訓練時間の推定値とした．取り上げたデータのみでも，C の最適値は 1 から 1,000,000 まであり，g も 0.1 から 5 まであるので，現実の新たなデータでは最適値の近辺に範囲を限定することはできず，50 回で最適化するのも困難と思われるが，比較を明確にするためにこのような設定にした．

表 2-7 に比較結果を示す．図 1-5 はクラスター数 c と共分散行列の低階数近似に用いる基底ベクトル数 r の識別精度に及ぼす影響を比較していて，c と r の組み合わせは多くの場合を試す必要がないことを示している．表の best hyper-parameter はパラメータの最適値を示している．LibSVM の最適値は，文献 [16, 17, 18,

表 2: Parking-C31 データでの訓練時間の比較

training sample **200,000**, test sample 70,000, feature dimension : **50**		
	FCMC (c=16)	LibSVM
best hyper-parameter	m=0.1929, γ=14.4254	C= 3.58, g= 5.40
test error	**0.51**%	**0.39**%
total training time	85.8+32.2+37.5=**155.5**s	estimated training time (50 times) **78531.0**s [21.8h]

	FCMC (c=32)	FCMC (c=64)
best hyper-parameter	m=0.1219, γ=8.5049	m=0.1913, γ=15.4249
test error	**0.31**%	**0.36**%
total training time	154.6+63.4+62.3=**280.3**s	293.1+127.7+113.6=**534.4**s [0.15h]

表 3: Parking-All データでの訓練時間の比較

training sample **900,000**, test sample 30,000, feature dimension : **50**		
	FCMC (c=64)	LibSVM
best hyper-parameter	m=0.1214, γ=7.9475	C=2.0, g=5.0
test error	**2.17**%	**1.92**%
total training time	1268.3+54.3+65.4=**1388.0**s	estimated training time (50 times) **1598300**s [444h]

	FCMC (c=128)	FCMC (c=256)
best hyper-parameter	m=0.1036, γ=6.4865	m= 0.0468, γ=2.7572
test error	**2.01**%	**1.91**%
total training time	2472.5+114.2+125.8=**2712.5**s	4948.3+258.3+246.1=**5452.7**s [1.5h]

表 1: ベンチマークデータ

data	features	training	testing
Parking-C31	50 (1,024)	200,000	70,000
Parking-All	50 (1,024)	899,853	30,000
USPS zero-one	50 (676)	266,079	75,383
KDD-CUP 1999	5 (127)	4,898,431	311,029
MNIST	50 (576)	60,000	10,000

19, 20, 21] で報告されているものを参考に，多くの試行を繰り返した中で知り得た，テストデータの識別精度が最も良い場合の値である．

total training time はパラメータ値をランダムに変更してテストを繰り返した時の訓練時間を含んだ総時間であり，たとえば，表 2 の FCMC(c=16) での 85.8+32.2 + 37.5 = **155.5**s は，その時間がクラスタリング

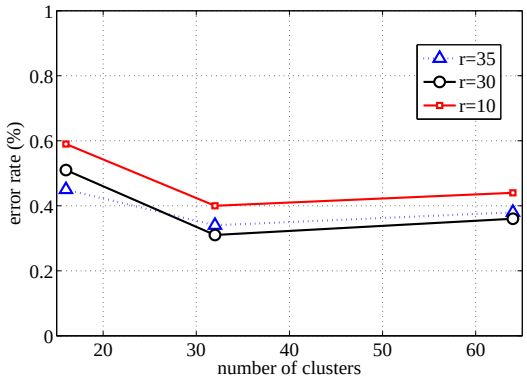

図 1: Parking-C31 データでのクラスター数と基底ベクトル数による識別精度の変化

の時間とテストデータの距離 $D(x_k, v_{qj}; S_{qj})$ の計算時間とテストデータでの探索（評価）を繰返した時間の和であることを示している．

用いた計算機は DELL T5400 3.16GHz, 4GB，OS

表 4: USPS zero-one データでの訓練時間の比較

training sample **266,079**, test sample 75,383, feature dimension : **50**		
	FCMC (c=4)	LibSVM
best hyper-parameter	m=0.1287, γ=6.5397	C=1.0, g=2.50
test error	**0.72**%	**0.46**%
total training time	35.3+8.6+20.1=**64**s	estimated training time (50 times) **28670.5**s [8.0h]

	FCMC (c=8)	FCMC (c=16)
best hyper-parameter	m=0.2059, γ=10.2247	m= 0.1291, γ=6.4942
test error	**0.42**%	**0.42**%
total training time	57.3+17.2+26.8=**101.3**s	98.3+34.4+40.2=**172.9**s

表 6: KDD-CUP データでの訓練時間の比較

training sample **4,898,431**, test sample 311,029, feature dimension : **5**		
	FCMC (c=2)	LibSVM
best hyper-parameter	m=1.2703, γ=10.9281	C=100, g=11
test error	**7.91**%	**5.24**%
total training time	94.5+8.0+60.0=**162.5**s	estimated training time (50 times) **3179560**s [883h]

	FCMC (c=4)	FCMC (c=8)
best hyper-parameter	m=1.1362, γ=18.5168	m= 0.8828, γ=4.2343
test error	**7.45**%	**5.36**%
total training time	149.5+15.9+70.2=**235.6**s	250.2+31.2+92.0=**373.4**s [0.1h]

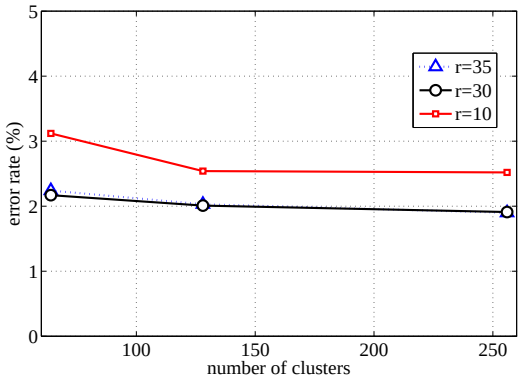

図 2: Parking-All データでのクラスター数と基底ベクトル数による識別精度の変化

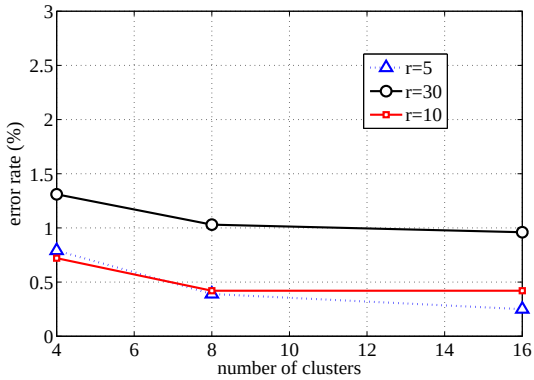

図 3: USPS データでのクラスター数と基底ベクトル数による識別精度の変化

は Windows Vista である.

3.1 Parking データ（駐車場の画像データ）

一つ目のベンチマークデータは駐車場に設置されたカメラの画像データで，屋外駐車場の満・空を判定するための車両検知システム [4, 6] に用いられたもので

表 7: MNIST データでの訓練時間の比較

training sample **60,000** test sample 10,000, feature dimension : **50**		
	FCMC (c=8)	LibSVM
best hyper-parameter	m=0.0606, γ=4.4303	C=4.44, g=0.82
test error	**0.94**%	**0.57**%
total training time	19.5+11.3+35.7=**66.5**s	estimated training time (50 times) **3498**s

	FCMC (c=16)	FCMC (c=32)
best hyper-parameter	m=0.1919, γ=11.8679	m= 0.1612, γ=9.2262
test error	**0.94**%	**0.75**%
total training time	34.7+22.5+61.8=**119**s	63.8+51.4+113.6=**228.8**s

表 5: KDD-CUP データでの LibSVM の結果

C	g	error rate	training time
1000000	0.7	6.37 %	51100 s
1000000	0.8	6.46 %	109230 s
100	11	5.24 %	8713 s
100	12	5.27 %	8103 s
100	30	7.54 %	140810 s
1	1	7.76 %	228310 s

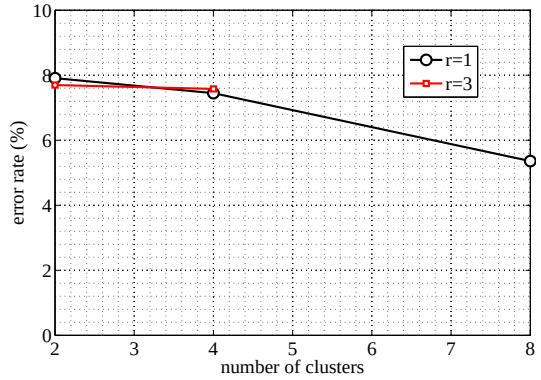

図 4: KDD-CUP データでのクラスター数と基底ベクトル数による識別精度の変化

ある．704 × 576 の JPEG フォーマットで保存されたカメラ画像は，車両や駐車スペースのサブウインドウに切り取られ，32×32×3 に間引かれる．その RGB 画像は HSV に変換されグレースケールの 32×32=1024 次元データは PCA (主成分分析，KL 展開) の 50 の基底ベクトルを掛ける （内積） ことで特徴量が 50 次元データに圧縮されている．

表 2 は，27 台の駐車スペースを前方上から撮影した誤識別の少ないカメラ画像での結果である．訓練データ数が 20 万件で余り多くないが，FCMC の総訓練時

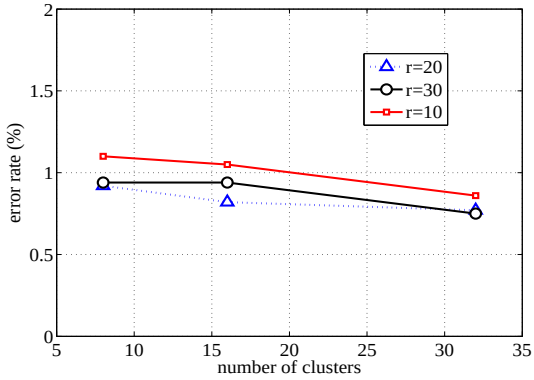

図 5: MNIST データでのクラスター数と基底ベクトル数による識別精度の変化

間はいずれの場合も，LibSVM での 50 回の訓練時間よりも短い．S_{qi} の低階数近似のための基底ベクトル数 r は 30 とした場合であるが，図 1 に示すように，r は 30 と 10 の場合を試せば十分で 35 の場合は 30 の場合とほぼ等しい．このことはクラスター数を変えても同じであるので，たとえば $c = 16$ の場合に二通りをテストすれば良い．また，クラスター数は少ない数から二倍ずつ増やして，たとえば 64 で改善されなくなれば 32 を採用する．その時の訓練時間 280.3 秒は，LibSVM の 50 回の訓練時間の 1 /280 である．表 2 の FCMC での三通りの c での訓練時間に r=10，c=16 での訓練時間を加えた総訓練時間は，LibSVM（50 回）の訓練時間の 1 /70 である．総訓練時間は表 2 に示すようにクラスター数にほぼ比例するので 8 個以下での計算時間を含めても総時間にほとんど影響しない．

LibSVM で，たとえば $C = 0.76, g = 1.58$ の時の 1 回の訓練時間は 684 秒で，表 2 での最適パラメータを用いた時は 1,330 秒で 2 倍であった．また，$C = 0.08, g = 8.93$ では，4,060 秒で 6 倍であった．一方，FCMC で

はクラスター数以外のハイパーパラメータは計算時間に影響しない．

表 3 に 3 箇所の屋上駐車場のカメラ画像を用いた結果を示す．三つの屋外駐車場のカメラ画像約 90 万件での結果である．駐車中の車両が前の車両に隠れて，誤識別が非常に多くなるデータである．大量データであるので，クラスター数を 2^8=256 まで増やした．Matlab のプログラムは，中間の計算結果をハードディスクに書き込んで更新を繰り返す方式に改善されているために，訓練データが 50 次元で 100 万件程であるが，クラスター数を 256 と大量にしても計算機メモリーの不足は発生しない．図 2 に示すようにクラスター数は 64 から 256 まで増やしているが精度はあまり変化していない．また，r は 10 と 30 では r=30 が良いのもクラスター数によらない．r=35 としてもほとんど変化していない．従って，c=128, r=30 を選べばよい．

訓練データ数が約 100 万件に増えているので，LibSVM（50 回）の総訓練時間の推定値は 444 時間（18.5 日），FCMC($c = 128$) での実測値 2712.5 秒（0.75 時間）は 1/589 である．LibSVM で現実的な時間内で訓練を終了させることはできないと思われる．表 3 の FCMC で，c の三通りの場合に r=10, c=64 での訓練時間を加えた総訓練時間は，LibSVM の 1/146 である．

3.2 USPS データ（0 と 1 の手書き文字）

このデータは USPS（アメリカ合衆国郵便公社）の 0 と 1 の手書き文字である．SVM の論文で取り上げられているもので，Tsang, Kwok, Cheung [19] により訓練データを 266,079 件，テストデータを 75,383 件に拡張されたベンチマークデータである (http: //www.kernel-machines.org/data/usps.mat.gz)．訓練データの部分集合から 50 の PCA 基底ベクトルを計算し，これを用いて各画像データを 50 次元の特徴ベクトルに変換した．

図 3 に示すように，S_{qi} の低階数近似のための基底ベクトル数は $r = 30$ より $r = 10$ の方が良いが，$r = 5$ としてもあまり変化がない．従って，$r = 30$ と $r = 10$ の二通りをテストすればよい．クラスター数は $c = 8$ と $c = 16$ では変化がないので $c = 8, r = 10$ を採用する．表 4 より，c=8 を採用した場合の精度は 0.42% で，総訓練時間は LibSVM（50 回）の 1/283 である．LibSVM で，たとえば C=2.0, g=4.0 とした場合の 1 回の訓練時間は 1717.6 秒で，表中の C=1.0, g=2.50 の場合は 875.5 秒で二倍以上の差がある．

また，表 4 の三通りの場合にさらに c=4，r=30 の場合の時間を加えた総訓練時間は 402.2(s) で LibSVM（50 回）の 1/71 である．訓練データ数が約 20 万件でそれほど多くないが，Parking-C31 データとほぼ等しい件数であり，LibSVM との訓練時間の差もほぼ等しい．

3.3 KDD データ（ネットワークへの侵入検知）

このデータは KDD1999 の学会 (Third International Knowledge Discovery and Data Mining Tools Competition) で使用された 127 次元 4,898,431 件のネットワーク侵入検知の KDD-CUP-99 データである．各データは 34 の実数値と 7 つのカテゴリカルデータからなる 7 週間分の TCP dump data である．(http://kdd.ics.uci.edu/databases/kdd cup99/kdd cup99.html)．他の 2 週間分の 311,029 件のデータがテスト用で 24 の攻撃タイプと訓練データにはない 14 のタイプがテストデータにのみ含まれている．正常な接続か攻撃であるかを識別することが問題である．

4,898,431 件の全ての訓練データを用いた．図 4 に示すように $r = 3$ でクラスター数を 8 以上にすると特異行列が発生したので c=8，$r = 1$ が最適であった．クラスター数を増やすと特異行列が発生するのは，大量データであるが狭い線形部分空間に分布しているためと考えられる．$c = 4$ 以下では r の値で変化はない．FCMC も LibSVM も PCA で 5 次元まで圧縮して用いた．圧縮のための PCA の基底ベクトルは最初の 300,000 件の訓練データから計算した．訓練データが大量であるので LibSVM の訓練時間は長くなっている．LibSVM で C=1,000,000（幾つかの文献で最適な値とされている）と g=0.7 とした時にテストデータの誤識別率が 6.37% となり文献 [16, 17, 18] などで報告されている LibSVM の誤識別率よりも小さくなった．さらに，C=100，g=11 とした時の誤識別率が 5.24% まで改善された．ただし，訓練時間は非常に長く，自動化によるパラメータ最適化は困難と思われる．表 5 に，文献で報告されている精度と比較して，ほぼ最適な 4 通りと，すこし離れた値での 2 通りの結果を示す．表の最下行の $C = 1$ で $g = 1$ の場合は 63 時間 (228,310s) であり非常に長時間である．そこで，この場合を除いた 5 回の試行の合計時間，317,956 秒（88 時間）の 10 倍を表 6 の LibSVM の 50 回の推定訓練時間として比較した．

FCMC で $c = 8$ の場合の訓練時間は LibSVM(50 回) の 1/8515 で，表 6 の FCMC での三通りに $c = 2, r = 3$ の場合を加えた総訓練時間は 934 秒で LibSVM(50 回)

の 1/3404 である．FCMC ではテストデータの誤識別率が 5.36% まで改善された．

3.4 MNIST データ（0 から 9 までの手書き数字）

MNIST データは 0 から 9 までの手書き数字の画像データから特徴抽出を行った 60,000 件の訓練データと 10,000 件のテストデータからなっている (http://www.cenparmi.con cordia.ca/~jdong/Hero Svm.html)．このデータは 10 クラス問題であるので LibSVM では 1 対 1 方式と言われる 2 クラス毎の組み合わせで識別してその結果の多数決で各データのクラスを判定している．組み合わせの数は多くなるが，クラス毎の訓練データは 6,000 件程度と少ないので LibSVM の訓練は非常に高速である．図 5 から，FCMC のクラスター数は $c = 32$ で，$r = 30$ が良い．クラスター数をこれ以上大きくすると特異行列が発生した．誤識別率は LibSVM の方が 0.2% 良くなったが，FCMC の訓練時間は LibSVM(50 回）の 1/15 である．表 5 の FCMC での三通りに $c = 8$，$r = 10$ の場合を加えた総訓練時間は 481 秒で，LibSVM(50 回）の 1/7 である．SVM の計算オーダーは，データ数の 2 乗から 3 乗とされているので，１対１方式でデータ数が少ない時は多クラス問題でも高速になる．しかし，FCMC はさらに高速である．

3.5 結果の考察

以上五つの大量ベンチマークデータを用いて LibSVM との性能比較を行った．最適なパラメータを選択できるまでの時間の公平な比較は困難であるので，LibSVM では 50 回の探索で最適化できると仮定して推定時間を測定した．比較的少量の訓練データでの先行研究 [3, 5, 25] ではクラスター数を少なくして，パラメータを多くした場合に汎化性能が良くなったが，文献 [26] の結果に比べて，大量データではクラスター数を多くした方が改善された．クラスター数を多くすると，訓練とテストに要する計算時間は長くなるが，取り上げた何れのデータでも LibSVM にくらべて総訓練時間は短いと推定される結果が得られた．また，何れのデータでも LibSVM と同等またはそれ以上の識別精度が得られた．

本論文で提案した四つの改善結果について以下に考察する．

1) クラスタリングの初期化でのクラスター中心と共分散行列の計算に用いるデータを減らせば，木構造の根ノードに近い段階の計算は速くなるが，最終の葉ノードでの計算には全てのデータを用いるので，それほど大きな効果はない．

2) データを複数個のブロックに分割して読み込み，更新されたメンバシップの中間結果もそのブロック毎にハードディスクに書き出す方式は，メモリーの制限にかかることがないようにしたもので，メモリー不足になった段階で分割数を増やせば解決する．また，分割数を大きめにすることで仮想メモリーが使われるのを防止できる．

3) クラスタリングの反復（更新）回数を 1 回のみとしたのは，計算時間を速くするためで，このようにしても LibSVM と同等の精度が得られ，前報 [26] の結果に比べても回数は 1 回に限定しても良いといえる．

4) サンプリングした少ないテストデータでランダムサーチを行い，得られた解の近傍で全テストデータでの探索を行う方式は，このようなことで多少テスト時間を短くできることを示しているが，最適な減らし方（サンプリング）がどの程度であるかは今後の課題である．

以下に，FCMC を用いる際の留意点を列挙する．

1)　クラスター数は，2 分法で増やすので 2 のべき乗個あるが，実質的には二三通りに限定することができる．データ数が十分多くないときにクラスター数を大きくすれば特異行列が発生して計算の初期時点で停止する．また，クラスター中心や共分散行列が正確に求まらないために識別精度が悪くなる．従って，クラスター数の上限は，50 次元データでは，各クラスターのデータ数が最低でも 200 件以上になるようにするのが経験的な目安である．

2)　共分散行列を用いるので，データの次元数（特徴量の数）は，50 以下とし，それ以上のデータは PCA で 50 次元に圧縮する．この圧縮する次元数もパラメータであるが，経験的に 50 が適当である．共分散行列の低階数近似に用いる基底ベクトル数は，50 次元に圧縮したデータでは，30 と 10 で比較する．５次元データなら 3 と 1 で比較する．

3)　通常はクラスタリングの反復回数を多くしてもテスト性能は改善されない．この点は少量のベンチマークデータでの場合 [3, 12] と同様である．

4)　クラスタリングに要する時間は訓練データ数に比例し，クラスター数にも比例する．パラメータ最適化のための探索時間は，その回数を一定にすればテストデータ数とクラスター数に比例するが，訓練データ数にはよらない．クラスター数を等しくすれば，データの次元数（圧縮後の特徴量の数）とデータ数が同じな

ら，ベンチマークデータが異なっても，m と γ の値が異なっても訓練時間はほぼ等しい．従って，おおよその訓練時間は推定可能で，この点は LibSVM に比べて FCMC の実用上の大きなメリットである．

4 おわりに

改良された FCMC の訓練時間を，高精度な識別器として広く認められている LibSVM と比較した．反復回数 50 回に固定した LibSVM に有利な条件での比較であるが，パラメータ最適化までを含んだ総訓練時間は，訓練データが大量になるほど LibSVM に比べて大幅に短く，テスト精度は LibSVM に劣らない結果が得られた．

Matlab のインタープリータでの計算時間は，コンパイルすることや部分的に c 言語に変更することでさらに高速化が図れると期待できるので，その点を検証することが課題である．

駐車場の画像データにはエッジ抽出や SIFT 特徴量 [31] などの一般的な特徴抽出法を用いずに PCA でのデータ圧縮のみを用いたが，何らかの特徴抽出法で性能改善が可能かを検討すること，カテゴリカルデータなどの異なる種類のベンチマークデータでも性能比較することも今後の課題である．

参考文献

[1] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.

[2] S. Miyamoto, H. Ichihashi, and K. Honda, *Algorithms for Fuzzy Clustering, Methods in c-Means Clustering with Applications*, Springer-Verlag, Berlin, 2008.

[3] 市橋秀友，野津亮，本多克宏，セミハードクラスタリングとその識別器への応用，日本知能情報ファジィ学会誌，vol. 22, no. 3, pp. 358-367, 2010.

[4] 市橋秀友, 堅多達也, 藤吉誠, 野津亮, 本多克宏, ファジィ c 平均識別器による駐車場のカメラ方式車両検知システム, 日本知能情報ファジィ学会誌, vol. 22, no. 5, pp. 599-608, 2010.

[5] H. Ichihashi, A. Notsu and K. Honda, "Semi-hard c-means clustering with application to classifier design," *Proc. of the IEEE International Conference on Fuzzy System, World Congress on Computational Intelligence*, Barcelona, Spain, pp. 2788-2795, 2010.

[6] H. Ichihashi, T. Katada, M. Fujiyoshi, A. Notsu and K. Honda, "Improvement in the performance of camera based vehicle detector for parking lot," *Proc. of 2010 IEEE International Conference on Fuzzy System*, Barcelona, Spain, pp. 1950-1956, 2010.

[7] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[8] V. N. Vapnik, *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, Berlin, 1982.

[9] C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[10] T. Joachims, "Making large-scale support vector machine learning practical, " *Advances in Kernel Methods: Support Vector Machines*, A. S. B. Scholkopf, C. Burges, Ed., MIT Press, Cambridge, MA, 1998.

[11] T. V. Gestel et al., "Benchmarking least squares support vector machine classifiers," *Machine Learning*, vol. 54, pp. 5-32, 2004.

[12] 市橋秀友，長浦一哉、野津亮，本多克宏，パラメータ化した FCM 識別器のベンチマークテスト，日本知能情報ファジィ学会誌，vol. 22, no. 5, pp. 609-620, 2010.

[13] J. Platt, "Fast training of support vector machines using sequential minimal optimization," *In B. Scholkopf, C. Burges and A. Smola (Eds.), Advances in kernel methods - support vector learning*, Cambridge, MA: MIT Press, pp. 185-208, 1999.

[14] R. Fan, P. Chen, and C. Lin, "Working set selection using second order information for training support vector machines, " *J. Mach. Learn. Res.*, vol. 6, pp. 1889-1918, 2005.

[15] C-C. Chang and C-J. Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

[16] 笠井航，長谷川修，カーネルマシンへの高速ベクトル量子化の導入，日本神経回路学会誌，vol. 16, no. 3, pp. 149-157, 2009.

[17] 笠井航，戸部雄太郎，申富饒，長谷川修，オンラインプロトタイプ生成による大規模データに対する高速 SVM 構築法，電子情報通信学会論文誌 D, vol. J92-D, no. 6, pp. 784-792, 2009.

[18] D. D. Nguyen, 松本 一則, 橋本 和夫, 滝嶋 康弘, 寺邊 正大, 大規模 SVM 学習のための 2 段式逐次ワーキングセット選択手法, 日本データベース学会論文誌, vol.7, no.1, pp. 61-66, 2008.

[19] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines - Fast SVM training on very large data sets," *Journal of Machine Learning Research*, vol. 6, pp. 363-392, 2005.

[20] I. W. Tsang, A. Kocsor, and J. T. Kwok, "Simpler core vector machines with enclosing balls," *Proc. of 24th international conference on Machine learning*, pp. 911-918, 2007.

[21] G. Loosli and S. Canu, "Comments on the 'Core vector machines: Fast SVM training on very large datasets'," *Journal of Machine Learning Research*, vol. 8, pp. 291-301, 2007.

[22] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proc. of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.

[23] J. Kennedy and R. C. Eberhart, "Particle swarm optimization ," *Proc. of the IEEE International Conference on Neural Networks*, Piscataway, NJ, vol. 4, pp. 1942-1948, 1995.

[24] M. Clerc, *Particle Swarm Optimization*, Wiley, 2006.

[25] H. Ichihashi, K. Nagaura, A. Notsu, K. Honda, "Performance comparison of FCMC and LibSVM for classification of large data sets," *Proc. of the 6th International Conference on Soft Computing and Intelligent Systems*, Okayama, Japan, pp. 228-233, 2010.

[26] 市橋秀友，本多克宏，野津亮，多数のパラメータを用いるファジィc 平均識別器の訓練データ数による性能比較，日本知能情報ファジィ学会誌，vol. 23, no. 2, pp. 254-263, 2011.

[27] S. Miyamoto, T. Yasukochi, R. Inokuchi, "A family of fuzzy and defuzzified c-means algorithms, " *Proc. of the International Conference on Computational Intelligence for Modelling, Control and Automation*, Vienna, Austria, pp. 170-176, 2005.

[28] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, pp. 443-482, 1999.

[29] F. Sun, S. Omachi, and H. Aso, "Precise selection of candidates for hand written character recognition," *IEICE Trans. Information and Systems*, vol.E79-D, no. 3, pp. 510-515, 1996.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[31] D. G. Lowe, "Object recognition from local scale-invariant features," *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 1150-1157, 1999.