6　追加実験

6.1　課題 X：スペクトル包絡を用いた話者の同定（つづき）

第 3.8 節では，音韻の違いによっても話者の違いによっても変化するスペクトル時系列に対して，時間平均という操作を通して，音韻による影響を除去し，直流成分を捉えることで話者モデルを作成し，それを使って話者を同定することを検討しました。もっと良い方法はないでしょうか？

たとえば，話者毎に「あ」「い」「う」「え」「お」の五母音のスペクトル包絡が得られる状況にあれば，平均操作などせず，話者モデル i の「あ」と入力話者の「あ」,「い」と「い」, ⋯ と比較することも可能です。つまり，個々の音韻毎に話者モデルと入力話者を比較する，という方法です。こうした方が平均操作などしないので，より厳密な判定が可能になりそうです。では，このアイデアはどうやれば実装できるでしょうか？この手法は，GMM-supervector + SVM という方式として知られています。GMM = Gaussian Mixture Model, SVM = Support Vector Machine です。以下，説明します。その後で，GMM-supervector + SVM を超える性能を出した i-vector という方法についても説明します。

6.2　Gaussian Mixture Model (GMM)

6.2.1　多次元正規（ガウス）分布

東大工学部の学部 3 年男子学生の身長を調査すれば，凡そそれは正規分布（ガウス分布）の形状を示します。では，(身長，体重)，(身長，体重，視力)，(身長，体重，視力，通学時間) と想定する確率変数の次元を上げていくと，どうなるでしょう。正規分布にはベクトル（多次元量）を対象とした分布も存在します。

$$\mathcal{N}(o;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(o-\mu)^2}{2\sigma^2}\right\} \tag{1}$$

$$\mathcal{N}(o;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{(o-\mu)^T\Sigma^{-1}(o-\mu)}{2}\right\} \tag{2}$$

上式は 1 次元正規分布，下式が多次元正規分布です。確率変数 o は上が一次元，下がベクトルです。上式の標準偏差（σ）が下では分散共分散行列（Σ）となっていることに注意して下さい。T は T 乗ではなく，ベクトルの転置を表します。

6.2.2　Gaussian Mixture Model (GMM)

一次元であれ多次元であれ，正規分布は単峰性の分布形状しか表現できません。しかし実際には，対象とする確率事象に対して，多峰性の分布形状を仮定したい場合が殆どです。このような場合，総和が 1.0 となる重み $\{w_n\}$ を導入し，これを用いて下記のようにして多峰性分布を表現することがあります。

$$\mathcal{N}(o;\{\mu_n\},\{\Sigma_n\},\{w_n\}) = \mathcal{N}(o;\Theta) = \sum_{n=1}^{N} w_n\, \mathcal{N}(o;\mu_n,\Sigma_n) \tag{3}$$

Θ は，モデルパラメータである $\{\mu_n\},\{\Sigma_n\},\{w_n\}$ 全体を意図する記号です。これを Gaussian Mixture Model (GMM) と言いますが，GMM は下記のように解釈することも可能です。N 個の正規分布があり，観測ベクトル o は，いずれかの正規分布より生起します。この時，観測ベクトル o が観測され，かつ，それが第 n 正規分布からの出力である同時確率を $P(o,n)$ とすると[19]，

$$P(o) = \sum_{n=1}^{N} P(o,n) = \sum_{n=1}^{N} P(n)P(o|n) = \sum_{n=1}^{N} P(n)\, \mathcal{N}(o;\mu_n,\Sigma_n). \tag{4}$$

[19]「どのようなベクトルが観測されるのか」「どの正規分布からそれが出力されるのか」の両方とも確率事象である。

つまり w_n は，どの正規分布から出力が出やすいのかの偏りを示す，事前確率 $P(n)$ [20]と理解できます。一番身近な例で考えれば，東大工学部３年の「男子」学生の身長の確率分布ではなく，東大工学部３年生，全体の身長の分布を考えてみましょう。つまり，男女一緒に考える，ということです。男性と女性が各々，単峰性の分布を示したとしても，両者全体を合わせた分布は多峰性（二峰性）になるでしょう。この時 $\{w_n\}$ を考えると，男性の重みの方が大きくなるはずです[21]。この w_n は $P(n)$ と解釈できます。

GMM のパラメータ Θ は，与えられた訓練データ $\{o_t\}$ を最も高い確率で生成するように推定されます（最尤推定法。$P(\{o_t\};\Theta)$ を考え[22]，これを最大化する Θ を推定する）。推定の実装は EM (Expectation Maximization, 期待値最大化）法が広く使われています。

ここで，日本語の異なり音素数が M だった場合，GMM の分布数を M として，ある話者の十分な音声サンプルに対する GMM を推定した場合，個々の分布は何に対応するでしょうか？多少楽観的な予想としては，個々の分布が個々の音素に対応することが期待されます。ケプストラムベクトルを使って GMM を構成すれば，平均ベクトル $\{\mu_n\}$ は各音素のスペクトル包絡に相当することになります。そう，ある話者の各音素のスペクトル包絡が凡そ求まるはず，と期待できます。

6.2.3 実際の GMM パラメータの推定

ある話者の入力音声から GMM を構成し，その話者の個々の音素のスペクトル包絡を求めるとした場合，どれくらいの入力音声が必要なのでしょう？もし 1 分の音声が必要であれば，話者を認識するのに 1 分もかかることとなりますが，これでは実用上，意味を持ちません。

ここで登場するのが，UBM-GMM (Universal Background Model) です。事前に多数の話者の多数の音声を使って，GMM を構成しておきます。つまり「人の声とは，こういう多峰性分布をするものである」という知識を事前に獲得しておきます。そして，ある話者の少量の音声サンプル群 $\{o_t\}(1 \leq t \leq T)$ を使って，その話者用に UBM-GMM を修正します。修正方針ですが $P(\{o_t\};\Theta)$ を最大化するように Θ を修正する（最尤推定する），のではありません。これを行なうと $\{o_t\}$ だけに頼ってパラメータ推定する（やり直す）こととなり，事前に得た知識が無駄になります。今の場合，事後確率 $P(\Theta|\{o_t\})$ を考え，これを最大化するように Θ を修正します。これは以下のように考えると分かり易いでしょう。まず人の声を表す GMM パラメータ Θ を求めた。今，人の声として新たなサンプル群 $\{o_t\}$ が得られた。その事実を考慮して Θ を修正する。つまり「元々の訓練データ＋ $\{o_t\}$」を使って最尤推定して Θ を推定し直したいが，元々の訓練データがもうない。そこで $\{o_t\}$ を条件として Θ の事後確率 $P(\Theta|\{o_t\})$ を考え，それが最大化されるように Θ を調整する，ということになります。ベイズの定理より $P(\Theta|\{o_t\}) \propto P(\{o_t\}|\Theta)P(\Theta)$ であり，

$$\log\left(P(\Theta|\{o_t\})\right) \quad \propto \quad \log\left(P(\{o_t\}|\Theta)\right) \quad + \quad \log\left(P(\Theta)\right) \tag{5}$$

となります。右辺第一項は当該話者の少量データ $\{o_t\}$だけに着眼した出力確率であり，第二項は Θ の事前確率です。結局，Θ の修正による左辺の最大化は，右辺の二項のバランスをとって最大化することとなります。詳細な議論は省きますが，修正後 GMM の平均ベクトル $\{\mu'_n\}$ は下記のようになります。

$$\mu'_n = \alpha_n \bar{o}_n + (1 - \alpha_n)\mu_n$$

これは，分布 n に対する $\{o_t\}$ の確率的平均ベクトル $\bar{o}_n$ と UBM-GMM の平均ベクトル μ_n との内分です。

$$\alpha_n = \frac{N_n}{N_n + r} \quad \text{(内分比)}$$

---

[20] 確率 $P(x)$ は，新たな確率変数 y を導入することで，$P(x) = \sum_y P(x, y) = \sum_y P(x|y)P(y)$ と書くことができます。条件付き確率 $P(x|y)$ は，ある条件下での確率であり，何某の事実が与えられた後の確率という意味で，事後確率と呼ぶことがあります。一方，$P(x)$ は，$y, z,$ など他の要因を全く考えずに x だけの確率を考える，という意味で事前確率と呼ぶことがあります。

[21] 理由は明白ですよね。

[22] $P(o|\Theta)$ と $P(o;\Theta)$ の違いは，Θ を確率変数として考えるか，(やがてある値が与えられる）パラメータとして考えるかの違いです。前者であれば，確率変数なので $P(\Theta)$ を考えることになりますが，後者では考えません。それは確率変数ではないのですから。

$$N_n = \sum_{t=1}^{T} P(n|o_t) \quad (\{o_t\} \text{ が分布 } n \text{ から生成される確率的個数})$$

$$P(n|o_t) = \frac{P(o_t, n)}{P(o_t)} = \frac{P(o_t|n)P(n)}{\sum_m P(o_t|m)P(m)} = \frac{w_n \mathcal{N}(o_t|\mu_n, \Sigma_n)}{\sum_{m=1}^{M} w_m \mathcal{N}(o_t|\mu_m, \Sigma_m)} \quad (o_t \text{が分布 } n \text{ に対して持つ帰属確率})$$

$$\tilde{o}_n = \frac{1}{N_n} \sum_{t=1}^{T} P(n|o_t) o_t \quad (\{o_t\} \text{ から計算される確率的平均ベクトル，期待値})$$

なお r は本来，UBM-GMM 構築時のサンプル数，及び，T から理論的に計算される量ですが，実際にはある定数を与えることが多いです。本実験でも「とある」定数を割り当てています。

このようにして UBM-GMM から得られた，入力話者用 GMM の平均ベクトル $\{\mu'_n\}$ は，各音素の「その話者の」スペクトル包絡に相当することになります。この $\{\mu'_n\}$ を混合分布数だけ結合した，非常に高次元のベクトルをスーパーベクトル（supervector）と呼びます。これを用いて話者を同定します。

6.3 Support Vector Machine (SVM)

6.3.1 Binary classification と線形分離平面

話者認識は，入力された音声が，用意された N 人の話者モデル（話者テンプレート）のいずれなのかを同定するタスクです[23]。N クラスのうちどれに属するのかを当てることになりますが，まずは $N = 2$ の場合を考えてみましょう。binary classification と言われます。

既に入力音声は，とあるベクトル空間の一点として表現されていました。binary classification の場合，その空間を二つに分け，その点がどちらに存在するのかで識別します。では，どうやって領域に分割するのでしょう？一番簡単なのは線形（超）平面で分けることです。空間が m 次元 $(x_1, x_2, ..., x_m)$ であれば平面は

$$a_0 + a_1 x_1 + a_2 x_2 + ... + a_m x_m = 0 \tag{6}$$

となります。今注目している話者ベクトルを使って左辺を計算し，その正負で識別することになります。

では，どうやって $(a_0, a_1, ..., a_m)$ を決めるのでしょう？図 1 を見てください。これは二つのクラスのサンプルが分布している様子を示しています。このサンプル群を適切に区分できる平面は無数に存在します。どのような平面がより適切な分離平面と言えるでしょうか？SVM では，分離平面の近くに存在するサンプルをサポートベクター（support vector）と呼び，これらから分離平面までの距離が最も長くなるように平面を設定します。分離平面に近いサンプルというのは，識別誤りを起こし易いサンプルですから，これらから出来るだけ遠ざけて平面を設定するのは，至極最もなことです。

6.3.2 非線形分離の線形分離化

以上は，二つのクラスに属するサンプル群が平面で奇麗に分離できることを前提としていました。これは実用上，どのくらい正しい前提なのでしょうか？もし非線形平面で分離する必要がある場合（つまり，平面で分離できない場合），どのようにすれば良いでしょうか？例えば 3 次元空間 (x_1, x_2, x_3) を考えます。この場合，各次元を使って，次のような拡張空間を考えてみます。

$$(x_1,\ x_2,\ x_3,\ x_1x_2,\ x_2x_3,\ x_1x_3) \tag{7}$$

「え？」と思う人も多いと思います。既に存在する次元を使って強引により高次元化しています。「なんだか冗長な空間定義に思える」人もいるでしょう。でもちょっと考えてください。自己相関関数でも扱ったように，二つの変量 x と y の積 xy は，両者の関係性，連動性を表現する変量となります。元々の空間 (x_1, x_2, x_3) では，各々の変量の関連性は，サンプル群の分布を見て判断することになりますが，上記のような定義は，そ

[23]「入力された音声が A さんの声であるか否かを検証する」場合は話者認証と言う。

の関連性を数値として一つの次元に明示的に表示する，という方策をとっています。じゃあ，なんでこんな冗長空間を定義するのでしょう？実は，元々の空間では線形分離が出来ない場合でも，このような冗長な多次元空間を用意すると見事に線形分離が出来ることが多いのです。一般に次元数が高いほど，サンプル群は平面で分離できるようになります。元々の空間の点を x とし，高次元化された空間の対応点を X とすると，

$$A_0 + A_1X_1 + A_2X_2 + ... + A_MX_M = 0 \qquad (M \text{ は非常に大きな値}) \tag{8}$$

で分離できるようになります。

6.3.3 高次元空間とカーネル・トリック

では，どうやって冗長空間を定義するのでしょうか？今考えるスーパーベクトルは既に数千次元ありますが，これを更に高次元化する必要があるのでしょうか？SVM ではカーネル法という方法で無限大次元にまで空間を拡張することがあります。また「え？」と思う人もいるでしょう。$M = \infty$ であれば「そもそも式 (**8**) の左辺が計算できねーじゃん。それが無限大だろ？」と突っ込みたくもなるでしょう。詳細な種明かしは参考文献に譲るとして，どうやって無限次元へと拡張するのか，少しだけ説明します。

ある X に対して式 (**8**) の左辺を計算する場合，実は X の各次元の値（X_i）そのものが分からなくても，計算できてしまいます!!!。$\{A_i\}$ を求める（分離平面を定義する）ためには訓練データが必要です。今，個々の訓練サンプルを X^j と書くことにします。そうすると，ある X に対する式 (**8**) の左辺値は，X^j と X の内積を全て用意すれば，何と計算できてしまうのです[24]。そして，内積 $X^j \cdot X$ の計算を元々の低次元空間の点 x^j，x を使って行なうのがカーネル関数 $K(x^j, x)$ です。つまり，元々の低次元空間の点を使った演算でもって，高次元空間の内積を計算し，それを使って式 (**8**) の左辺を計算する訳です。計算上必要なのは $K(x^j, x)$ だけであり，高次元空間の各次元が具体的にどう表現できるのか，知らなくてもよい，ということになります。別の言い方をしてみます。低次元空間の二点 x，y を使って行なう「ある演算」を考えます。で「その演算が内積となる高次元空間は存在するのか？」という問題を考えます。そうすると，その演算を内積とする高次元空間が存在するための必要十分条件が導出されます。そう，この条件を満たす様々なカーネル関数が知られており，これらを試し，最高精度を示すものを採択することが広く行なわれています。どんな高次元空間なのか，を具体的に把握しなくても，適切なカーネル関数を選択する＝適切な高次元空間を選択する，ということになるのです。識別精度を高めることだけを目的とすれば[25]，大切なのは高次元空間定義ではなく，$K(x^j, x)$ の定義，ということになります。

6.3.4 2 クラスから N クラスへ

SVM に関する情報提供として，最後に，N-class classifier の構成の仕方について説明します。SVM は binary classifier です。話者認識は N 人の誰なのかを当てるタスクです。つまり，N-class の classifier が必要です。SVM をどのように拡張して N-class に対応させるのでしょう？

二つの方法が広く知られています。one-vs.-all と one-vs.-one です。前者は「あるクラスとそれ以外」を分離する binary classifier を N 個用意します。入力 X に対して N 種類の binary classifier を適用し，分類結果と，各々の分離平面からの距離を参照することで最終的な識別結果を得ます。もう一つは N 種類あるクラスに対して，入力 X を「クラス i かクラス j」に分類する binary classifier を ${}_NC_2$ 個用意します。得られた ${}_NC_2$ 個の判定結果を参照して，最終的な識別結果を得ます。

---

[24] カーネル・トリックと言われます。

[25] 機械学習の分野では，こういう研究姿勢はよく見られます。しかし，どういう空間が使われているのか？何故，その空間なら識別性が上がるのか，ということを（識別性が上がる，という「事実」以上のレベルで）理解しようとすれば，この姿勢は必ずしも良いものではありません。Performance（精度）vs. interpretation（理解）という言葉を時々聞きますが，これを端的に示しています。

6.4 i-Vector

6.4.1 GMM-supervector + SVM の復習

GMM-supervector と SVM，凡そ理解できたでしょうか？ここまで来るのにかなり苦労した学生もいるのでは，と思いますが，実はここまでが準備段階です。GMM-supervector とは何だったでしょう？各話者において，各音素（相当の音響事象）のケプストラムどのような分布を有するのか，その音素分布を音素数（M）だけ持ち寄って，その分布の平均ケプストラム（= 平均対数スペクトラム包絡，であることは理解できてますよね？）を M 個分連結した，非常に高次元のベクトルでした。

このベクトルに対して，カーネルトリックを用い，上記の高次元空間を更に高次元化し，時には無限大次元化することで，各話者を線形平面で分離できるようにし，この超高次元空間において話者を同定する，という方法を実装しているのが，SVM でした。

一般にパターン認識・同定問題とは，入力されたメディア情報（何某のセンサーから得られる物理情報）から特徴ベクトルを抽出し，その特徴ベクトルを，当該タスクに tune された（訓練サンプルを使って学習された）識別機（classifier）に入力することで認識・同定する問題を指します。

GMM-supervector + SVM，かなり強力そうですが，これを超える精度を示す方法があります。それが i-vector です。さて，GMM-supervector + SVM のどこに改善の余地が残されているのでしょうか？

GMM-supervector は，音素相当の音響事象 M 個として説明してきましたが，実際には M = 数千と，音素数を遥かに超える事象数を対象とすることが多いです。こうすると，さすがに特徴ベクトルとしては次元数が多すぎて，訓練サンプルを使って学習しても過学習[26]となる可能性が生まれます。過学習が起きなかったとしても，GMM の M 個の山は，話者の違いだけでなく，マイクの違い，背景雑音の違い，伝送特性の違いなど，様々な音響変動の影響を直接的に受けます。つまり，音声入力に不可避的に混入してしまう，様々なノイズに対する安定性（専門用語だと頑健性と言います）が低くなることがあります。

こういう時に使われる常套手段として，GMM-supervector そのもの（対象を細かく描いた特徴そのもの），ではなく，GMM-supervector を「敢えてぼかして」低次元化した特徴を使う，という方法があります。

6.4.2 主成分分析

まず，ベクトル v を代数的に表現する際に,「基底ベクトルの重みが使われている」という事実を確認します。$v = (x, y, z)$ とした場合，それは「直交かつ長さ = 1」の三つの基底ベクトル $i_x = (1, 0, 0)$, $i_y = (0, 1, 0)$, $i_z = (0, 0, 1)$ を用いて（各々，各軸上の単位ベクトル），

$$v = x\, i_x + y\, i_y + z\, i_z$$

となります。逆に言えば，上式右辺で表現されるベクトルを (x, y, z) と書くことになります。

さて，三つの基底ベクトルは i_x，i_y，i_z でなければならないのでしょうか？三つの各々が「直交かつ長さ = 1」を満たす基底ベクトルは無数にあります。それこそ，i_x，i_y，i_z の三つを原点を中心にしてぐるぐる回して定義される新たな三つの基底ベクトルは，常に「直交かつ長さ = 1」，という条件を満たしています。

質問します。3 次元の特徴ベクトルを考えます（身長，体重，年齢とか）。数千個のサンプル群が与えられ，これらを 3 次元空間にプロットする場合に，これらのデータに「最適な」基底ベクトルは何でしょうか？

これまで何も考えずに，$i_x = (1, 0, 0)$, $i_y = (0, 1, 0)$, $i_z = (0, 0, 1)$ で張られた空間にデータをプロットしてきたと思いますが，i_x, i_y, i_z を回転して新たに定義される無数の基底ベクトル群を考えた場合，どれが最適な基底ベクトル（軸）になるのでしょう？そもそも，最適ってどういう意味で最適なのでしょう？

`http://goo.gl/tzYMVw`

を見てください。

[26] 識別機を訓練するために使われた訓練サンプルに対しては非常に高い精度となるが，それ以外のデータ（評価データ）に対してはの精度が予想以上に低くなってしまう。

左図では，2 次元ベクトルで表現される数千個のデータがラグビーボールの形状で分布しています。この楕円形状を表現するための最適な（2 次元）基底ベクトルとは何でしょう？一つの指針を考えます。

与えられたデータ群に観測される「ばらつき」の様子を一番効果的に示せる基底ベクトルをまず選ぶ（設計する)。次に，その基底ベクトルでは表現できない「ばらつき」を一番効果的に示せる基底ベクトルを次に選ぶ（設計する)。これを繰り返す。

具体的な手順を考えます。まず，楕円の中心に原点を移動します（中図)。次に長軸方向に第一基底ベクトルを考えます。次に，短軸方向に第二基底ベクトルを考えます（右図)。このようにして張られた基底ベクトルを $i_{x'}$，$i_{y'}$ とすると，

$$v = x\, i_x + y\, i_y = v_0 + x'\, i_{x'} + y'\, i_{y'}, \qquad\qquad v - v_0 = x'\, i_{x'} + y'\, i_{y'}$$

となります。v_0 は新たな原点です。新たな軸を使うと，$v = (x', y')$ となる訳ですが，x'，y' の意味は分かりますか？第一基底ベクトルに対する重み，第二基底ベクトルに対する重みです。そして，基底ベクトルは先頭から順に，使用したデータ群の「ばらつき」をできる限り効果的に表現できるよう，選択されています。

特徴ベクトルの次元数が上がっても同様,

$$v - v_0 = \sum_{n=1}^{N} x_n\, i_{x_n}$$

となります。「ばらつき」を一番効率的に示せる基底ベクトル（の方向）のことを第一主成分，二番目を第二主成分，n 番目を第 n 主成分と呼びます。つまり，主成分分析を通して基底ベクトルを選ぶ（設計する）と，基底ベクトルを「ばらつきを表す貢献度，重要度」という観点から順位付けして並べることになります。

6.4.3 i-Vector

話を少し前に戻します。GMM-supervector を「敢えてぼかして」低次元化した特徴を使う，という話をしました。GMM-supervector を沢山集めて（沢山の人から集めて)，それに対して主成分分析を行うことを考えた場合，どうやってぼかしますか？

既に気付いていると思いますが，重要な順に基底ベクトルが求まっているのですから，

$$v_0 + \sum_{n=1}^{N} x_n\, i_{x_n}$$

に対して

$$v_0 + \sum_{n=1}^{M} x_n\, i_{x_n} \qquad \text{但し, } M < N$$

を求めれば，ぼけたベクトル（細かな詳細，細かな変動を除去したベクトル）になるはず，というのは分かるでしょう。そして，v_0 は全データに対する共通項ですから，上記のサンプルは

$$(x_1, x_2, ..., x_M)$$

という N よりも小さい，M 次元空間の点で表現できることになります。この重みベクトルのことを話者認識の業界では，i-Vector と呼んでいます[27]。

[27]なお，i-Vector 計算で使われる主成分分析は，学部生が教わる「普通の」主成分分析とは少し異なります。後者の「普通の」分析はデータ群の分散共分散行列を計算し，その逆行列を求め,,,, という手順を実行しますが，Supervector の次元数は高いので逆行列計算が通常できません。そういう場合に主成分分析（相当）の処理を行う方法が知られており，それを使っています。

6.4.4 コサイン距離による識別・同定

GMM-supervector の代わりに，i-Vector を導出しました。では，SVM の代わりは何でしょう？SVM はカーネルトリックだの，無限大次元空間だの，ケッタイなもの，仰々しいものが出てきましたが，i-Vector の場合，拍子抜けするくらい，シンプルな識別機が広く使われています。

二つのベクトル x，y の違いを定量化する場合，ユークリッド距離が広く使われていますが，パターン認識の世界では，両者の角度を使うことも広く行なわれています。$x = \overrightarrow{\mathrm{OX}}$，$y = \overrightarrow{\mathrm{OY}}$ の場合，$\angle$XOY を見る訳です。角度そのもの，より，その余弦，$\cos\angle$XOY に着眼します。これは，

$$\cos\angle\mathrm{XOY} = \frac{x \cdot y}{|x||y|}$$

より求まります。なお，i-Vector は長さ $= 1$ に正規化することも広く行なわれているので，その場合は，二つの i-Vector の差異は，$x \cdot y$ と内積を計算するだけで得られます。

訓練データを使って各話者の i-Vector を複数求めておき，入力音声から構成された i-Vector がコサイン距離的にどの i-Vector に近いのかを求めれば，識別できることになります。