

予測と制御による階層的運動学習

講師：春野 雅彦

レポート：大羽 成征・王 懐成

1 運動学習におけるモジュール性と階層性

「運動学習におけるモジュール性と階層性」に関する話を少しだけしましょう。

私たちは毎日異なった環境で異なった物体に対して運動してますが、運動というのは一般に高次の目的があるわけで、それを達成するために、一般的に階層的な構造を持っています。

また運動には文脈があります。人間においてはインプットといえば聴覚であり視覚であるわけですが、言語だ何だといったって何かそこで働きかける運動をしなければいけないわけですね、だから一言で運動といっても、そこに現れる運動の文脈にはあらゆるマルチモーダルな物事が含まれるわけで、それが変化するに従って、運動の入出力が変わる。だからそういうことを考えに入れて運動の指令が変化しなければならないわけです。

（質問）運動の入出力といった場合の出力と、運動指令とは違うのか？

（春野）一緒です。ほとんど一緒です。

次にモジュール性というのは本当かどうかなのですが、本当かどうかは知りません。ただ例えば腕のリーチング運動で、まずふつうにやらせておいて、次にしばらくフォースフィールドを外からかけた環境でやらせて適応させる。それで、適応させたらまた元に戻す、適応させて元に戻すを繰り返すと、スイッチングがどんどん早くできるようになる。

つまり新しい適応によってもともとあるものが完全に壊れてしまうことはないので、すると、なにかモジュール性があるのかな、と言える気がします。

昨日の榎野さんの講義に、プリズム適応で両方の表現が残っているという話もありましたよね、ああいうのもまたモジュール性ですよ。つまり新しい学習をしても昔の学習が壊れてしまわずに残っているわけです。そこで、まずここではじめの問題なのですが、モジュール性は受け入れてもらうことにして、ここからの話をしたい。

（質問）視覚学習の場合、適応後のものが何年も続くということは、以前のものが壊れてしまって新しく変化したものが続いているということになりますよね。

（春野）それはもとにもどしたら、また最初から学習するということですか？

（質問者）課題はあくまでも、あるものを良くするという課題であるから、課題というのは課題 specific な適応を導くわけで・・・

(春野) 課題は日常ではどんどん条件がスイッチしていますよね、それに追従しないと
いけない。そのときに、前のものが毎回完全になくなってしまおうのか、それとも、変わっ
たものに徐々に適応するけれども、もとのものにすぐに戻れるのかという問題です。日常
全く体験しないような入力に対する適応はよくわからないけれど、入力はほぼ同じである
ふつうの条件ではという・・・

(質問者) わかりました

モジュール性はそういうものだとな納得していただいたとして、いまコントローラーが複
数並んでいるとしましょう、こういう場合にどのコントローラーが適切かを選ぶ問題があ
ります。

制御は生きている中でやるわけで、あれもこれも試して選ぶということではできないわけ
です。やる前にどれがよいかを決めねばならない、それを考えるのが前半の話になります。

2 確率に関する復習

[スライド：ベイズ則手書きスライド]

細かい話をやる前に、簡単な話をしておきましょう。

n 個の選択肢があってそこから i 番目を選ばねばならないとき、事前確率 $p(i)$ は何の根
拠もなしに i 番目の確率はこれくらいだ、と信じているということを意味します。

これに対して事後確率 $p(i|X)$ は、観測 X を見た後に i 番目の確率はこれくらいだと言
うことです。

条件付き確率の定義に戻ると、 X のもとでの i の確率 $p(i|X)$ というのは、 X と i とが
両方起こる確率 $p(i, X)$ に比例しますよね。

$$p(i|X) \propto p(i, X)$$

X と i が両方起こる確率 $p(i, X)$ は、 i のもとでの X の確率 $p(X|i)$ と i の確率 $p(i)$ をかけ
たものですよね。

$$p(i, X) = p(X|i)p(i)$$

X の確率 $p(X)$ というのは、 j と X と両方起こる確率 $p(j, X)$ を j に関して和をとったも
のですよね。

$$p(X) = \sum_{j=1}^N p(j, X)$$

必読論文の中で λ だなんだとごちゃごちゃうるさいのがあったとおもいますが、あれはこ
れだけのことなんです。

こんどは、 j と X の両方起こる確率 $p(j, X)$ をばらすと、 j がおこったときに X が起こ
る確率 $p(X|j)$ と、 j の起こる確率 $p(j)$ をかけておけばいいわけですよね。

$$p(j, X) = p(X|j)p(j)$$

そうすると、全体はこのようになります。

$$p(i|X) = \frac{p(i, X)}{p(X)} = \frac{p(i)p(X|i)}{\sum_{j=1}^N p(j, X)} = \frac{p(i)p(X|i)}{\sum_{j=1}^N p(j)p(X|j)}$$

次に、 x という vector があります。例えば、今日のいろいろな株価とすることにしましょう。 y は明日の株価平均とすることにしましょう。

$$X = (x, y), y = f_i(x) + \epsilon_i$$

N 個の関数 $f_i(x)$ があって、それぞれが明日の株価を予測するということを考えましょう。 i というのはそれぞれ、野村証券、丸三証券、山一証券とすることにしましょう。それぞれは、今日の株価の vector x をみて、明日の株価平均の予測をしています。それぞれの予測 $f_i(x)$ にはノイズが含まれているためピタッとは重ならず、誤差があります。ここで i 社たとえばつぶれてしまった山一証券の予測が正しいものである確率を考えましょう。山一証券の予測確率は、 ϵ が Gaussian ノイズであるとするならば、 i 社のもので、こういう株価の関係になった確率はこうなります。

$$p(X|i) = p(x, y|i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2\sigma_i^2}|y-f_i(x)|^2}$$

数日間ぶんのデータと各社の予測を見て、それでどの会社に任せようかなということを考えるとしましょう。

はじめは山一証券が大きい会社であることから、事前確率 $p(i)$ としてここに任せようと思っていたかも知れない。その後各社の予測をみて、データを見た上で、各社を選ぶ確率はこういうようにしたらいいのです。

$$p(i|X) = \frac{p(i) \exp(-\frac{1}{2\sigma_i^2}|y-f_i(x)|^2)}{\sum_{j=1}^N p(j) \exp(-\frac{1}{2\sigma_j^2}|y-f_j(x)|^2)}$$

みなさんに読んでいただいた論文では、ここ $p(i)$ に $\hat{\lambda}_i$ がはいついて、ここ $p(i|X)$ に λ_i が入っているわけです。もしこういうフレームワークになじみのない人がいたら、こういうことだったと理解してください。

ここでは各証券会社がどういう予測をするか知っていたわけですがけれども、それを知らない場合に、コントローラーとか予測モデルを入れて予測部分の学習もしようというのが、ここからの話です。ここまで理系の人にはつまらん話かも知れませんが、あの論文にはこれを書くスペースがありませんでしたので以上のような解説をしました。

3 mixture of experts

[スライド： mixture of experts]

これまでの関連研究はいろいろあるのですが、その中から mixture of experts を挙げます。

これを構成する要素に expert network、gating network があります。先ほどと同じで、 x と y のセットを考えます。 x は実数値のベクターで、 y を予測します。 x が入力で、 y が出力です。各 expert network は入力を受け取って、それぞれが y 予測値を出します。これだけですとどれを信用していいかわからないので、gating network を考えます。gating network は、 x を受け取ってその信頼度 g^1, g^2, \dots を出します。このとき全体の出力 y は、重みをかけて足すことで、

$$y = \sum_i^N g_i y_i$$

というように計算します。信用するエキスパートは重みを小さくして、山一証券は信用しない・・・ここに山一証券の関係者いないでしょうね？（会場笑）うちの親は山一だとか。

細かい学習則は説明しませんがポイントとなるは、gating net. の重みを決めるのに、入力を見てやるのだということです。

mixture of experts の親戚に hierarchical mixture of experts というのがあって、それは mixture of experts が縦につながっているのですが、縦につながっている場合にも gating net. は x だけみるわけです。このことの何が気に入らないかというと、

[スライド：2 ページ目]

gating net. が入力を分割するという困難な仕事を請け負っているため分割がうまくいかないことが多いのです。全部を gating net. のせいにすることがよいことかどうか、これだけでははっきりと何かを主張することはできませんが、脳のモデルに少しでも関連を持たせるとしたら、あらゆる階層のレベルで、入力を直接観測しているというのはどうも信じにくい。それぞれの階層でそれぞれの重みを決めるのに、すべてが最初の入力を見て決めているとは思えない。

4 Multiple paired of forward inverse models

じゃあどうするんだというのが、

[スライド：MFIM]

こういうモデルです。フィードバック誤差学習がここにあります。

これら一つ一つの箱がモジュールの種（たね）のようなもので、それぞれ等質的なものが n 個並んでいます。各モジュールの種は inverse model、forward model、responsibility predictor からなります。各モジュールに関して見ていくと、まず inverse model に理想軌道が与えられてそれぞれを実現するための運動コマンドを inverse model が出します。各モジュールの inverse model がそれぞれ制御コマンドを出してきて、どれを選ぶかと言うところで forward model を使います。制御コマンドを実行したらどういう軌道が実現するかを、forward model がもとに戻して予測するわけです。さっきのヘルムホルツマシン

に似てますよね？ n 個並んでいる各 forward model が次の時点の位置・速度・加速度を出します。その中でどれがよいかを、ベイズ則に似たこういう式で決めます。

$$\lambda_t^i = \frac{e^{-|x_t - \hat{x}_t^i|^2 / 2\sigma_i^2}}{\sum_{j=1}^N e^{-|x_t - \hat{x}_t^j|^2 / 2\sigma_j^2}}$$

なんの事前知識もなしにこの λ_t^i を決めるので、ベイズ則の $p(i)$ に当たる部分を省略して上のようになります。

(質問) 各モジュールで順モデルは全部同じですか？

(春野) 違います違います。

それぞれが違う関数を使って、次の状態 x を予測しています。 x_t が理想状態で、 \hat{x}_t^i はそれぞれの予想状態です。ついでに言えば、inverse model も違います。理想軌道が入ってくると各 inverse model が運動指令を出し、同時に各 forward model が自分の予測を出すわけです。

(質問) 評価したいのは制御信号の善し悪しなのに、それを異なる順モデルで予測して良いのか？ そのバイアスがかかってこないか？

(春野) 正確にはそうなのですが、forward model は学習によって徐々に正しくなっていくものです。ある時点でみると、forward models 同士にたまたま差があって、たまたま一つのモデルの予測がよい場合、これにばかり教師信号がいきます。つまり現時点の context においてこの model だけが学習します。それにつれて、inverse model と forward model が一緒に正しい学習をします。

しかし context がバーンと変わったら今度はこのモデルは不適切なわけですね。こんどは他のどのモデルがよいか再び決めて、そこにばかり誤差が集められてそこが学習すると、そういうことになります。

学習則の話をししましょう。 [スライド：学習則]

λ_t^i はそれぞれのモジュールの良さを表します。これが事後確率にすぎないと言うことは分かっていただけだと思います。これを使って学習するとき、学習則には誤差があって、入力があって、 i 番目のモジュール重みを持って学習するわけです。だから、 λ_t^i がゼロであるモデルは学習しないわけです。たまたま良かった人のところばかりに誤差が行って学習しますが、良くない人には誤差が行かず、学習が行なわれません。

全体の運動指令は、各 inverse model の出力をそれぞれの良さに応じて足して決めます。つまり λ_t^i の値の大きさに応じてモデルの制御コマンドを信用してやるわけです。inverse model の学習も forward model の学習と全く同じようにします。誤差の項 $u_t^* - u_t^i$ と責任信号 λ_t^i と入力を掛け合わせています。夏の学校ではここまでの話が内部モデル講習会になってましたので理解していただいていると思いますが、制御コマンドの誤差は本当は分からないので、そのために、かわりにフィードバック誤差 u_{fb} を使っています。

$$\Delta \alpha_t^i = \epsilon \lambda_t^i \frac{d\psi_i}{d\alpha_t^i} (u_t^* - u_t^i) = \epsilon \frac{du_t^i}{d\alpha_t^i} \lambda_t^i u_{fb}$$

ここに feedback controller からの出力 u_{fb} が入ってきます。

まとめると、「責任信号を事後確率で計算して、それによって順モデルと逆モデルを同時に学習を進める」ということです。

さてはじめに、「運動というのは入力がいっぱいで、出力が一つです。だからいろんな情報を一緒に入れられる扱える枠組みにしなければいけませんね」という話をしましたが、それをするのが、responsibility predictor です。

[スライド：責任信号予測器]

責任信号予測器というのは視覚などの運動にあまり関係ない入力から、次の状態の責任信号を予測しています。それがこの式です。

$$\hat{\lambda}_t^i = \eta(\gamma^i, y_t)$$

勘のいい人は分かると思いますが、ものを見て予測をたてるというのをここでやっています。予測をたてることは事前確率に相当します。 y_t というのは時刻 t における外からの文脈入力で、何か視覚入力でもいいし、音でもいいです。 γ というのは responsibility predictor のウェイトです。出力 ($\hat{\lambda}$) は時刻 t における responsibility の予測値です。わかりにくかったようですが、 η は例えば γ をウェイトとした 3 層 perceptron によって、予測値を出します。実際に運動してみると本当の λ_t^i が分かるわけですから、本当の λ_t^i を教師信号として予測値 $\hat{\lambda}_t^i$ の学習が進められるわけです。例えば視覚入力に対して予測信号を出力した場合その誤差が後から帰ってくるので、それを使ってこれ自体の学習を進めると言うことになります。それですから、責任信号予測器で計算したこの $\hat{\lambda}_t^i$ というのは事前確率と思えばいいわけです。つぎに事前確率と運動した結果を混ぜるにはどうすればいいかというと、これ

$$\lambda_t^i = \frac{\hat{\lambda}_t^i e^{-|x_t - \hat{x}_t^i|^2 / 2\sigma_i^2}}{\sum_{j=1}^N \hat{\lambda}_t^j e^{-|x_t - \hat{x}_t^j|^2 / 2\sigma_j^2}}$$

をつかうわけです。これは先ほど見ていただきました、Bayes の形をしています。 $p(i)$ のところが $\hat{\lambda}_t^i$ になっているだけです。

これで、全体でこのアーキテクチャの一番簡単な学習がどうなるかについて、そのなかでキーとして働く責任信号というのがどういうものかという説明をしました。

5 シミュレーション

では、実際に簡単なシミュレーションをしたらどうなるのか、

[スライド：シミュレーション]

腕のモデルとしてこういう線形のモデルを考えます。

$$M \dots x + B\dot{x} + Kx = u$$

外からの外力として u があり、腕のダイナミクスが時間とともに変わります。具体的には M と B と K の値が 5 秒ごとに、 α, β, γ に変わります。これに関していい例がないので、昔銅屋さんに教えてもらった例を出しますが、「スノーボードをしていて急にアイスバーンにつっこむような感じ」です。5 秒ごとにアイスバーン（会場笑）。

3 つのダイナミクスがあるので 3 つの順逆モデルを用意します。また各モジュールの順逆モデルは、ともに線形ネットワークだと思って学習します。

[スライド：結果の 3 次元グラフ]

初期値を与え、学習をがらがら回していくと正解へ収束していきます。ニューラルネットのウェイトを使って、forward model から予測される M, B, K と、inverse model から予測される M, B, K とを計算できるのですが、それを見ると、いつも forward model の方がよいということが分かります。それはなぜかということ、まずフォワードモデルの方の学習がある程度進んで、正しく問題を分割できるようになってはじめてインバースモデルの学習が進んでいるからです。

本当にこんなことができるのか？ということを見てもらうために、プログラムを走らせてもらいましょう。

（演習）

[スライド：責任信号予測器と size weight illusion]

今度は responsibility predictor に対する入力として、4 かける 4 の行列のうえで表現された「形」を入れます。（こんなのを size weight illusion なんていうと絶対怒られるんですけど。）同じ大きさで、重さの違うのを持ち上げると、絶対ちいさいもののほうが重たい。それは自分が予測したものととの差が大きいので、重いと感じるのですが、そういうのを size weight illusion と呼びます

さて、運動と視覚的なものの入力とのアソシエーションとしてこういう例を与えます。学習の時は A と α, B と β, C と γ というペアにして学習させます。 α, β, γ 、というのは物体の特徴です。そして学習後にはじめて A と γ の組み合わせを与えて責任信号と加速度の誤差を見ます。いままでは、 A と α の関連しか知らないで、運動前は視覚入力に基づいて α が選択されているわけです。しかし実際に運動が選択されてみると違いに気付いて、正しいもの (γ) が選択されるわけです。こうして外からの文脈としての入力と実際の運動とを連合させて学習することができていることがわかります。

（質問）途中で真ん中のモジュールが選択されているのは何ですか？

（春野）これは、何なんでしょう？ノイズのせいだと思うんですけど

（質問者）A かな B かなと迷って C だったと？

（春野）そうですね。ノイズが軌道に入っているなのでそのせいでたまたまだと思うんですけど、両方の入力の組み合わせが効いてきますよね、形からの出力と運動の誤差と。そのせいでこのへんでは迷っている、と、そういうことだと思います。

ここまでのところで、モジュールの選択が学習にもよるけれども、初期値（事前分布）にも依存するということが分かってきました。なぜかということ、ノイズがあるときに、そ

の時点その時点の誤差だけから制御をして良いのか、実際の人間の行動でも、もうちょっとある程度、しばらくはモジュールがすべき行動が変わらないとか、トップダウンの予測などがあるのではないか？

そこで、このモデルに時間的なこと、急には変わらないぞと入るためにマルコフ的なことを入れてみました

6 マルコフ的にモジュール遷移する MPFIM と EM アルゴリズムによる学習

[スライド：マルコフ性を利用した MPFIM]

マルコフ性というのは強化学習で学んだので、大体ご存知だと思います。今やっていたことを時間的に書いてみると図のようになります。 S_1 、 S_2 、 S_t があってそれぞれ状態に対応します。 t は時間で、それぞれが時刻 t にどれかのモジュールの値を取りながら、逆モデルと順モデルを使って制御する状況になっています。

ここで考えたいのは、 S_n によるモジュール選択がちょこちょこ変わらないという大きなレベルでの予測をしたいということで、 S_t から S_{t+1} に移るときにその移りやすさを遷移行列にしました。マルコフ性の性質によって、 S_{t+1} は S_t にしか依存しないと考えます。 i 番目の順モデルの線形パラメータを W_i つまり、重みとして、標準偏差を σ_i とします。今 $\forall A$ モジュール i だと考えて、そのモジュール i がデータを生成した確率が次式のように正規分布になっています。この状態は、外からでは、わからない、つまり、隠れ変数というわけです。隠れ変数がわかったとして、 i 番目の隠れ変数を取っているとしたときの尤度は、次式のようにあらわされます。

$$L_i(Y|W_i, \sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2\sigma_i^2}(Y - X_i W_i)'(Y - X_i W_i)\right)$$

あとは、石井先生に習ったよう EM アルゴリズムを使って計算できます。

[スライド：結果比較 1]

前記の課題に上のレベルから安定化するような要素を入れたもの (HMM) と前記の方法を適用したものを比較した結果を図に示しました。上側のパネルで、実線は目標加速度、点線は実現加速度、下側のパネルは、責任信号の変化です。左側のパネルは HMM の方法を、右側は前記の方法の結果を示しています。見て、おわかりの通り HMM では、切り替えが適切に行われるのに対し、前記の手法は頻繁に切り替えが起こっています。これに伴い HMM では、加速度の誤差も小さくなっています。これは HMM では、状態遷移確率を考慮してモジュールの切り替えを安定化させるためだと考えられています。

(質問)

それは、3 状態ですか？

(春野)

そうです、3 状態です。

(質問者)

学習をしたのは、遷移行列の対角成分が、割に大きくて、状態 A にいるとしたらそこにしばらくいるということを学習したのですか？それとも 5 秒経ったら、次に移り変わるような学習をしているのですか？

(春野) それは、単に時間的に一定で、対角成分も大きくて、次の状態に移るものがちょっとあるということです。

[スライド - 結果の比較 2]

これを、様々な切り替え数(何回ごとに切り替えるか)、とノイズレベルの違い、つまり初期状態の違いで両者の比較を行いました。表に様々な間隔でモジュールを切り替えた場合の両者の誤差の比を示します。具体的には、分子が HMM の誤差、分母が前記の方法での誤差です。各ネットワークの初期値として 20%、40% のノイズを加え、50 回の試行平均値を示しました。結果からわかることは、初期値が真の値から遠く(ノイズが大きい)、モジュール切り替えの頻度が低い場合に HMM の頻度が顕著に高くなることがわかります。

7 階層的な MPFIM

[スライド: 小脳-皮質ループと階層的 MPFIM]

階層性を取り扱うようなメカニズムとして、もっと深い階層をやろうと思います。現在の研究では大脳基底核ループが非常に人気がありまして、中原さんや銅野さんもやっております。階層性のもとになるのは何であろうかと考えてみます。小脳に内部モデルがあるとして、階層性というのはいろいろなレベルの内部モデルがインタラクションして仕事を行っているとおもいます。

都合のいいことに、脳にはそのような構造があって、図のように歯状核が視床を介しているところとループになっていることがわかりました。M1 やワーキングメモリをすると言われている 46 野や pre-SMA、SMA など高次なところとループを作って、それぞれ異なる部位のシステムで、別々の内部モデルを使って学習をしているのではないかと思います。それは、非常に高次なところと、運動制御に密着した低次なところが皮質の双方向結合を用いて双方向的にインタラクションをして、階層的な所動をしているのではないかと考えられます。この仮説のモデルとして階層的な MPFIM を考えます。

(質問)

それは、M1 と上位のレベルでやるのか。それとも M1 の中にいろいろなモジュールがあってその間でやるのですか。

(春野)

M1 のループの中に複数のモジュールがあって、同じようなタスクはこのループをやっている、上位の方は、また違うタスクを行います。

MPFIM 図でみせた横に並んだモジュールが、この一つのループに対応していると考えられています。

[スライド：階層的な MPFIM]

この階層的な MPFIM を安直に図にしますと二つの階層の簡単なものになります。それぞれのモジュールの中に逆モデルと順モデルがあり、それぞれ三つ並んでいます。その上に上位レベルをつけました。下のモデルでは、順モデルが軌道予測をしていて、逆モデルが制御コマンドを送っています。先ほど、責任信号予測の話で、上から事前確率を計算しているという話をしましたが、ここでは上のレベルというのは、それぞれ下のレベルの事前確率を計算していることになります。下のモデルの順、逆モデルは、制御に密着した入出力をしています。上のレベルではもうちょっと抽象化したレベルのことで取り扱いたいので、下のレベルのどれがいいかという事前情報を出力しています。事前情報を出せば、次の時点、答えが返ってくるので、それを使って上位レベルの学習をすることができます。

[スライド：階層性 MPFIM と双方向性]

先ほどもお話ししたようにぼくは言語に興味がありまして、言語とか運動とか一緒ですけど、その生成と認識というのが密接に結びついていて、それは先ほどの Helmholtz マシンにもありましたね。このモデルの楽しいところは生成するときと認識するときで同じアーキテクチャが使えるということです。このことをまず系列物体制御のシミュレーションを使ってお見せします。

[スライド：系列物体制御のシミュレーション]

異なる物体をもち与えられた軌道に従って腕を 30 秒間動かす課題です。物体の特性は以下の 2 つの系列に従い 5 秒毎に変化します。

系列 A : α, β, γ

系列 B : β, α, γ

勘のいい方は、お気づきでしょうが、先ほどの、下三つ、上二つのモジュールからなるシステムの図がありまして、

[スライド：階層性 MPFIM と双方向性]

下のモジュールにはそれぞれ順、逆モデルがありましたので、そのモジュールは、このシミュレーションのそれぞれ α 、 β 、 γ の学習を行い、上の二つはシーケンス 1 とシーケンス 2 に対応する学習をしてほしいわけです。

[スライド：系列の学習]

結果は図から系列 A の要素が学習できていることがわかります。下位レベルは、 α 、 β 、 γ に対応するシーケンスができており、上レベルでは、学習時はだめだが、学習後は、それぞれの順番に対応して発火します。系列の学習を行うと上レベルにそれぞれ系列に対応するパターンができていくことがわかります。下の図は、系列 1 と系列 2 の運動制御をしているわけですが、それをやっている間に上のレベルのそれぞれに系列 1 と系列 2 に対応するパターンが生成されたことを示しています。実際、サル記憶主導による系列

運動課題で、歯状核の前頭連合野とループ構成する部分にこのような系列に特異的に反応するニューロンが見つかっています。

[スライド：階層性 MPFIM と双方向性]

他人が制御した信号を、理想軌道として入力すれば、自分が学習をしたシーケンスに基づいて、その信号を分けることができます。今までは、生成するときは、自分の理想軌道を入力していたが、人がした運動系列を実現するためには、自分だったらどのようなモータコマンドを出力するか、上のレベルはどのように活動すればいいのかという結果が出てくるわけです。そうすると観測したほかの人のデータのリダイヤルつまり、自分がデータを生成するときや、異なる人の似たアクションを見たときにセグメンテーションをやっているということができます。 [スライド：ノイズがある場合の生成]

最後に階層性がどのようなメリットあるのかをお見せして終わりにしたいと思います。

このグラフは、先ほどの系列 A を実行しているときに5%のノイズを感覚フィードバックに加えたものの結果を示しております。中央のパネルは上のレベルの責任信号を示しています。下の二つのパネルは下の階層での尤度と責任信号を示していて、その中で上のほうは、トップダウン信号を切ってしまったとき、つまり、運動誤差だけから決めた責任信号のようなものの推移を表しています。ノイズのせいでガタガタしているのがわかります。一方、下のもう一つの方のパネルは上のレベルからの系列予測の信号があって、その結果、安定しているのがわかります。

8 おわりに

[スライド：おわりに]

予測と制御に基づく階層的運動学習のモデルを MPFIM を中心にして述べました。シミュレーションの結果を通してこのようなモデルが階層的な運動を学習し得ることを示しました。また、単なる工学的なモデルではなく、このようなモデルを用いて、数理的な背景をもつ実験を設計できる可能性について述べました。

(質問) 高次 46 野のループから pre-SMA, SMA のループを通して、信号が伝わっていくとき、それぞれのモジュールの役割の違いはどうなっていますか？

(春野) ここでは大脳基底核のループを考慮に入れていないので、わかりませんが、小脳帝国主義の見地からいいますと、46野と小脳のループがシーケンスのメインのところであって、pre-SMA や SMA は上の信号を受けているだけで、ニューロンがシーケンスに特異的に発火しているだけだと考えられます。

(質問) 先生の興味はモジュールを切り換えることが中心で、それをたまたま、運動制御に応用して見たということによろしいのでしょうか。

(春野) それは、good question です。もともとは、統計学をやっていたので、そういう事がいえますが、今は、こちら(脳のモデル)にシフトしています。

(質問者) モジュールに分けてやるメリットは何ですか？

(春野) 一個所のところで学習させたのでは、新しい学習をさせると今までの学習が保存できませんから。

(質問) 運動制御のときは軌道目標があって、それを自分の中に取り込んでアクションを起こし、その結果をフィードバックして学習しますが、言語認識の場合は言葉を聞いて瞬間的にフィードバックをするわけじゃない、つまり、例えば僕はこういう事が分かりました、あなたは、こう言っていることはあってますかいいえ違ってますというようなフィードバックがあるわけではありませんね。それについてはどのように考えていますか。

(春野)

[スライド：階層性 MPFIM と双方向性]

言葉の場合は、五味先生がやっておられますが、調音パラメータやもっと上のレベルの意図があるわけですね。まず、運動制御の話しからいいますと(下のモジュールを指して)この辺のレベルがいきなりわかったわからないというのではなく、まず、パッと話しを聞いて(音声波形を見て)それを自分がやるのであればどういう制御コマンドをつかい、音の系列に直したらどうなってとかを考えるのではないのでしょうか。

(質問者) はじめは、おっしゃるような方法で、エラーを含んだ情報で構築したものを使ってやってみる。その誤差を耳で聞いて、学習していく。つまり、聴覚系から運動系への情報を使って学習していくわけなんですよ。

(春野)

はい、聴覚系から運動系へのなにかしらの関数があって、聴覚のフィードバックを誤差として運動系で学習をするわけです。

(質問者)

しかし、耳が聞こえても話せない方の場合は、どのように言語認識してるのでしょうか。

(春野) それは、わかりません。

(質問) このモデルの目標は、不連続なあるいは離散的な変化に対応するようなモデルをたてたいという事からはじまって、それを言語のような離散的なものに応用していこうということですね。

(春野) 特に、離散的にパシッと変ることにこだわっているわけではなく、複数のコンテキストによって変るといふことのがやりたいというのが一つです。それと順モデルと逆モデルを組み合わせるとパーセプションとジェネレーションとい話しができるのでそういうことがしたいのです。