数式処理について７　　　　　　　　2013/10/06（日）

　式のパターンをかなり、知的に評価して、式を変形し、その先の目標を達成するようにもっていくという、高度のセンサーシステムも必要になるようです。こうゆうのは、基本的に認知プロセスを作り置き、その上に、高度の目標をもったプロセスを機械学習で得ていくという方向に持っていくことが重要と考えます。そんな、例を次に挙げたいと思います。東大の２０１０年の数学の問題です。

【問題】

（１）　すべての自然数 k に対して、次の不等式を示せ。

1/(2*(k+1))<∫<from_case:0;to_case:1>((1-x)/(k+x))dx<1/(2*k)

（２）m>n であるようなすべての自然数 m と n に対して、次の不等式を示せ。

(m-n)/(2*(m+1)(n+1))<log(m/n)-Σ<from_case:n+1;to_case:m>1/k<(m-n)/(2*m*n)

【解答】

　この問題の「∫」の中身をまともに計算することは困難です。先ず、そのことをセンサーで知る必要が有ります。分母に x が無くなればいいなと。幸い不等式ですから、おおざっぱな計算に持っていけるはずです。そこで、x を消すことを考えてみましょう。

　k<k+x<k+1 です。そこで、

∫<from_case:0;to_case:1>((1-x/k+1))dx<∫<from_case:0;to_case:1>((1-x/k+x))dx

<∫<from_case:0;to_case:1>((1-x/k))dx

∫<from_case:0;to_case:1>((1-x))dx=x-(1/2)x^2<from_case:0;to_case:1>

=1-(1/2)=1/2

そこで、（１）が言えることが分かります。

（２）を言うには、log の式がでていることから、∫(1/x)dx のパターンがどこかに現れることを推定します。実際、∫<from_case:0;to_case:1>((1-x)/(k+x))dx の中にそのパターンがありますから、実際にこれを計算する必要があると、見込ます。

(1-x)/(k+x)=(1+k-(k+x))/(k+x)=(1+k)/(k+x)-1 と変形できますから、実際に∫を計算しますと、((1+k)log(k+x)-x) <from_case:0;to_case:1>=(1+k)log(k+1)-1-(1+k)log(k)

結局、（１）から、

1/(2*(k+1)*(k+1)<log(k+1)-log(k)-1/(k+1)<1/(2*k*(k+1))と（２）式に近いパターンが得られます。

この問題の肝は、ここで、1/(2*k*(k+1))が(1/2)*(1/k-1/(k+1))と分解し、階差数列を作ることに気づくことです。その階差性のヒントは、log(k+1)-log(k)のパターンから得られます。

実際、log(m/n)-Σ<from_case:n+1;to_case:m>1/k

=log(m)-log(n) -Σ<from_case:n+1;to_case:m>1/k で、求める不等式の中間の式に限りなく近いかたちです。そうして、

Σ<from_case;n;to_case:m-1>(log(k+1)-log(k)-1/(k+1))

=log(m)-log(n) -Σ<from_case:n+1;to_case:m>1/k

で、Σ<from_case;n;to_case:m-1>1/(2*(k+1)*(k+1))

＞Σ<from_case;n;to_case:m-1>1/(2*(k+1)*(k+2))

＝Σ<from_case;n;to_case:m-1>(1/2)*(1/(k+1)-1/(k+2))=(1/2)*(1/(n+1)-1/(m+1))

=(m-n)/(2*(m+1)(n+1))

で、Σ<from_case;n;to_case:m-1>1/(2*k*(k+1))

=Σ<from_case;n;to_case:m-1>(1/2)*(1/k-1/(k+1))=(1/2)*(1/n-1/m))

=(m-n)/(2*m*n)

証明おわり。

この問題のように、こう、パターンを追う問題って多いですね。基本を押さえて、正攻法で解く問題もあります。その場合でも、式の持っている特徴をセンサーで調べて、そのうえで式の変形とか立式とかを試行錯誤するという形に、成っていくのが、数学の問題なような気がします。

おわり