

1 時系列データの変動傾向の学習

データの変動には何らかの規則性があるかもしれません。そこでここでは、前年と比べて次の年の平均気温がどうなっているかについて、規則性を学習することを考えます。学習の結果として、たとえば平均気温の低い年が続いたら次の年は気温が上がる、などといった規則性が得られることを目標とします。

2 入力値の変化を調べます

updown.c

```
/*
updown.c
入力値の変化を調べます。
1 行に 1 つの実数が記述されたファイルを入力し、
増加 (+) 減少 (-) を出力します。
使い方
./updown <(入力ファイル名)> <(出力ファイル名)>
*/

#include <stdio.h>
#include <stdlib.h>

#define TEXTLENGTH 4096

int main(int argc, char *argv[])
{
    char    text[TEXTLENGTH];
    double newval, lastval;

    if(fgets(text, TEXTLENGTH, stdin) == NULL){
        fprintf(stderr, "データがありません\n");
        exit(1);
    }

    if(sscanf(text, "%lf", &lastval) < 1 ){
        fprintf(stderr, "データが不正です (%s) \n", text);
        exit(1);
    }

    while(fgets(text, TEXTLENGTH, stdin) != NULL){
        if(sscanf(text, "%lf", &newval) == 1 ){
            if((newval - lastval) >= 0){
                printf("+\n");
            }else{
                printf("-\n");
            }
        }
    }
}
```

```

    }
    lastval = newval;
  }
}

return 0;
}

```

3 「updown」の操作方法

「updown」の操作方法の例

```
./updown < kion2.txt > pm.txt
```

上記の「./」は、Linuxでのプログラムを実行する時の例です。
 ウィンドウズでは、updown < kion2.txt > pm.txt 「Enter」と入力してください。

4 前年比で気温の高低を、学習結果から予測するプログラム

m11.c

```

/*
  m11.c
  前年比での気温の高低を、学習結果から予測するプログラム
  入力は標準入力から与え、出力は標準出力に出力します。
  使い方
  ./m11 < (入力ファイル名) > (出力ファイル名)
  入力ファイルでは、気温の高低を+および-で記述します。
  出力ファイルには、予測結果を表示します。
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TEXTLENGTH 4096
#define PLUS 0
#define MINUS 1
#define NODATA -1

void learning(int last, int lastlast, int last3, int kb[2][2][2]);
void prediction(int last, int lastlast, int last3, int kb[2][2][2]);

```

```

int main(int argc, char *argv[])
{
    char    text[TEXTLENGTH];
    int     i;
    int     last, lastlast, last3;
    char    chr;
    int     kb[2][2][2] = {0};

    last3 = lastlast = last = NODATA;

    while(fgets(text, TEXTLENGTH, stdin) != NULL){
        if(sscanf(text, "%c", &chr) == 1 ){
            switch(chr){
                case '+':
                    last3    = lastlast;
                    lastlast = last;
                    last     = PLUS;
                    break;
                case '-':
                    last3    = lastlast;
                    lastlast = last;
                    last     = MINUS;
                    break;
                default:
                    break;
            }
        }else{
            break;
        }
        learning(last, lastlast, last3, kb);
        prediction(last, lastlast, last3, kb);
    }

    printf("學習結果\n");
    printf("+++ : %d\n", kb[PLUS][PLUS][PLUS]);
    printf("++- : %d\n", kb[MINUS][PLUS][PLUS]);
    printf("+-+ : %d\n", kb[PLUS][MINUS][PLUS]);
    printf("+-+ : %d\n", kb[MINUS][MINUS][PLUS]);
    printf("-++ : %d\n", kb[PLUS][PLUS][MINUS]);
    printf("-+- : %d\n", kb[MINUS][PLUS][MINUS]);
    printf("--- : %d\n", kb[PLUS][MINUS][MINUS]);
    printf("--- : %d\n", kb[MINUS][MINUS][MINUS]);

    return 0;
}

void learning(int last, int lastlast, int last3, int kb[2][2][2])
{
    if((last != NODATA) && (lastlast != NODATA) && last3 != NODATA){

```

```

        ++kb[last][lastlast][last3];
    }
}

void prediction(int last, int lastlast, int last3, int kb[2][2][2])
{
    if(kb[last][lastlast][PLUS] >= kb[last][lastlast][MINUS]){
        printf("+\n");
    }else{
        printf("-\n");
    }
}
}

```

5 「ml1」の操作方法

「ml1」の操作方法の例

```
./ml1 < pm.txt
```

上記の「./」は、Linuxでのプログラムを実行する時の例です。
 ウィンドウズでは、ml1 < pm.txt 「Enter」と入力してください。

6 「ml1」の実行結果の例

「ml1」の実行結果の例

```

+
-
-
+
-
-
+
学習結果
+++ : 7
++- : 15
+-+ : 27
+-- : 12
-++ : 15
-+- : 23
--+ : 11
--- : 3

```