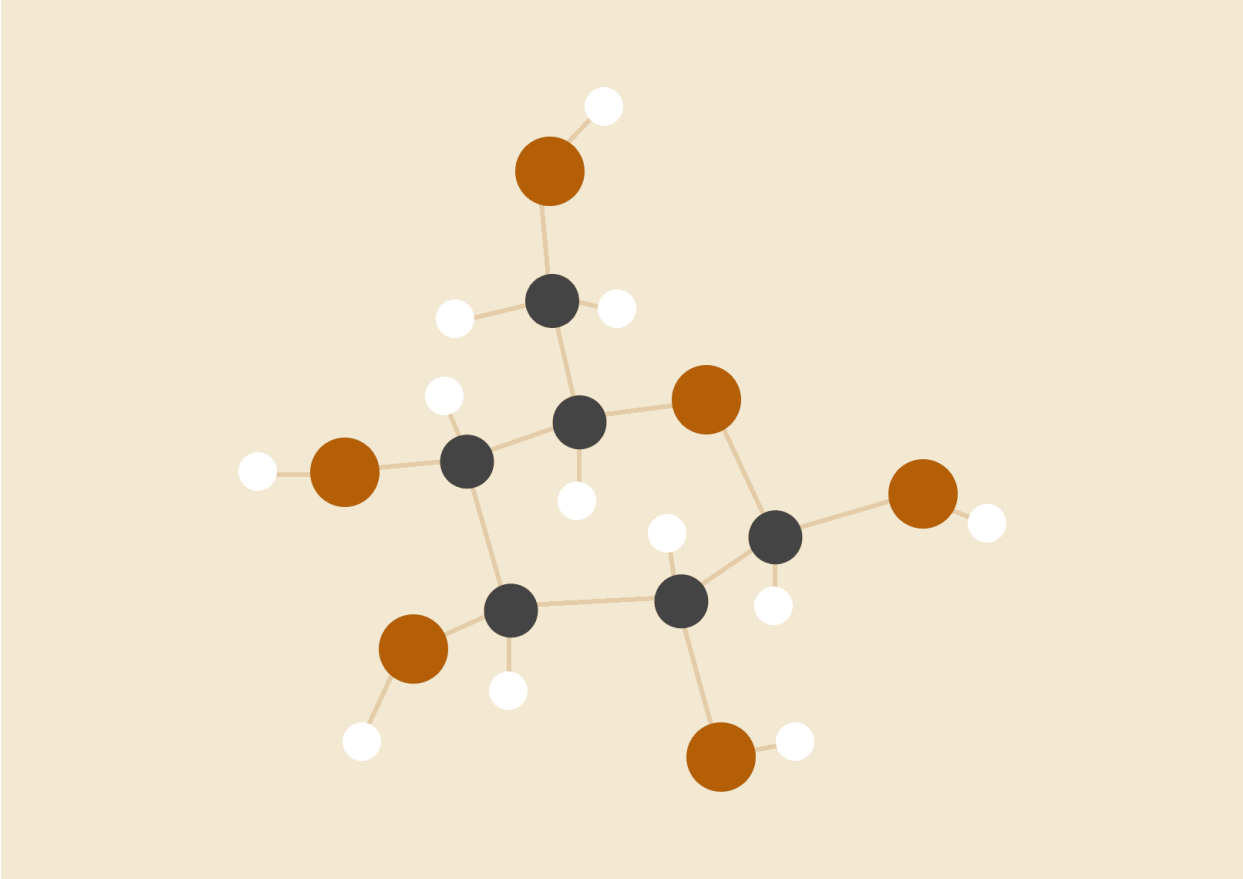
EXAMEN COMPUTACION PARALELA Y DISTRIBUIDA

20 de Diciembre de 2022



Jose Alfredo Calfuen Saalzar

20/12/2022

PROBLEMÁTICA

Una pequeña empresa (25 trabajadores), dedicada al rubro de la minería, trabaja manipulando materiales peligrosos, a pesar de las medidas de seguridad, siempre hay accidentes, que aunque no sean graves, disminuyen la fuerza laboral y afectan a la formación de los equipos.

Para su análisis, se adjunta un csv con los datos históricos, desde que se iniciaron las

operaciones (06/10/2017) hasta la fecha de corte (10/12/2022):

● Fecha. Esta es la fecha en la que sucedió el accidente.

● Cantidad de accidentes. Es el número de accidentes ocurridos.

Se solicita desarrollar una aplicación que, dada una fecha futura, pueda predecir la cantidad de accidentes esperados para el día de entrada. Esta aplicación debe implementar un modelo matemático, que use los datos históricos, esté codificada en C o C++ y use OpenMP o MPI para maximizar el rendimiento.

Entrada.

La aplicación debe leer la fecha como el primer argumento de la línea de comandos, el formato

de fecha corresponde al ISO 8601.

Resultado.

La salida del programa, debe mostrar en pantalla:

● La fecha (en formato ISO 8601).

● La cantidad de accidentes esperados.

● Un salto de línea.

Una vez entregue el resultado, el programa debe terminar.

HIPÓTESIS

A través de la regresión lineal es posible predecir el número de accidentes en una fecha determinada gracias al análisis del comportamiento actual de las variables.

PROCEDIMIENTO

1. Leer el archivo “datos_examen.csv” para luego poder extraer y transformar la información relevante como coeficientes (variables dependientes e independientes) que pueda utilizar la regresión.
2. Crear una función de predicción que retorne el valor de la predicción deseada por pantalla a partir de un valor de entrada.
3. Optimizar la función de predicción con openMP que permite la paralelización del proceso en cuestión.

DATOS

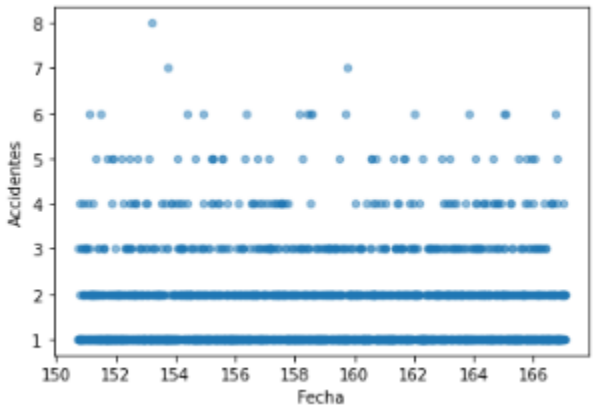
COLUMNA	TIPO DE DATO	
Fecha	string	2022-12-10
Accidentes	int	2

CÓDIGO

1. Función de transformar fechas a entero (di) date to int; extrae la fecha en el formato ISO 8601 y la transforma al número de días de diferencia a la fecha 01/10/2017. Esto para tener valores numéricos ordenados en la regresión lineal.

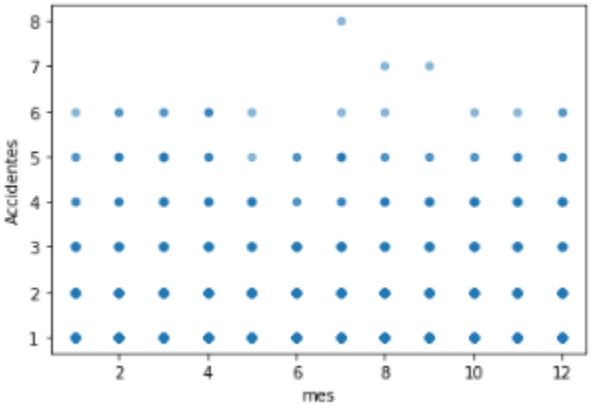
Al revisar los datos desde python se puede observar que no existe una gran correlación entre los datos transformados de fecha por este método de conversión a entero por lo que no se espera una gran predicción ya que no existe ninguna tendencia.

	Accidentes
Accidentes	1.000000
Fecha	0.003773



Tambien probe el separar los datos por dia fecha y año para ver si algun parametro funcionaba mejor para realizar el modelo y estos fueron los resultados:

	Accidentes
Accidentes	1.000000
mes	0.031764
ano	-0.002117
dia	-0.013485



Como se puede observar la mayor tendencia es con el mes que tampoco resulta ser muy significativa la correlación, por lo que; para mayor comodidad, se seguirá utilizando la primera versión con regresión lineal ya que no existe una gran variación entre las 2.

```cpp
int di(string iso_date) {
  // Fecha en formato ISO 8601

  // Extrae el año, mes y día de la fecha
  int year = stoi(iso_date.substr(0, 4));
  int month = stoi(iso_date.substr(5, 2));
  int day = stoi(iso_date.substr(8, 2));

  // Transforma la fecha a un número
  // El número es el número de días transcurridos desde el 01 de octubre de 2017
  long long number = (year - 2017) * 365 + day;
  for (int i = 2017; i < year; i++) {
    if ((i % 4 == 0 && i % 100 != 0) || i % 400 == 0) {
      number++;
    }
  }
  for (int i = 10; i < month; i++) {
    if (i == 2) {
      if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
        number += 29;
      } else {
        number += 28;
      }
    } else if (i == 4 || i == 6 || i == 9 || i == 11) {
      number += 30;
    } else {
      number += 31;
    }
  }
  return number;
}
```

2. Creación del modelo de regresión lineal

Para la construcción del modelo de regresión lineal se deben calcular los siguientes parámetros

$$Pendiente = \ m = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}$$

$$Interseccion = b = \frac{\sum y - m(\sum x)}{n}$$

$$\sum xy = suma\ del\ productos = x_1y_1 + x_2y_2 + x_3y_3 \ldots\ldots x_ny_n$$
$$\sum y = suma\ de\ valores\ de\ y = y_1 + y_2 + y_3 \ldots\ldots y_n$$
$$\sum x = suma\ de\ valores\ de\ x = x_1 + x_2 + x_3 \ldots\ldots x_n$$
$$\sum x^2 = suma\ de\ valores\ de\ x^2 = x_1^2 + x_2^2 + x_3^2 \ldots\ldots x_n^2$$

función para cálculo de la pendiente m

```cpp
// Calcula el coeficiente de la variable independiente (m) para una regresión lineal
double calc_m(const std::vector<double>& x, const std::vector<double>& y) {
  int n = x.size();
  double sum_x = 0;
  double sum_y = 0;
  double sum_xy = 0;
  double sum_x2 = 0;

  // Añade la directiva #pragma omp parallel for para paralelizar el bucle for
  #pragma omp parallel for reduction(+:sum_x,sum_y,sum_xy,sum_x2)
  for (int i = 0; i < n; i++) {
    sum_x += x[i];
    sum_y += y[i];
    sum_xy += x[i] * y[i];
    sum_x2 += x[i] * x[i];
  }
  return (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x * sum_x);
}
```

función para el cálculo de la variable b.

```
// Calcula el término independiente (b) para una regresión lineal
double calc_b(const std::vector<double>& x, const std::vector<double>& y, double m) {
  int n = x.size();
  double sum_x = 0;
  double sum_y = 0;

  // Añade la directiva #pragma omp parallel for para paralelizar el bucle for
  #pragma omp parallel for reduction(+:sum_x,sum_y)
  for (int i = 0; i < n; i++) {
    sum_x += x[i];
    sum_y += y[i];
  }
  return (sum_y - m * sum_x) / n;
}
```

función para predecir el valor independiente y

```
// Predice el valor de la variable dependiente (y)
double predict(double x, double m, double b) {
  return m * x + b;
}
```

3. Función main

En la primera parte de la función se pide el ingreso de una fecha en formato ISO 8601 para realizar la predicción, se habilita el uso de OpenMP estableciendo el número de hilos a utilizar, se abre el archivo csv con los datos y crean los vectores X e Y para guardar los datos del archivo posteriormente.

```
int main() {
  cout << "Ingrese la fecha en formato ISO 8601 (aaaa-mm-dd): ";
  string fechaStr;
  cin >> fechaStr;

  // Habilita el uso de OpenMP
  #pragma omp parallel
  {
    // Establece el número de hilos a utilizar en el programa
    omp_set_num_threads(2);
  }

  // Abre el archivo CSV para su lectura
  ifstream file("datos_examen.csv");
  if (!file.is_open()) {
    cerr << "Error: no se pudo abrir el archivo" << std::endl;
    return 1;
  }

  // Vector para almacenar las variables independientes (fechas)
  vector<double> X;
  // Vector para almacenar las variables dependientes (accidentes)
  vector<double> Y;
```

Siguiendo con la función main, se recorre el archivo csv para extraer los datos y guardarlos en X e Y, para la variable X se pasan las fechas por la función date_to_int para guardarlas como un valor entero definido por dicha función, cerrando así el archivo csv luego de terminar la extracción. Una vez tenemos los vectores de x e y, los ingresamos a las funciones para calcular la pendiente(m) y la intersección (b), luego, se pasa la fecha de entrada por la función date_to_int para poder realizar la predicción de accidentes ingresando los valores de b,m y x_pred a la función predict. Finalmente se muestran los datos por pantalla y se termina la ejecución del programa.

```
// Lee línea por línea del archivo
string line;
  // Saltamos la primera fila de encabezado
getline(file, line);
while (getline(file, line)) {
  // Divide la línea en campos separados por ';'
  istringstream line_stream(line);
  string field;
  //Nos saltamos la primera fila de cabecera
  getline(line_stream, field, ';');
  // Almacena el primer campo (fecha) en el vector de variables independientes
  X.push_back(date_to_int(field));
  // Lee el segundo campo (entero) y lo almacena en el vector de variables dependientes
  int dependent_var;
  line_stream >> dependent_var;
  Y.push_back(dependent_var);
}

// Cierra el archivo
file.close();

// Calcula el coeficiente de la variable independiente (m) y el término independiente (b) para la regresión lineal
double m = calc_m(X, Y);
double b = calc_b(X, Y, m);

// Predice el valor de y para x = fecha ingresada utilizando la regresión lineal
double x_pred = date_to_int(fechaStr);
double y_pred = predict(x_pred, m, b);
cout << "Para la fecha: " << fechaStr << " Se esperan : " << y_pred << " accidentes"<< endl;

return 0;
```

REFERENCIAS

1. http://integrauctemuco.blogspot.com/2011/12/regresion-lineal-lenguaje-de.html
2. https://www.youtube.com/watch?v=0GYWLbKC2QA&ab_channel=DavidArmend%C3%A1riz
3. https://learn.microsoft.com/es-es/cpp/parallel/openmp/a-examples?view=msvc-170