

Timeseries Statistics

Metodi Matematici e Statistici (6 CFU)

Stefano Borzì

Indice

- Introduzione
- Moving Average & Linear Weight MA
- Trend
- Noise
- Smoothing (Noise Removal)
- Forecast
- Referenze

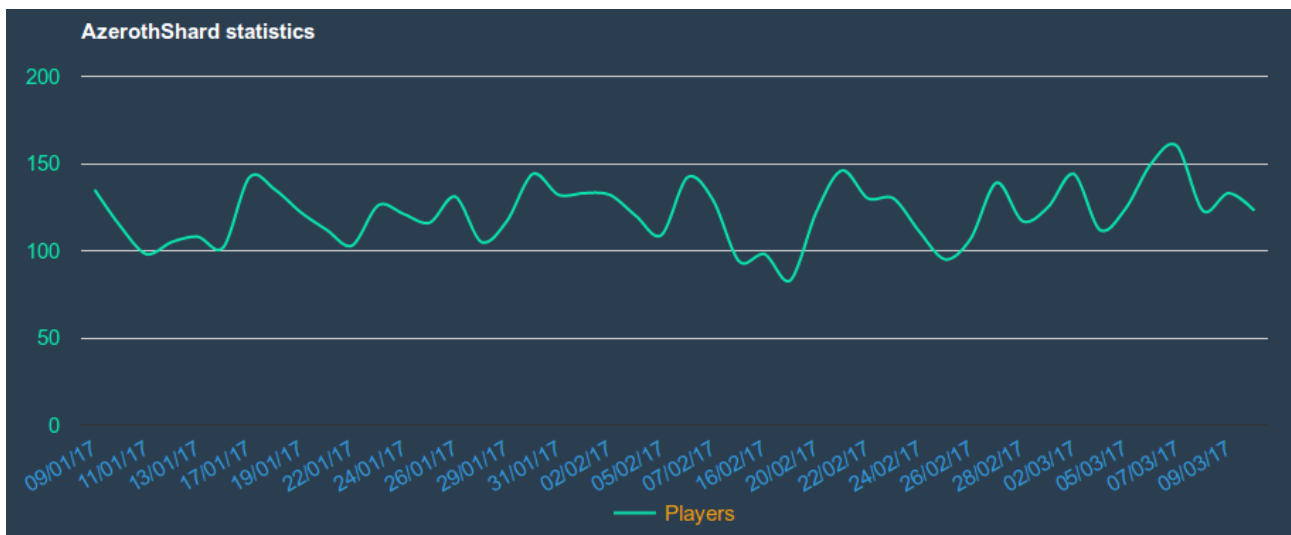
Introduzione

Il progetto è stato scritto in **Javascript**, in particolare è stata scritta una restful API in **NodeJS** che ottiene dei dati da un server, applica uno degli algoritmi relativi alle serie storiche e, tramite le API di Google Chart, genera un grafico (Line Chart), mostrato successivamente in una pagina web tramite un frontend sviluppato in **AngularJS**.

I dati che principalmente l'applicazione elabora sono delle coppie (data – visite) di dati presenti in una matrice, un esempio di applicazione può essere il numero di visite di un sito con associata la relativa data.

L'elaborazione dei dati viene effettuata sfruttando il framework timeseries-analysis che permette di eseguire una serie di algoritmi utilizzati comunemente per le serie storiche, come il *forecasting* per esempio che permette di predire alcuni dati futuri basandosi sui dati passati.

Ecco dei dati campione che verranno utilizzati dal tool Timeseries Statistics



Mean: 115.9622641509434

Standard Deviation: 21.460769178702453

Min: 70

Max: 160

Moving Average & Linear Weight MA

La Moving Average (**Media Mobile**) è uno strumento utilizzato per l'analisi di serie storiche che mostra la media dei valori di una serie basandosi per ogni iterazione sui valori precedentemente analizzati.

Data una serie storica $\{y_t\}, t=1,2,\dots,T$, contenente i valori osservati di una variabile Y dal tempo 1 al tempo T, siano:

- m_1 il numero dei periodi precedenti a t ;
- m_2 il numero dei periodi successivi a t ;
- θ_i il peso da attribuire all' i -esimo valore osservato;

si definisce media mobile al tempo t il valore:

$$mm_t = \frac{1}{k} \sum_{i=-m_1}^{m_2} \theta_i y_{t+1}$$

dove $k = m_1 + m_2 + 1$ è il periodo o l'ordine della media mobile, ed equivale al numero degli addendi.

La media mobile pesata si basa sullo stesso principio solo che considera anche il peso dei dati.

Ecco il relativo codice utilizzato per effettuare i due algoritmi:

Media Mobile (Moving Average)

```
// Moving Average
timeseries.prototype.ma = function(options) {
  options = _.extend({
    period: 12
  }, options);
  var i;
  var j;
  var l = this.data.length;
  var sum = 0;

  // Reset the buffer
  this.buffer = [];

  // Leave the datapoints [0;period[ intact
  this.buffer = this.data.slice(0, options.period);

  for (i=options.period;i<l;i++) {
    sum = 0;
    for (j=options.period;j>0;j--) {
      sum += this.data[i-j][1];
    }
    this.buffer[i] = [this.data[i][0], sum/options.period];
  }
  this.data = this.buffer;
  return this;
}
```

Linear Weight Moving Average

```
timeseries.prototype.lwma = function(options) {
  options = _.extend({
    period: 12
  }, options);
  var i;
  var j;
  var l = this.data.length;
  var sum = 0;
  var n = 0;

  // Reset the buffer
  this.buffer = [];

  // Leave the datapoints [0;period[ intact
  this.buffer = this.data.slice(0, options.period);

  for (i=options.period;i<l;i++) {
    sum = 0;
    n = 0;
    for (j=options.period;j>0;j--) {
      sum += this.data[i-j][1]*j;
      n += j;
    }
    this.buffer[i] = [this.data[i][0], sum/n];
  }
  this.data = this.buffer;
  return this;
}
```



Trend

La stima del **Trend** è una tecnica statistica per facilitare l'interpretazione dei dati.

Quando una serie di misure di un processo vengono trattate come una serie storica, la stima del trend può essere usata per giustificare alcune dichiarazioni delle tendenze dei dati, mettendo in relazione le misure ai tempi in cui si sono verificati.

Questo modello può quindi essere utilizzato per descrivere il comportamento dei dati osservati.

Questa stima del trend è basata sull'algoritmo fatto [John Ehlers](#)

```
// DSL, iTrend
timeseries.prototype.dsp_itrend = function(options) {
  options = _extend({
    alpha: 0.7,
    use: 'main'
  }, options);
  var i;
  var j;
  var l = this.data.length;

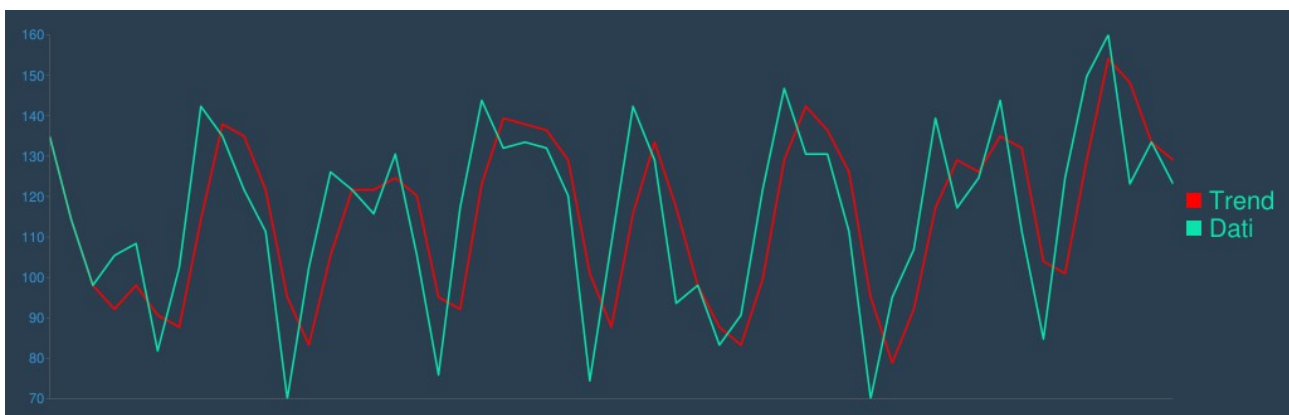
  var trigger = [];

  // Reset the buffer
  this.buffer = [];

  // Leave the datapoints [0;period[ intact
  this.buffer = this.data.slice(0, 3);
  this.trigger = this.data.slice(0, 3);

  for (i=3;i<l;i++) {
    this.buffer[i] = [
      this.data[i][0],
      (options.alpha-(options.alpha*options.alpha)/4)*this.data[i][1] +
      (0.5*(options.alpha*options.alpha)*this.data[i-1][1] - (options.alpha - 0.75*(options.alpha*options.alpha))
      * this.data[i-2][1] + 2*(1-options.alpha)*this.buffer[i-1][1] - (1-options.alpha)*(1-options.alpha)*this.buffer[i-2][1]
    ];
    this.trigger[i] = [
      this.data[i][0],
      2*this.buffer[i][1]-this.buffer[i-2][1]
    ]
  }
  if (options.use == 'trigger') {
    this.data = this.trigger;
  } else {
    this.data = this.buffer;
  }

  return this;
}
```



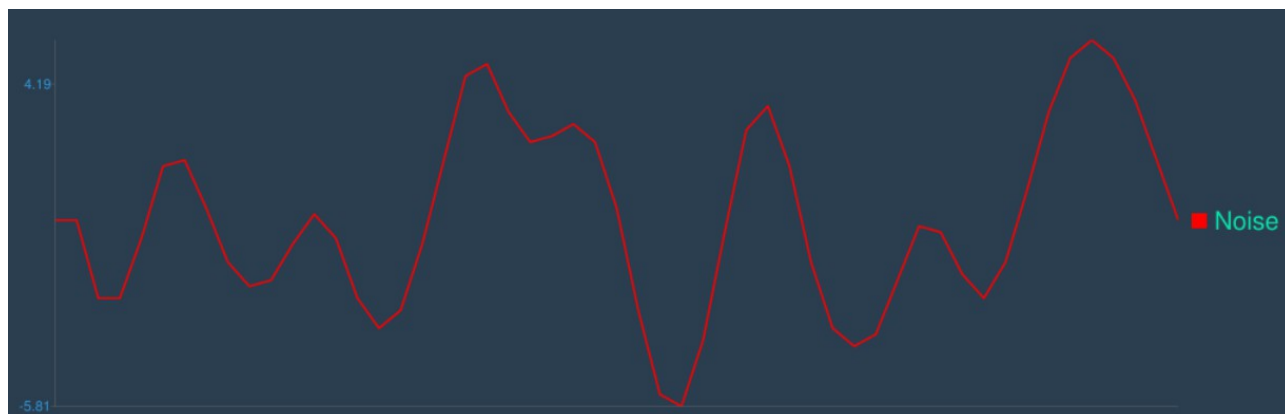
Noise

Il Noise è il “rumore” presente nei dati, ovvero un “residuo” presente nei dati statistici, una variazione *inspiegabile* in un campione.

```
// Extract the noise out of the data
timeseries.prototype.noiseData = function() {
    var i;
    var j;
    var l      = this.data.length;
    var sum    = 0;

    // Reset the buffer
    this.buffer = [];

    for (i=0;i<l;i++) {
        this.buffer[i] = [
            this.data[i][0],
            this.original[i][1]-this.data[i][1]
        ];
    }
    this.data = this.buffer;
    return this;
}
```



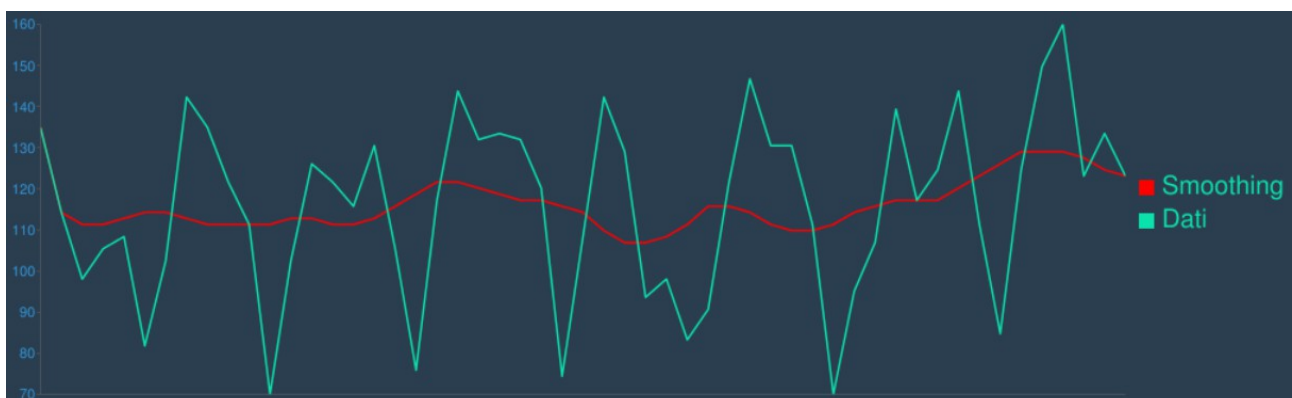
Smoothing

In statistica, lo smoothing di un insieme consiste nell'applicazione di una funzione di filtro il cui scopo è evidenziare i pattern significativi, attenuando il rumore o diversi fenomeni di disturbo legati a fattori di scala molto piccoli.

Si tratta di fare una media tra valori contigui oppure molto vicini nello spazio o nel tempo.

// Iterative Noise Removal

```
timeseries.prototype.smoother = function(options) {  
    options = _.extend({  
        period: 1  
    }, options);  
    var i;  
    var j;  
    var l = this.data.length;  
    var sum = 0;  
  
    // Reset the buffer  
    this.buffer = this.data.slice(0);  
  
    for (j=0;j<options.period;j++) {  
        for (i=3;i<l;i++) {  
            this.buffer[i-1] = [  
                this.buffer[i-1][0],  
                (this.buffer[i-2][1]+this.buffer[i][1])/2  
            ];  
        }  
    }  
    this.data = this.buffer;  
    return this;  
}
```



Forecasting

Il Forecasting (**Previsione**) è il processo che cerca di predire dati futuri basandosi su dati passati e presenti e, comunemente fa uso dell'analisi del trend.

In questa analisi sono stati utilizzati i metodi **Least Square** e **Max Entropy**, per calcolare i coefficienti di **Auto-Regression**.

```
timeseries.prototype.regression_forecast_optimize = function(options) {
  options = _.extend({
    data: this.data,
    maxPct: 0.2,
    maxSampleSize: false
  }, options);

  var l = options.data.length;

  var maxSampleSize = Math.round(l * options.maxPct);
  if (options.maxSampleSize) {
    maxSampleSize = Math.min(maxSampleSize, options.maxSampleSize);
  }

  var maxDegree = Math.round(maxSampleSize);
  var methods = ['ARMaxEntropy', 'ARLeastSquare'];
  var ss; // sample size
  var deg; // degree
  var MSEData = [];
  var i;
  for (i = 0; i < methods.length; i++) {
    for (ss = 3; ss <= maxSampleSize; ss++) {
      for (deg = 1; deg <= maxDegree; deg++) {
        if (deg <= ss) {
          var mse = this.regression_forecast_mse({
            method: methods[i],
            sample: ss,
            degree: deg,
            data: options.data
          });
          console.log("Trying method(" + methods[i] + ") degree(" + deg + ") sample(" + ss + ")(" + mse);
          if (!isNaN(mse)) {
            MSEData.push({
              MSE: mse,
              method: methods[i],
              degree: deg,
              sample: ss
            });
          }
        } else {
          break;
        }
      }
    }
  }

  // Now we sort by MSE
  MSEData = MSEData.sort(function(a, b) {
    return a.MSE > b.MSE;
  });

  console.log("Best Settings: ", MSEData[0]);

  // Return the best settings
  return MSEData[0];
}
```

```
timeseries.prototype.sliding_regression_forecast = function(options) {
  options = _.extend({
    method: 'ARMaxEntropy', // ARMaxEntropy | ARLeastSquare
    sample: 50,
    degree: 5
  }, options);

  var i;
  var j;
  var l = this.data.length;

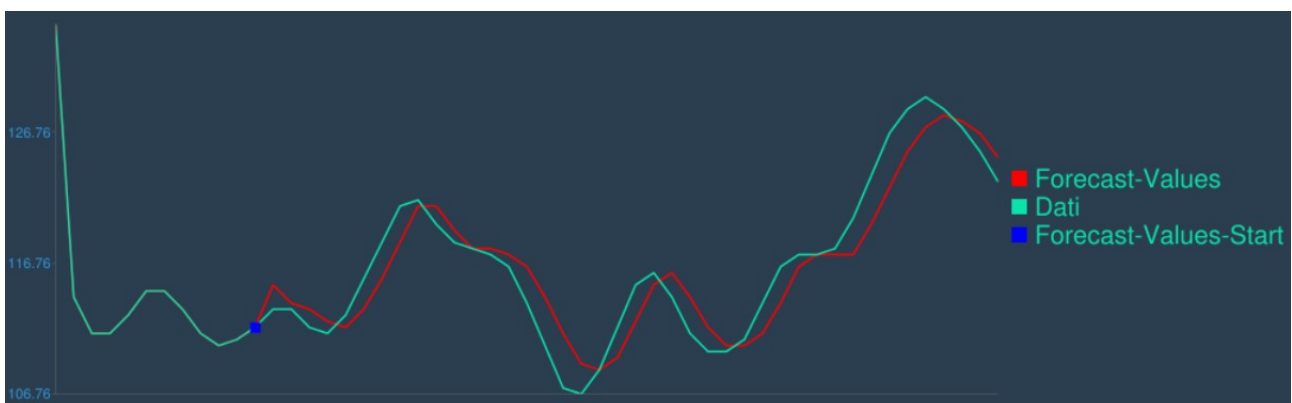
  var mean = this.mean();
  this.offset(-mean);
  var backup = this.clone();
  var buffer = this.clone();

  for (i = options.sample; i < l - 1; i++) {
    var sample = buffer.slice(i - options.sample, i);
    // The current data to process is only a sample of the real data.
    this.data = sample;
    // Get the AR coeffs
    var coeffs = this[options.method]({degree: options.degree});
    buffer[i + 1][1] = 0; // backup[i][1] * 1;
    for (j = 0; j < coeffs.length; j++) {
      if (options.method == 'ARMaxEntropy') {
        buffer[i + 1][1] -= backup[i - j][1] * coeffs[j];
      } else {
        buffer[i + 1][1] += backup[i - j][1] * coeffs[j];
      }
    }
    //buffer[i + 1][1] -
  }

  this.data = buffer;

  // Put back the mean
  this.offset(mean);

  return this;
}
```



Referenze

- Moving Average: https://en.wikipedia.org/wiki/Moving_average
- Trend: http://www.davenewberg.com/Trading/TS_Code/Ehlers_Indicators/iTrend_Ind.html
- Smoothing: <https://en.wikipedia.org/wiki/Smoothing>
- Noise Statistics: https://en.wikipedia.org/wiki/Statistical_noise
- Forecasting: <https://en.wikipedia.org/wiki/Forecasting>
- Auto-Regression https://en.wikipedia.org/wiki/Autoregressive_model
- Timeseries-Analysis framework: <https://www.npmjs.com/package/timeseries-analysis>