# Derivazione dell'algoritmo di punteggio

Emanuele Ricco

11 giugno 2025

## 1 Obiettivo

L'obiettivo di questo calcolo è trovare un algoritmo affine a dati reali, che avendo in entrata un tempo di risposta ed una soglia di difficoltà, fornisca un punteggio: positivo se la prova superata, negativo se la prova non è stata superata.

## 2 Analisi dei dati

### 2.1 Human Benchmark

Per la campionatura dei dati, abbiamo utilizzato le informazioni fornite dal Human Benchmark vedi Figura 1 . Campioniamo i dati per applicare un curve fitting.

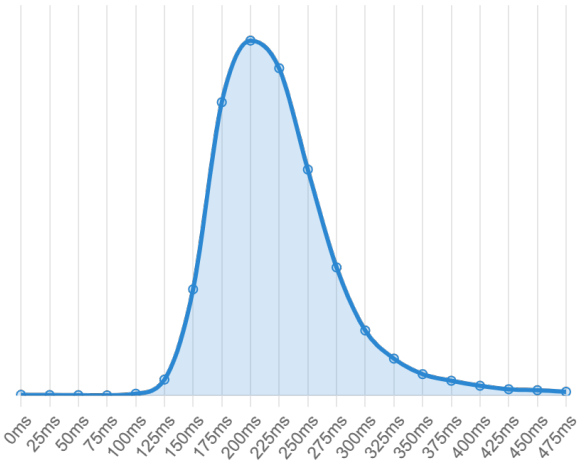


Figura 1: Tempo di risposta degli user su Human Benchmark

$$y = [0; 0; 0; 0; 0; 0.042; 0.296; 0.819; 0.984; 0.911; 0.628; 0.358; 0.183; 0.099; 0.054; 0.038; 0.026; 0.014; 0.016; 0.01];$$

$$x = [0 : 25 : 475]$$

### 2.2 Interpolazione lineare

Eseguiamo un'interpolazione polinomica su MatLab fino al decimo grado, con la funzione polyfit, che utilizza il metodo dei minimi quadrati.

```
y = [0 0 0 0 0 0.042 0.296 0.819 0.984 0.911 0.628 0.358 0.183 0.099
    0.054 0.038 0.026 0.014 0.016 0.01];
x = (0:25:475);
a=10;
b=20;
```

```
elementi= zeros(a,b);

plot(x, y, 'o', 'DisplayName', 'Dati originali'); % punti dati
hold on;

for i=1:a
    elementi(i,:)=polyval(polyfit(x,y,i),x);
    plot(x, elementi(i,:), '-', 'DisplayName', string(i) +' grado');
    % curva di fitting

end

xlabel('ms');
ylabel('valore');
title('Fitting di un polinomio sui dati');
legend('show');
hold off;
```

La Figura 2 mostra i dati campionati e le interpolazioni fino al decimo grado. Notiamo che anche l'approssimazione di grado maggiore presenta delle imprecisioni non trascurabili. La curva deve essere nulla per valore di una varianza maggiore di 60 dalla soglia. Ci troviamo quindi obbligati a studiare un'interpolazione gaussiana, rinunciando ad un costo energetico minore.

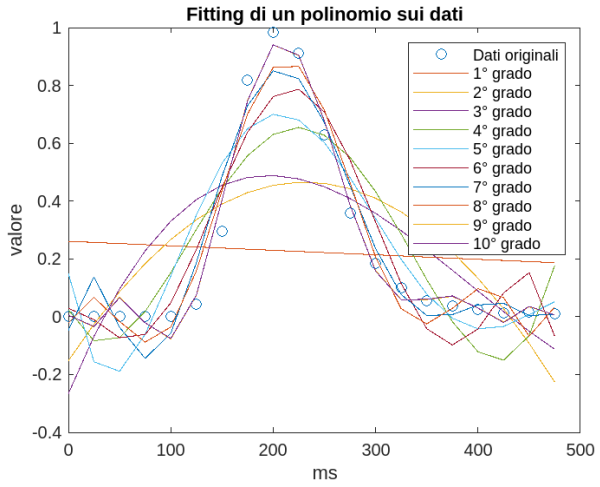


Figura 2: Dati campionati ed interpolazione polinomica fino al decimo grado

## 2.3 Curve Fitter: Interpolazione Gaussiana

MatLab fornisce un programma di interpolazione avanzata utile al nostro caso, inseriamo i dati campionati e scegliamo fitting Gaussiano, Figura 3.

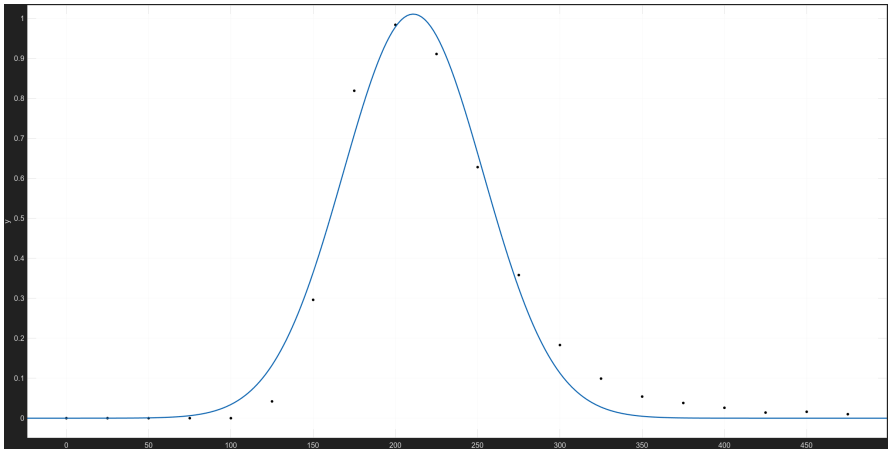

Figura 3: Interpolazione Gaussiana

Il programma considera la formula generica di una curva Gaussiana, variando i valori dei coefficienti $a_1, b_2, c_1$ per trovare l'equazione con gli scarti quadrati minori.

L'equazione generica ed i coefficienti trovati sono riportati in seguito:

$$f(x) = a_1 e^{-(\frac{x-b_1}{c_1})^2}$$

| | Value | Lower | Upper |
|---|---|---|---|
| a1 | 1.0108 | 0.9339 | 1.0877 |
| b1 | 210.8289 | 207.0873 | 214.5704 |
| c1 | 60.2168 | 54.9254 | 65.5081 |

Tabella 1: Coefficienti della curva Gaussiana.

La curva che otteniamo, considerando i valori medi, è quindi:

$$f(x) = 1.0108 e^{-(\frac{x-210.8289}{60.2168})^2}$$

## 2.4 Conformità della curva per l'utilizzo

L'algoritmo che utlizziamo per calcolare il punteggio deve essere:

1. traslabile con un valore in entrata chiamato *sogliams*;
2. fornire valori positivi per tempi di reazione minori di *sogliams* e valori negativi per tempi di reazione maggiori;
3. facilmente computabile, dove possibile.

Per il punto 3, poniamo i coefficienti $a_1 = 1, b_1 = 200, c_1 = 60$, ottenendo così una funzione il cui Codominio è compreso tra 0 ed 1.

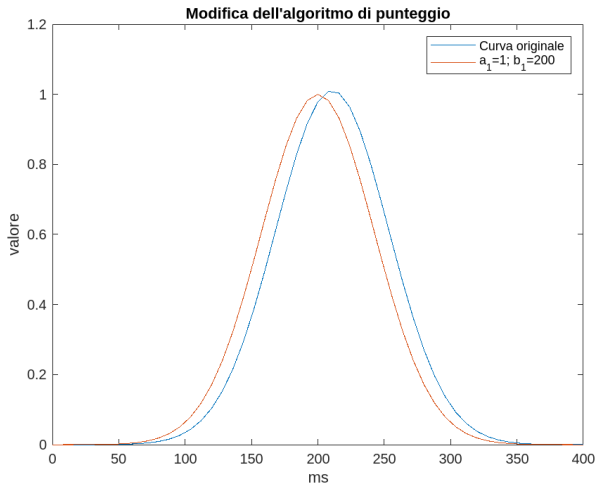


Figura 4: $f(x) = 1e^{-(\frac{x-200}{60})^2}$

Per il punto 2, la polarità, sottraiamo all'esponenziale 1 e poniamo il numeratore pari a x, portando il codominio a valori completamente negativi e la funzione simmetrica all'asse y.

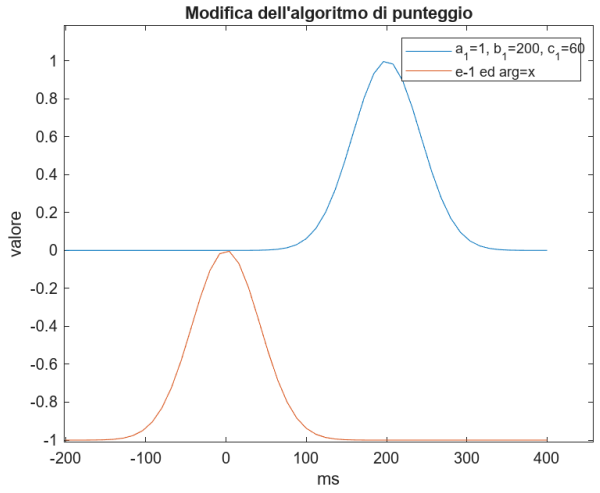


Figura 5: $f(x) = (e^{-(\frac{x}{60})^2} - 1)$

Quindi moltiplichiamo la funzione per una retta $g(x) = x$ , ottenendo così valori positivi a sinistra e valori negativi a destra.

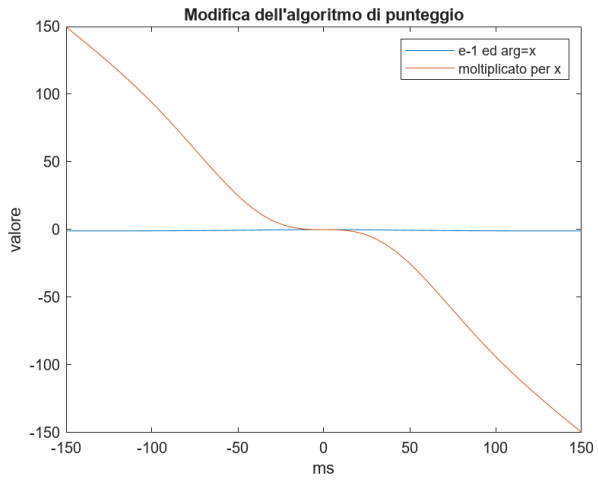


Figura 6: $f(x) = (e^{-(\frac{x}{60})^2} - 1)(x)$

Per il punto 1, poniamo l'argomento della funzione $x \to x - 200$ , in modo che l'intera funzione trasli al variare del valore sottratto ad x.

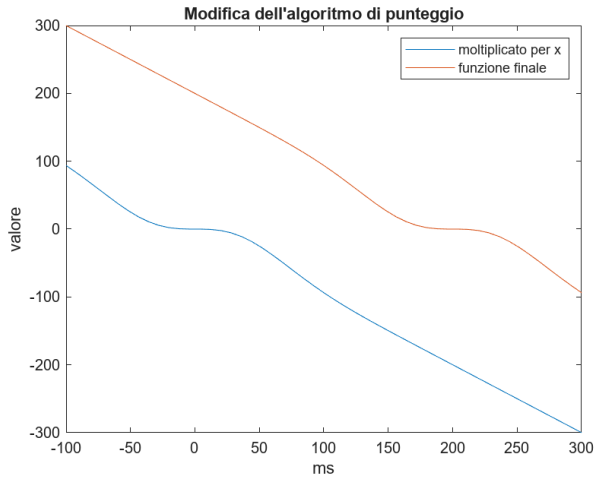


Figura 7: $f(x) = (e^{-(\frac{x-200}{60})^2} - 1)(x - 200)$

Come ultimo passaggio, moltiplichiamo tutta la funzione per 100, ottenendo così valori più prominenti. Sostituiamo $x \to tempoRispostams$ e $200 \to sogliams$ per adattare l'algoritmo al codice.

La funzione finale è:

$$f(x) = 100e^{-(\frac{tempoRispostams - sogliams}{60})^2}(x - sogliams)$$