Passando dalla formalizzazione matematica vista nella sezione precedente giungiamo alla sua implementazione. In algorithm1 è riportato lo pseudocodice che da i dati di input e la loro previsione, calcola i valori dei moltiplicatori lagrangiani e successivamente permette di trovare il valore di W e b.  
L’obiettivo è quindi quello di rappresentare il problema di massimizzazione formalizzato in (6). Primo passaggio è trasformare la ricerca di un massimo nella ricerca di un minimo che si risolve cambiando il segno della nostra funzione. Successivamente vanno definite le matrici, e i constraint che ci permettono di rappresentare il QP:

[FORMULA] min 1/2\*x’Qx + qx lower\_bound <= x <= upper\_bound

Analizziamo quindi i singoli componenti come vengono formalizzati:

* **Q**: da (6) si nota che è rappresentato dalla matrice di Kernel. Dato che alcuni kernel generano una matrice che non risulta essere *positive semi-definite*, basti pensare al caso in cui il kernel scelto contiene *tanh*; la formalizzazione che viene usata all’interno dello pseudocodice ci assicura di ottenere sempre una matrice con questa caratteristica.  
  DUBBIO: Quindi la forma PSD di Q ci assicura qualche proprietà rispetto alla soluzione del QP? COSA???  
  Che quindi imo è la risposta a perché dobbiamo stare attenti e non possiamo usare solo K
* **q**: viene costruita in modo da unire tra loro entrambi gli elementi che contengo *alfa* e *alfa\**
* **linear constraint**: il vincolo lineare presente sui moltiplicatori lagrangiani ci permette di definire lb=0 e ub=C

Data questa formalizzazione tramite algorithm2 verrà risolto il problema di minimizzazione ottenendo i valori dei moltiplicatori lagrangiani. A questo punto sarà quindi possibile calcolare il valore di W e b in modo tale da rappresentare la nostra funzione

[FORMULA

] F(x) = wx +b = SUM{i] (−αi + αi∗)K(xi , x) + b

DOMANDA: è veramente necessario definire w? Che poi non dipende dal valore di x che consideriamo per calcolare l’output? Quindi non abbiamo una forma di w generale o no?

Per determinare il valore di b andremo ad applicare quanto già evidenziato precedentemente nella sezione 1 in (FORMULA DI B AGGIUNGI REF), sarà necessario trovare un elemento che si trovi al di fuori dell’e-tube e rispetto quest’ultimo calcolare il valore.  
DUBBIO: Abbiamo letto diversi paper in letteratura e abbiamo trovato diversi modi in cui viene determinato il valore b. Alcuni dicono che basta usare un punto che sia fuori dall’e-tube, altri invece lo calcolano su tutti i punti per poi fare una media. C’è un metodo più corretto?

DUBBIO BETA: Nel calcolo del Polyak stepsize utilizziamo un parametro *beta* che nel caso di deflection first deve essere minore uguale del coefficiente di deflection. Nelle slide e appunti però non abbiamo visto nessuna formula rispetto la quale è definito, esiste?

DUBBIO GAMMA: Abbiamo visto studiando dagli appunti presi a lezione e dalle slide che *gamma* viene definito nel seguente modo

[FORMULA] argmin { … }

Come lei ha detto non è obbligatorio avere un valore di gamma variabile e che si adatta a quello che è il comportamento della nostra ricerca del minimo, anche se avere un valore adattivo ci permette di ottenere risultati migliori. Dobbiamo quindi analizzare entrambe le possibilità? Nel caso in cui dobbiamo risolvere (FORMULA) andremo a calcolare per ogni iterazione nel nostro algoritmo di ricerca un problema di minimizzazione? Non diventa “pesante” dal punto di vista della computazione?