

Relazione per il progetto di Intelligenza Artificiale e Laboratorio

Docente: **Gianluca Torta**

Gruppo di lavoro:

- **Alessandro Salogni**
Matricola: 906298
E-mail: alessandro.salogni@edu.unito.it
- **Riccardo Perotti**
Matricola: 906237
E-mail: riccardo.perotti@edu.unito.it

1. MPE e MAP

L'implementazione di MPE e MAP è stata testata con reti di diversa morfologia e dimensioni: reti di piccola, media e elevata grandezza, fitte e vicine ad alberi.

Dagli esperimenti effettuati, si è dedotto che:

- 1) Quando MAP viene utilizzato allo stesso modo di MPE (ponendo tutte le variabili che non sono evidenza come variabili MAP, e ponendo le stesse evidenze sia da una parte che dall'altra) si osserva come ci si aspetta che MAP è più lento e inefficiente di MPE: infatti, il primo ha la necessità preliminare di effettuare una Variable Elimination sulle variabili non-MAP, al contrario del secondo. Queste ultime in questo caso sono assenti, quindi Variable Elimination degenera in una moltiplicazione di fattori (un'operazione onerosa, sia in tempo che in spazio, specie se i domini delle variabili sono di elevata grandezza).
- 2) Se in MAP le variabili fossero tutte da massimizzare, l'efficienza dipenderebbe dalla grandezza della rete, dai domini delle variabili e dalle dimensioni delle CPT risultanti dei fattori. Questo perché, come si anticipava al primo punto, la Variable Elimination non eliminerebbe nessuna variabile, e comporterebbe la moltiplicazione di tutti i fattori. Ecco perché, in generale, si evidenzia che più variabili MAP si hanno (e il numero è vicino al numero totale di variabili), più l'efficienza ne risente.
- 3) Contrariamente al punto precedente, meno variabili da massimizzare ci sono in MAP (ovvero il numero è molto lontano dal numero totale di variabili), più l'efficienza dipende dalla morfologia della rete (una computazione di MAP effettuata con maggior numero di variabili MAP può essere migliore di una con numero minore, a causa della morfologia). Se una rete possiede un discreto numero di nodi che hanno un numero elevato di padri, significa che questi nodi avranno delle CPT di grandi dimensioni. Una variabile non-MAP di questo tipo, porta a problemi di efficienza.
- 4) Se le evidenze fossero padri di nodi che a loro volta hanno tanti padri, sia MAP che MPE migliorerebbero: cosa non sempre vero se le evidenze fossero nodi foglia, o i propri figli hanno un numero limitato di padri. Supponiamo un nodo A, che ha i padri B, C e D. Se si ponesse come evidenza ad esempio il nodo B, la CPT di A sarebbe di minor dimensione in confronto a quella che invece si avrebbe se B non fosse evidenza. Ovviamente, se tutti i padri di A fossero evidenza, per A sarebbe il caso migliore, poiché la CPT sarebbe la più piccola possibile. Se al contrario il nodo A non avesse evidenze tra i suoi padri, la situazione senza dubbio sarebbe la peggiore: e

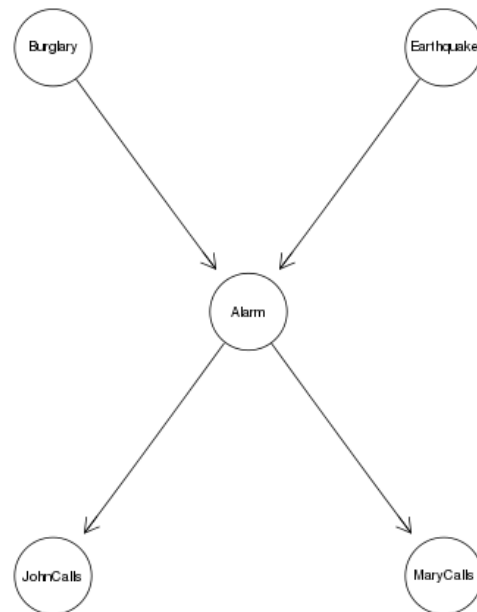
ancor peggio nel momento in cui le variabili abbiano un dominio di cardinalità elevata.

Di conseguenza, una CPT molto grande, sia per MPE che per MAP si traduce in un fattore di grandi dimensioni, che se moltiplicato con altri fattori, provocherebbe onerosi costi di spazio e tempo. Ciò comporta che l'avere buone evidenze all'interno della rete può ridurre sensibilmente i tempi computazionali degli algoritmi.

- 5) L'ordine topologico è un fattore molto importante soprattutto quando si ha a che fare con MAP: infatti, se viene selezionato un ordine topologico particolarmente infelice, nel quale la variabile che si deve eliminare viene trascinata fino alla fine della computazione (perché viene considerata al termine a causa dell'ordine, nonostante possa essere eliminata in fasi precedenti se solo si avesse scelto un differente ordine topologico), provoca il fatto di portarsi dietro fattori di maggiore dimensione, cosa che non accadrebbe se la variabile fosse eliminata precedentemente con un diverso ordine topologico.

Segue una descrizione degli esperimenti effettuati, nei quali si tenta di evidenziare le differenze tra i due algoritmi sulla base delle diverse reti, riportando i tempi di esecuzione (ovviamente non sono sempre uguali a fronte di medesimo input, piccole variazioni sono dovute al calcolatore in uso). In ogni esempio, inoltre, si mettono in evidenza le osservazioni riportate precedentemente.

Rete di piccole dimensioni: Earthquake



Il primo esperimento che viene effettuato, considera il run di MPE sulla rete con un' evidenza sul nodo *MARYCALLS*, e un run di MAP con la stesse evidenza e le variabili rimanenti impostate come MAP (nessuna variabile non-MAP). In questa situazione, le inferenze dovrebbero produrre lo stesso risultato: ci si aspetta però una maggiore inefficienza da parte di MAP, come evidenziato nel punto 1).

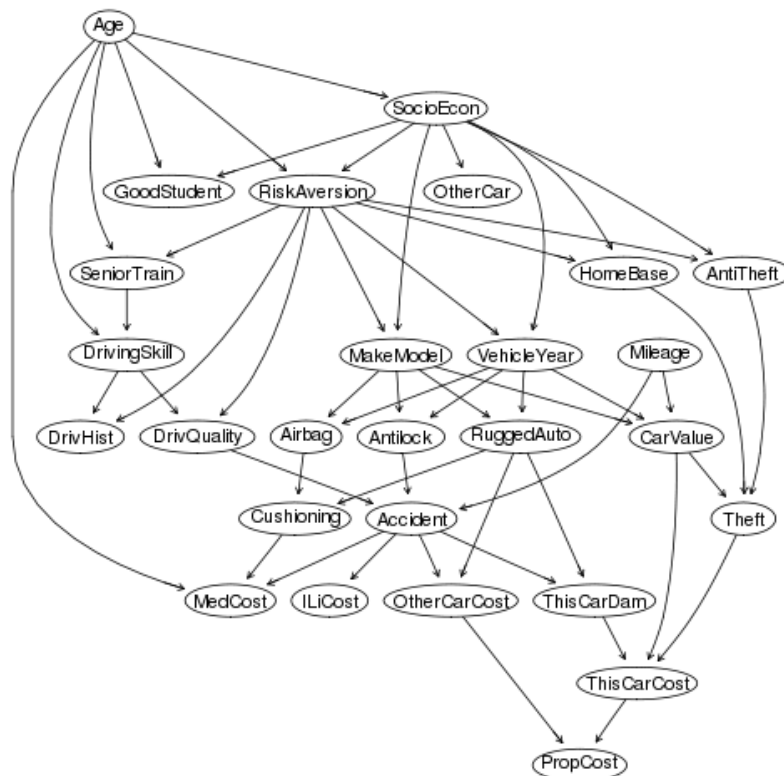
MAP Probability: 0.009207683099999999
BURGLARY -> False
EARTHQUAKE -> False
ALARM -> False
JOHNCALLS -> False
MAP - Execution time in **ms: 7.3159**

MPE Probability: 0.009207683099999999
JOHNCALLS -> False
ALARM -> False
EARTHQUAKE -> False
BURGLARY -> False
MPE - Execution time in **ms: 0.8276**

Figura 1: 1° esperimento su Earthquake.

Già su reti di piccole dimensioni, è possibile constatare in Figura 1 quanto ci si aspettava: MAP è circa 9 volte più lento di MPE.

Rete di medie dimensioni: Insurance



Si cambia rete, e si considera *Insurance*, nella quale si hanno 27 nodi e 52 archi. Si effettuano dapprima 2 esperimenti: nel primo si prova ad impostare 15 variabili MAP, nella seconda invece se ne impostano 12. Come descritto al punto 2), ci si aspetta che il secondo esperimento sia più lento ed inefficiente del primo.

1) MAP Probability: 0.0002926779594995

MILEAGE -> FiftyThou
 AGE -> Adult
 SOCIOECON -> Prole
 GOODSTUDENT -> False
 RISKAVERSION -> Adventurous
 OTHERCAR -> True
 HOMEBASE -> City
 ANTITHEFT -> False
 VEHICLEYEAR -> Older
 SENIORTRAIN -> False
 MAKEMODEL -> FamilySedan
 DRIVINGSKILL -> Normal
 ANTILOCK -> False
 RUGGEDAUTO -> Football
 CARVALUE -> Million
 MAP - Execution time in **ms: 78587.04**

2) MAP Probability: 0.000738694945571

RISKAVERSION -> Adventurous
 OTHERCAR -> True
 HOMEBASE -> City
 ANTITHEFT -> False
 VEHICLEYEAR -> Older
 SENIORTRAIN -> False
 MAKEMODEL -> FamilySedan
 DRIVINGSKILL -> Normal
 ANTILOCK -> False
 RUGGEDAUTO -> Football
 CARVALUE -> Million
 AIRBAG -> True
 MAP - Execution time in **ms: 9689.253**

Figura 2: Esperimenti su MAP, variando numero di variabili MAP.

Come si può notare dai due run di MAP in Figura 2, con tante variabili MAP rispetto alla totalità della rete, il primo run termina dopo 78 secondi, mentre il secondo dopo 10 secondi.

Per evidenziare il punto 3), si riportano altri due esperimenti, con stesso numero di variabili MAP, ossia 1. Si vuole evidenziare quanto la morfologia della rete possa influire su un numero basso di variabili MAP.

1) MAP Probability: 0.0141488798683	2) MAP Probability: 0.0172595716617
CUSHIONING -> Good	ACCIDENT -> Severe
MAP - Execution time in ms: 4905.0444	MAP - Execution time in ms: 678.3178

Figura 3: Esperimenti di MAP sull'influenza della morfologia della rete.

Si osserva in Figura 3 che il primo esperimento, nel quale è stata scelta la variabile *CUSHIONING* come variabile MAP, sia molto più lento rispetto al secondo esperimento, nel quale è stata scelta invece *ACCIDENT*. Si analizza il motivo: *CUSHIONING* ha due variabili padre e un dominio di cardinalità 4, mentre *ACCIDENT* ne ha tre con dominio di cardinalità 4. Inoltre, la prima ha padri con dominio di cardinalità 2 e 3, mentre il secondo ha padri da 3, 2 e 4 valori di dominio. Quindi la CPT della prima sarebbe di 24 valori, mentre quella della seconda di 96 valori. È chiaro che se la variabile MAP è *CUSHIONING*, la Variable Elimination è costretta a fare sum-out su *ACCIDENT*, che ha una CPT molto più grande rispetto a *CUSCHIONING*, e dato che ha tanti figli (4) la deve moltiplicare per ognuno di essi. Viceversa, se si imposta come MAP *ACCIDENT*, il sum-out su *CUSHIONING* è molto più efficiente, e si evita il sum-out su *ACCIDENT* che è molto inefficiente.

Per dimostrare il punto 4), si sperimenta ancora sulla rete *insurance*. Si riprende la variabile precedente *ACCIDENT* e la variabile *CUSHIONING* (scelta come variabile MAP), e si tenta di provare che impostando le evidenze nei padri della prima, la computazione risulta molto più efficiente rispetto a non farlo.

MAP Probability: 0.31104041092	MPE Probability: 9.029351518554931E-4
CUSHIONING -> Poor	OTHERCAR -> True
MAP - Execution time in ms: 5331.3174	... Altre variabili
	MPE - Execution time in ms: 6235.5845

Figura 4: 1° esperimento su evidenze.

Il primo esperimento riporta il run di MAP senza impostare evidenza alcuna, impostando come MAP la variabile *CUSHIONING*, e il run di MPE, anche esso senza evidenze. Come si osserva in Figura 4, entrambi i run appaiono molto lenti, circa sui 5-6 secondi. Si prova ora ad impostare come evidenza i padri di una variabile che ne ha molti, ad esempio *ACCIDENT*. Di questa variabile si impostano come evidenza i padri *DRIVQUALITY*, *ANTILOCK*, *MILEAGE*. La computazione restituisce i risultati in Figura 5.

MAP Probability: 0.002959052182983	MPE Probability: 4.9302534148775946E-6
CUSHIONING -> Excellent	OTHERCAR -> True
MAP - Execution time in ms: 201.0887	... Altre variabili
	MPE - Execution time in ms: 136.6931

Figura 5: 2° esperimento sulle evidenze.

Si nota come i tempi di esecuzione diminuiscano drasticamente. Infatti, come spiegato al punto 4), le evidenze sui padri di *ACCIDENT*, fanno sì che il sum-out su quest'ultima coinvolgano una CPT di piccolissime dimensioni, dato che i padri sono tutte evidenze. Si riporta un terzo esperimento, dove viene asserita come evidenza solamente *MILEAGE*. Ci si aspetta quindi che la computazione risulti migliore della prima, ma peggiore della

MAP Probability: 0.031104041092000004	MPE Probability: 0.0002488367380747049
CUSHIONING -> Poor	OTHERCAR -> True
MAP - Execution time in ms: 1517.5248	... Altre variabili
	MPE - Execution time in ms: 1824.6566

Figura 6: 3° esperimento su evidenze.

seconda. Si conferma quanto appena detto osservando le elaborazioni in Figura 6.

Un quarto esperimento riguardante il punto 4), viene effettuato per evidenziare un altro particolare. Invece di *MILEAGE* (cardinalità di dominio = 4), si imposta come evidenza *ANTILOCK* (cardinalità 2). Ci si aspetta chiaramente un peggioramento rispetto al precedente esempio, ma una situazione comunque migliore rispetto al primo. In Figura 7 si ha la conferma.

MAP Probability: 0.09459374907000001	MPE Probability: 0.0003902191331052106
CUSHIONING -> Good	OTHERCAR -> True
MAP - Execution time in ms: 2748.8408	... Altre variabili
	MPE - Execution time in ms: 3004.3064

Figura 7: 4° esperimento su evidenze.

Un quinto e ultimo esperimento è stato effettuato prendendo come evidenza un nodo foglia (per quanto si vuole provare andrebbe bene un nodo verso il "fondo" della rete – dell'ordine

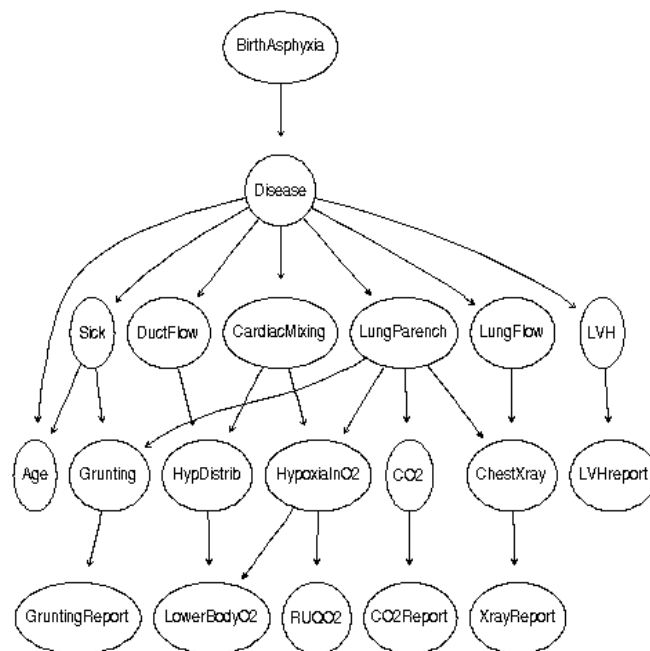
topologico -, e che abbia figli che hanno a loro volta pochi padri). Il nodo scelto è *PROPCOST*.

Si nota in Figura 8 che l'evidenza impostata è totalmente inefficace: infatti, la computazione è totalmente paragonabile a quella che si ha senza avere evidenza alcuna (Figura 5). Come anticipato nel punto 4), avere buone evidenze all'interno della rete permette un notevole miglioramento nell'efficienza dei due algoritmi.

MAP Probability: 0.014148879868314498	MPE Probability: 1.874046357611844E-5
CUSHIONING -> Good	OTHERCAR -> True
MAP - Execution time in ms: 4923.915	... Altre variabili
	MPE - Execution time in ms: 5709.7554

Figura 8: 5° esperimento su evidenze.

Rete di medie dimensioni: Child morfologicamente diversa



Questa rete è di dimensioni simili alla precedente, ma morfologicamente molto più semplice e molto più simile ad un albero. Su questa rete si sono ripetuti gli esperimenti precedenti, e si è osservato quanto segue:

- Il punto 2) si evidenzia nuovamente nella rete;
- Il punto 3) invece si evidenzia notevolmente meno: come si può notare dalla rete, i nodi in generale hanno pochi padri (massimo 2).
- Anche il punto 4) risulta meno importante: non essendoci più nodi con tanti padri, le evidenze giocano un ruolo meno fondamentale.

Rete di grandi dimensioni

Le inferenze MAP e MPE sono state testate su reti di grande dimensione, come la rete *Munin* e *Diabetes*. Si osserva che gli aspetti individuati nei precedenti esempi vengono chiaramente amplificati a causa delle dimensioni della rete. Questo comporta un'elevata complessità in tempo, ma soprattutto in spazio: quest'ultima in generale non permette il termine dell'elaborazione, poiché i fattori diventano così grandi da eccedere lo spazio riservato alla Heap del calcolatore utilizzato.

2. Kalman Filter

Il Kalman Filter è stato sviluppato attraverso l'utilizzo del linguaggio di programmazione R. Quest'ultimo, infatti, permette di lavorare in maniera molto agevole su vettori e matrici – senza dunque dover implementare le varie operazioni – e consente di visualizzare in modo semplice ed efficace i risultati. Proprio per questi motivi si adatta molto bene alla risoluzione del problema proposto.

Il problema preso in considerazione si basa su un modello che considera dei punti bidimensionali – coordinate x e y – in un piano cartesiano, i quali si spostano sullo stesso con un certo errore e vengono osservati con un ulteriore errore. Inoltre, sono state definite le 2 matrici di transizione e di osservazione, F e H , come segue:

$$F = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

In pratica, con F si sta affermando che la posizione di x nel modello all'iterazione successiva dipende anche per uno 0.2 dalla posizione di y , oltre che dalla posizione di x stesso. Per quanto riguarda y , esso dipende solo dalla sua precedente posizione. Con H si afferma invece che sia l'osservazione di x che y dipendono solo dalla loro posizione nel modello.

Osservazioni ed esperimenti sul Kalman Filter

Una volta definito il problema e le matrici F e H , si è sperimentato il filtro implementato con differenti rumori di osservazione e di transizione.

Per prima cosa abbiamo testato i casi in cui questi errori sono simili fra di loro:

- ❖ **Errore nullo:** in questo caso si osserva come l'algoritmo esegua delle stime perfette dei valori, dato appunto il fatto che le osservazioni sono uguali ai punti reali del modello;
- ❖ **Errore medio:** in questo caso, con matrici diagonali di covarianza pari a 10, si osserva che la stima comincia ad allontanarsi dei valori reali per posizionarsi in una via di mezzo tra i punti del modello e i punti osservati (esempio in Figura 9);

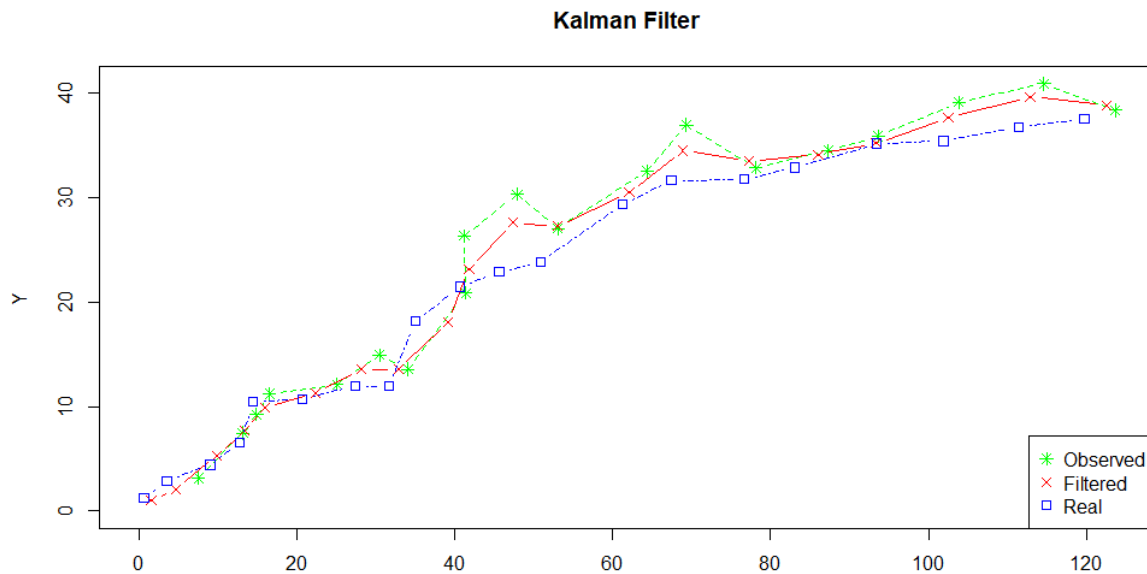


Figura 9: Matrici di covarianza con errore medio.

- ❖ **Errore alto:** in questo caso, come si nota in Figura 10, con matrici diagonali di covarianza pari a 1000 si osserva come la stima tenda a posizionarsi sempre tra i due valori, anche se si nota una tendenza maggiore nel credere all'osservazione rispetto che ai punti reali.

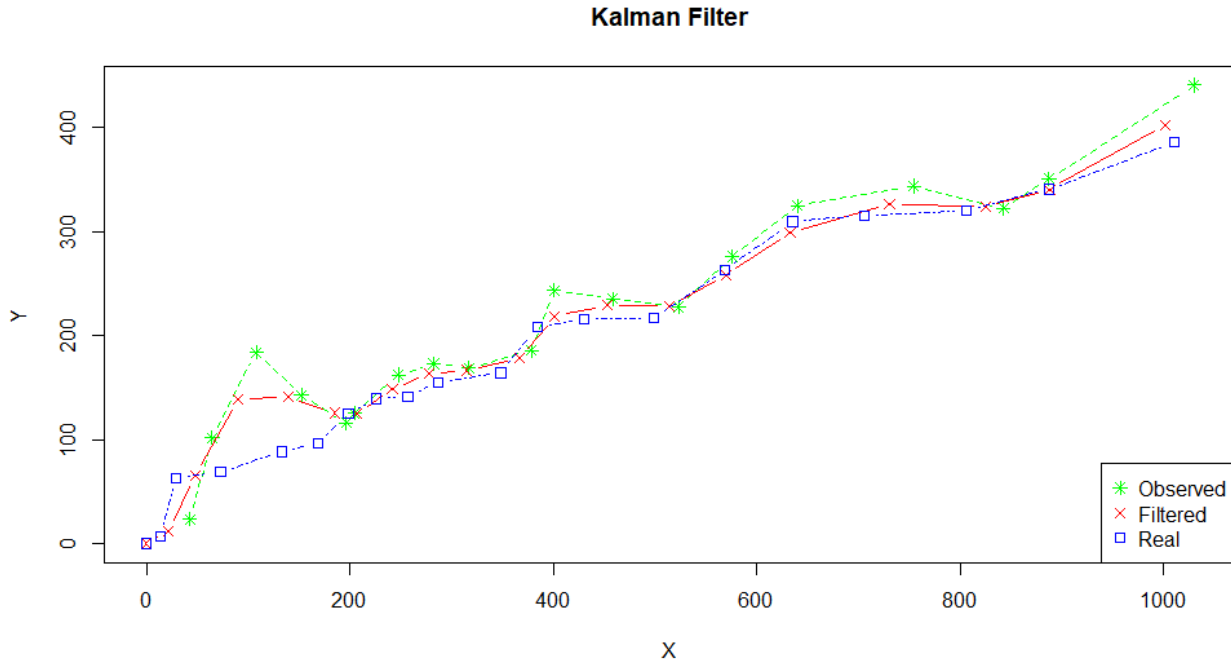


Figura 10: Matrici di covarianza con errore alto.

In seconda battuta si è proceduto all'analisi delle stime ottenute quando gli errori di transizione e di osservazione assumono valori tra loro differenti:

- ❖ **Errore di transizione alto e errore di osservazione nullo:** in questo caso si osserva come i valori filtrati vadano a sovrapporsi esattamente ai valori reali di processo dato

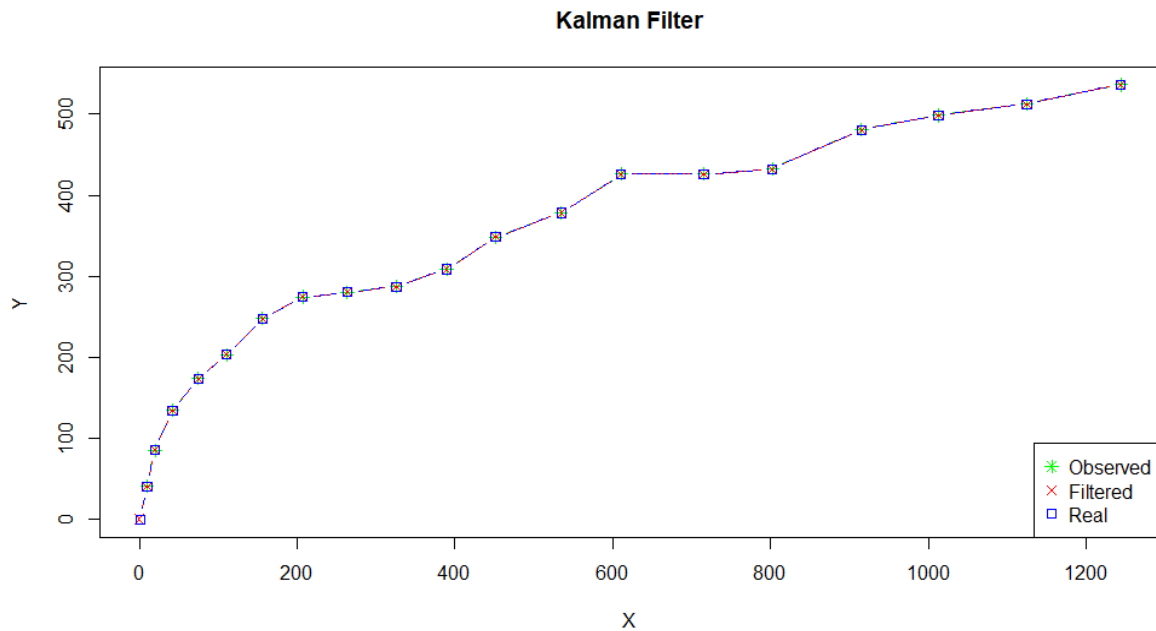


Figura 11: Errore di transizione alto e errore di osservazione nullo.

che l'osservazione che viene fatta è perfetta. Questa stima è dovuta al fatto che si crede esattamente a quanto osservato dato che il Kalman Gain risulta essere perfettamente uguale ad 1. Inoltre, si osserva come i valori reali di processo tendano ad allontanarsi fra di loro a causa dell'errore elevato. Un esempio è riportato in Figura 11. Infine, si osserva come al variare della matrice di covarianza della stima al tempo 0 il risultato stimato non subisca cambiamenti e dalla seconda iterazione in poi si sovrapponga esattamente ai valori osservati;

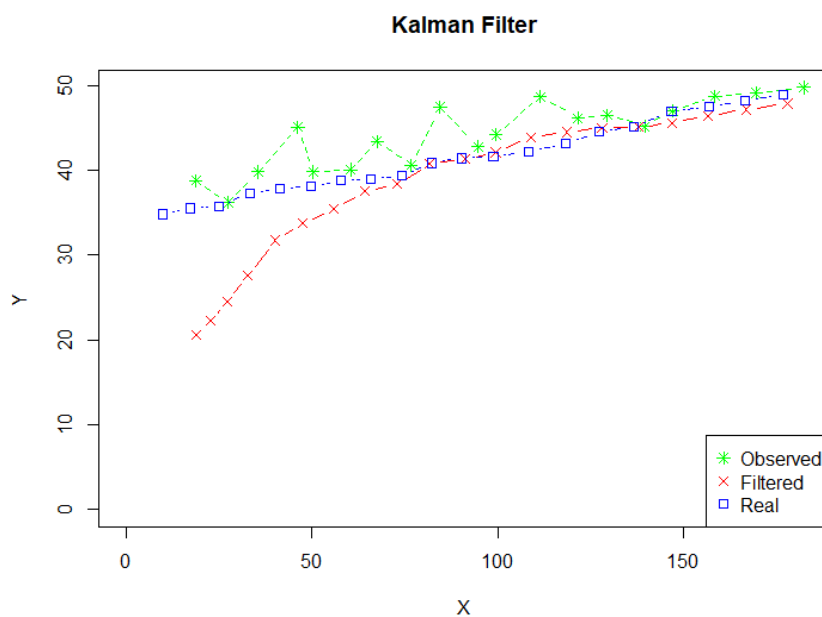


Figura 12: Matrice di covarianza della stima iniziale nulla, errore di transizione basso, errore di osservazione medio.

- ❖ **Errore di transizione basso e errore di osservazione medio:** in questo caso, con matrici di covarianza uguali a 1 e 10, si osservano situazioni differenti rispetto al cambiamento della matrice di covarianza della stima iniziale;
 - **Matrice di covarianza della stima iniziale nulla:** il filtro tende a raggiungere i valori reali in più iterazioni (Figura 12);
 - **Matrice di covarianza della stima iniziale alta:** il filtro, in generale, è molto più veloce nel numero di iterazioni a raggiungere la stima dei valori reali (Figura 13);
- ❖ **Errore di transizione nullo e errore di osservazione alto:** in questo caso si osserva come la stima ottenuta non consideri i valori osservati, ma tenda a fidarsi maggiormente della previsione precedente. Si possono distinguere differenti situazioni, dovute ad una matrice di covarianza della stima iniziale differente:
 - **Matrice di covarianza della stima iniziale nulla:** in questo caso si osserva come il Kalman Gain sia pari a 0 e il filtro si fidi esattamente della stima precedente creando una linea retta orizzontale (Figura 14);
 - **Matrice di covarianza della stima iniziale media:** in questo caso si osserva come il filtro si fidi ancora molto della stima precedente, anche se ora si nota una leggera tendenza nell'avvicinarsi piano piano ai valori reali: non viene più creata una linea retta a partire dai volari stimati (Figura 15);

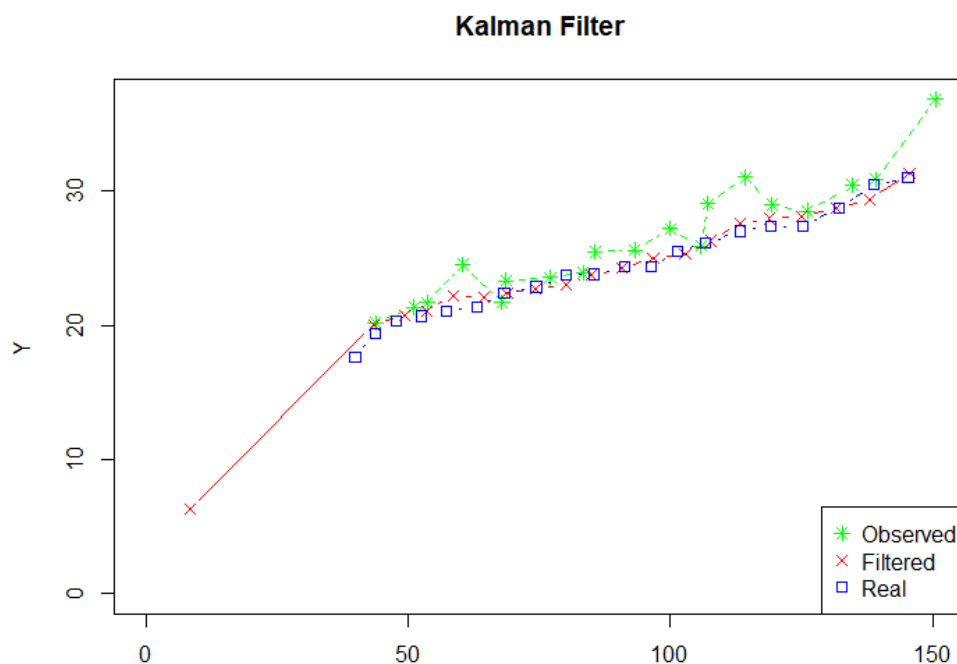


Figura 13: Matrice di covarianza della stima iniziale alta, errore di transizione basso, errore di osservazione medio.

- **Matrice di covarianza della stima iniziale alta:** in questo caso si osserva come il filtro tenda nuovamente a fidarsi delle osservazione, restando molto più distante dai punti del modello (Figura 16).

Oltre a sperimentare diverse combinazioni di errori di transizione e di osservazione, sono stati considerati anche i cambiamenti derivanti dalla variazione della matrice di covarianza del punto iniziale. Si è notato come avere una matrice di covarianza del punto iniziale con

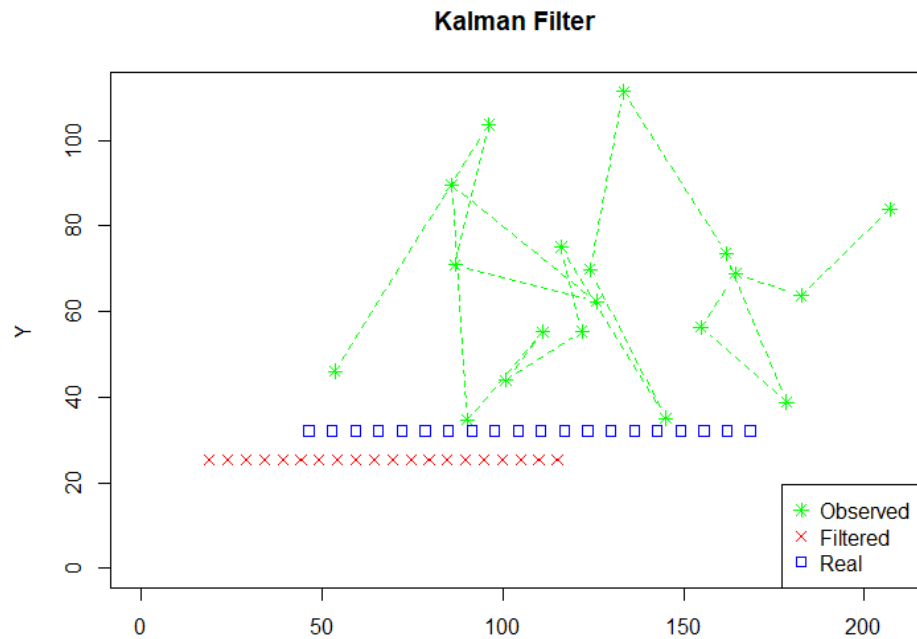


Figura 14: Matrice di covarianza della stima iniziale nulla, errore di transizione nullo e errore di osservazione alto.

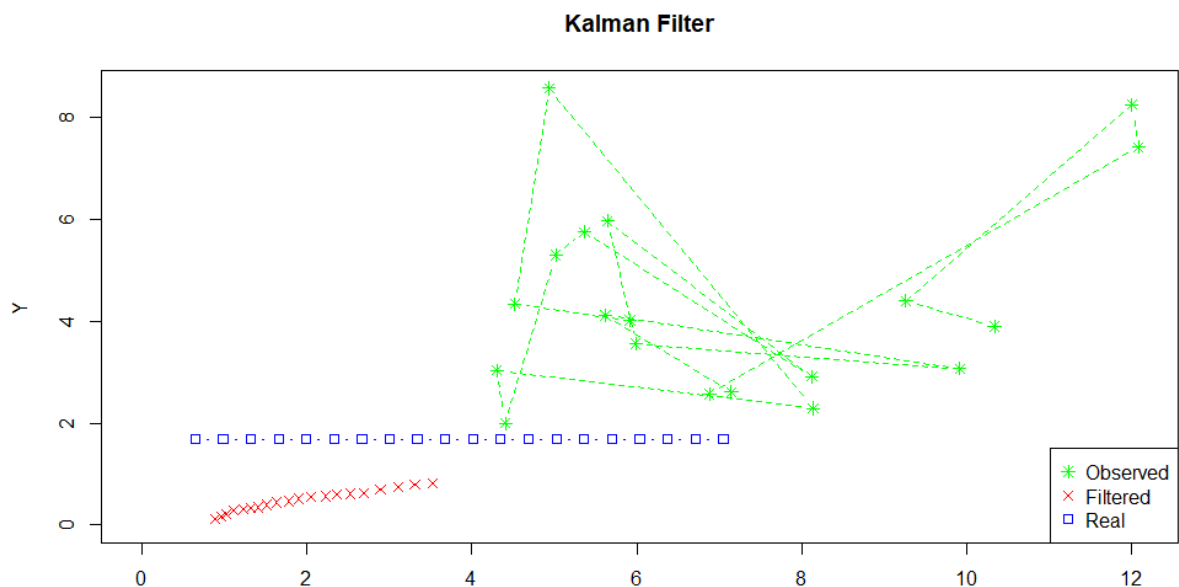


Figura 15: Matrice di covarianza della stima iniziale media, errore di transizione nullo e errore di osservazione alto.

valori molto bassi faccia avvicinare molto il primo punto del modello e quello stimato dal filtro; ovviamente ciò è dovuto al fatto che l'errore di estrazione del primo valore è basso e quindi i valore random saranno molto più vicini.

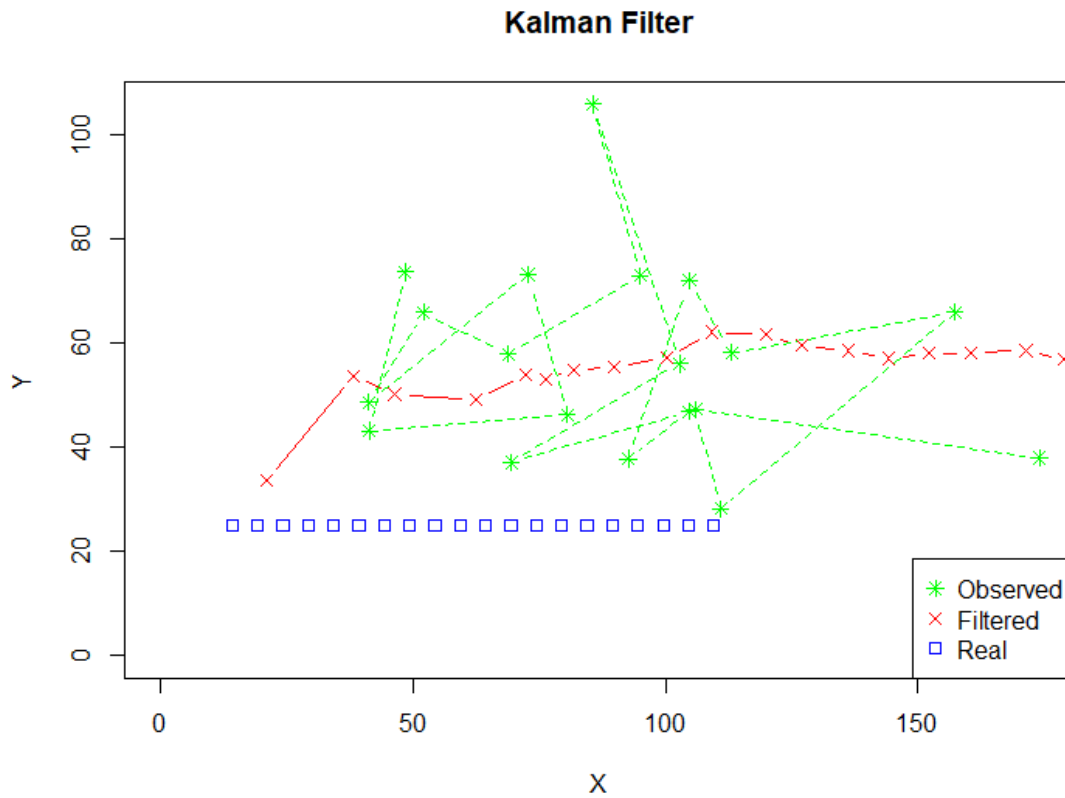


Figura 16: Matrice di covarianza della stima iniziale alta, transizione con errore nullo, osservazione con errore alto.