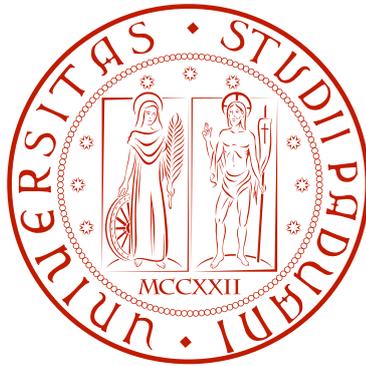


UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE



CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

**STUDIO DI ALGORITMI PER LA
COMPRESSIONE DI SEGNALI SPAZIO
E TEMPO VARIANTI IN RETI DI
SENSORI RADIO**

Relatore: **Ch.mo Prof. Michele Rossi**

Laureando: **Francesco Agnolazza**

Anno Accademico 2013-2014

A mio nonno Chechi

Sommario

Le reti di sensori radio stanno acquisendo sempre maggior importanza nell'ambito delle reti di telecomunicazioni. Esse sono formate da un insieme di sensori di modeste dimensioni e tipicamente alimentati a batteria, ognuno dei quali è equipaggiato con un microprocessore, uno o più sensori e un modulo per le comunicazioni radio.

Grazie ai progressi della microelettronica e delle tecnologie radio, i sensori sono piccoli e posso essere posizionati a basso costo su diversi tipi di ambienti per monitorare nello spazio e nel tempo variazioni di grandezze fisiche quali la temperatura, l'umidità o la pressione. I nodi sensori tuttavia sono vincolati dalle loro risorse limitate, in particolar modo dal punto di vista energetico. Per come vengono sviluppati oggi tali dispositivi, è risaputo che la comunicazione è il processo più dispendioso e quindi il design delle metodologie per raccogliere i dati viene considerato come il problema centrale per questa tipologia di reti.

Un modo efficiente per vincere questa sfida è approssimare tramite modelli matematici e statistici l'evoluzione delle misure rilevate dai sensori nello spazio e/o nel tempo. Infatti, ogniqualvolta un modello teorico può essere usato in sostituzione delle misure reali, si possono ottenere guadagni significativi in termini di risparmio energetico nelle comunicazioni perché è possibile trasmettere un insieme di valori molto ristretto rispetto all'insieme di misure reali. Poiché nella maggior parte dei casi non ci sono informazioni a priori sulle misure prese dai sensori, il modello deve cercare di ottenerle in maniera automatica tramite l'utilizzo di tecniche di stima (parametrica e non), che

permettono di modellare le variazioni delle misure future sulla base di quelle passate.

Il contributo di questa tesi è quello di studiare alcune strategie per la compressione dei dati in reti di sensori radio utilizzando la tecnica del *Compressive Sensing* (CS) in combinazione con la *Principal Component Analysis* (PCA) per poi confrontarle con gli altri algoritmi proposti in letteratura. La peculiarità di questo approccio è l'abbattimento dei costi computazionali relativi alla compressione dei dati sul singolo nodo che è reso possibile grazie allo studio di alcune proprietà statistiche del segnale quali la sua matrice di covarianza. Questo permette di demandare tutto il carico di lavoro per la ricostruzione del segnale alla stazione radio base, garantendo così una maggior longevità alla rete.

Indice

1	Introduzione	1
1.1	Reti di sensori radio	1
1.1.1	Scenario	2
1.2	Compressione dati	3
1.2.1	Compressione dati distribuita	4
1.2.2	Compressione dati locale	7
2	Generazione del Segnale	9
2.1	Il segnale	9
2.1.1	Generazione segnale sintetico	10
3	Principal Component Analysis	13
3.1	Principio di funzionamento della PCA	13
3.2	Studio della varianza nella PCA	15
3.3	Matrice di covarianza	17
3.3.1	Diagonalizzare la matrice di covarianza	19
3.4	Risolvere la PCA tramite decomposizione in autovettori	21
4	Compressive Sensing	23
4.1	Formulazione matematica	24
4.1.1	Restricted Isometry Property	25
4.2	Soluzioni sparse di sistemi mal posti	26

INDICE

5	Framework SCoRe1	27
5.1	Descrizione	27
5.2	Utilizzo congiunto di CS e PCA	29
5.3	Approccio Bayesiano	31
6	Risultati	33
6.1	Dimensione Training Set ottima	33
6.2	Correlazione spaziale e temporale	35
6.3	Consumo energetico	37
6.4	Clustering	41
6.5	Confronto tra i vari algoritmi	43
7	Conclusioni	45
7.1	Sviluppi futuri	46
	Bibliografia	49

Capitolo 1

Introduzione

In questo capitolo vengono trattati gli argomenti generali in cui si inserisce il lavoro di questa tesi.

1.1 Reti di sensori radio

Le reti di sensori radio sono una classe di reti emergenti capaci di monitorare diverse tipologie di ambienti con un'alta accuratezza spazio-temporale. La rete è composta da piccoli dispositivi chiamati sensori radio o nodi, tipicamente equipaggiati con un microprocessore, una memoria, una radio, una batteria e uno o più sensori di grandezze fisiche come la temperatura, l'umidità, la luce o il suono.

La trasmissione dei dati misurati dal sensore verso la stazione base incontra però diversi problemi legati alla limitatezza delle risorse a disposizione dei nodi in termini di energia, capacità di calcolo, disponibilità di memoria e throughput per l'invio e la ricezione dei dati.

In particolare il modulo che si occupa della comunicazione tra il sensore e la stazione radio base riveste un ruolo molto importante poiché, da una parte l'ammontare dei dati che può essere trasmesso al di fuori della rete è limitato dalla sua capacità e dall'altra la comunicazione radio è il processo energeticamente più dispendioso, identificato in molte casi come il fattore primario

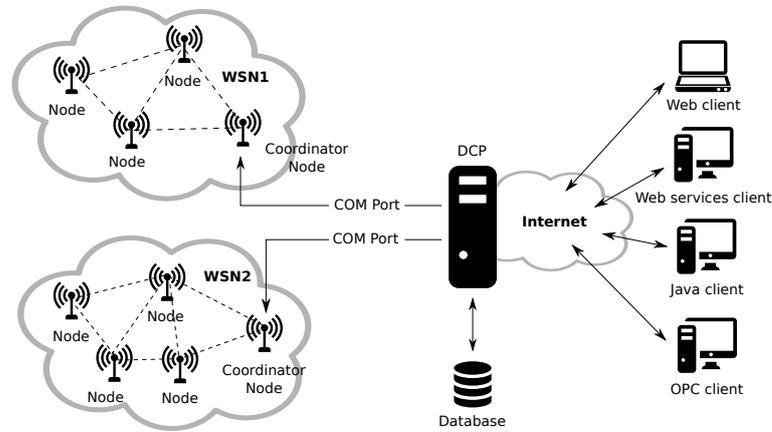


Figura 1.1: Rete di sensori radio

di riduzione del ciclo di vita del sensore e quindi della rete [1].

L'uso di risorse ad energia rinnovabile non è sempre possibile adottarlo e il cambio delle batterie è un'operazione di manutenzione costosa e non sempre attuabile. Per questo motivo, le risorse energetiche dei nodi radio sono nella maggior parte dei casi limitate all'energia - finita - presente nelle loro batterie al momento del posizionamento. Questo fatto rende il design di strategie per la raccolta dei dati in modo efficiente un bisogno primario per estendere la durata della vita delle WSN (*Wireless Sensor Network*).

1.1.1 Scenario

In un tipico scenario per una rete di sensori radio si posiziona un insieme di sensori in un ambiente come un edificio, una città o una foresta e si usano le misure raccolte per monitorare o registrare l'evoluzione di un fenomeno fisico sullo spazio e nel tempo. La rete è tipicamente connessa ad una stazione radio base (chiamata anche *Data Collection Point*) tramite un albero di instradamento come illustrato nella figura 1.1. La stazione base permette di centralizzare la raccolta dei dati dalla rete e agisce da *gateway* tra la rete di sensori e Internet, da cui si può osservare in modo remoto il fenomeno.

Un'applicazione promettente per le WSN è il monitoraggio ambientale, dove ci si aspetta che il loro utilizzo rivoluzioni la qualità delle ricerche scientifiche

per campi quali la biologia o l'ecologia per esempio. Il loro costo ragionevolmente basso e i ridotti requisiti infrastrutturali permettono una raccolta di informazioni su aree vaste fornendo dati su scale spazio-temporali che una volta non era possibile ottenere. Inoltre, le capacità radio dei sensori rendono possibile monitorare in tempo reale l'evoluzione dei fenomeni che, combinate con la forte predisposizione delle WSN ad essere distribuite - ad esempio su ambienti ostili o inaccessibili all'uomo quali potrebbero essere foreste pluviali o vulcani - permettono all'utente finale di reagire in tempi brevi agli eventi. Infine, grazie alle dimensioni molto ridotte dei nodi sensori, si riduce considerevolmente l'impatto ambientale causato dai tradizionali sistemi di monitoraggio.

1.2 Compressione dati

È evidente che, selezionata la tecnologia hardware con cui è realizzata la rete di sensori, gli aspetti dove è possibile andare ad agire per attuare le politiche di risparmio energetico sono principalmente i protocolli e le strategie utilizzate per la trasmissione dei dati acquisiti dai sensori e inviati al DCP (*Data Collection Point*).

In letteratura [6] vengono distinti due approcci di natura diversa che mirano a raggiungere questo obiettivo: *duty cycle* e *in-network processing*.

Nell'approccio *duty cycle* si coordina e si stabilisce lo *scheduling* per la sveglia e lo spegnimento dei nodi della rete, andando quindi a risparmiare energia sfruttando lo spegnimento periodico dell'interfaccia radio. D'altro canto, l'approccio *in-network* risolve il problema riducendo l'ammontare dei dati che vengono trasmessi attraverso tecniche di compressione e aggregazione degli stessi.

Le tecniche di aggregazione, tipicamente utilizzate in reti di sensori dense e multi-hop, coinvolgono modi differenti di instradare i pacchetti combinandoli insieme sfruttando le caratteristiche estratte e le statistiche dei dati, come i valori massimo, minimo e medio [2]. Questi dati aggregati sono poi inviati

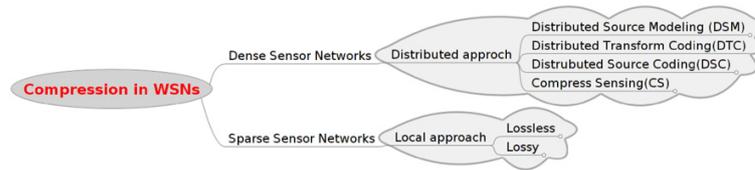


Figura 1.2: Classi algoritmi per compressione dei dati

al DCP. Tuttavia questa tecnica può perdere molta dell'informazione della struttura originale dei dati estratti perché fornisce solo statistiche grezze senza, ad esempio, variazioni locali come la distribuzione dei dati in un'area [3]. Questo fatto viene risolto in diversi articoli andando a studiare alcune informazioni ricavate dalla rete come la differenza tra due misure consecutive, la posizione dei nodi e la capacità della rete stessa che vanno poi ad influire sulle regole e i coefficienti che determinano il grado di aggregazione dei dati. Relativamente alle tecniche di compressione dei dati, vengono classificate in due classi differenti: *distributed data compression approach* e *local data compression approach*. Le prime sono utilizzate per risolvere i problemi di perdita di informazione sulla struttura dei dati presenti nelle tecniche di aggregazione e agiscono distribuendo un algoritmo di compressione attraverso la rete. Le ultime funzionano bene su reti di sensori con una topologia sparsa e la compressione viene realizzata localmente in ogni nodo, indipendentemente dal resto della rete.

Si da ora una breve descrizione di tali algoritmi, alcuni dei quali saranno oggetto di discussione in questo lavoro.

1.2.1 Compressione dati distribuita

Le principali classi di algoritmi per la compressione dati distribuita, tipicamente applicati in reti di sensori dense, sono: *Distributed Source Modeling* (DSM), *Distributed Transform Coding* (DTC), *Distributed Source Coding* (DSC) e *Compressive Sensing* (CS).

Distributed Source Modeling

DSM è una tecnica che mira a cercare una funzione o un modello che approssimi bene un insieme di misure acquisite da uno specifico gruppo di sensori usando due diverse strategie: *parametric modeling* e *non-parametric modeling*. La sostanziale differenza tra i due modelli è che nel primo caso l'algoritmo tratta i dati ricevuti dai sensori come processi aleatori che deve stimare in modo ottimo conoscendo i suoi parametri statistici, come la media e la varianza. Chiaramente questa tecnica funziona molto bene quando la struttura statistica del processo casuale osservato è nota. Nel secondo caso, per rappresentare i valori dei sensori viene utilizzata una regressione basata sul *kernel*¹ dove i coefficienti sono ricavati trattando i dati come input-output di una qualche funzione deterministica. In questo caso, è richiesta una conoscenza a priori sulla natura dei dati del processo casuale molto limitata e la tecnica è considerata molto robusta.

Distributed Transform Coding

In DTC si decompone l'output di una sorgente in componenti (o coefficienti) che sono poi codificati secondo le loro caratteristiche proprie. Ci sono molte tecniche note come la trasformata di *Karhunen-Loeve*, la trasformata del coseno (*DCT*) e la trasformata delle *wavelet* molto diffusi in applicazioni senza vincoli energetici come algoritmi per la compressione di immagini, audio e video. Il motivo per cui questo tipo di algoritmi è raramente implementato nelle WSN, se non nelle *Wireless Multimedia Sensor Networks*, deriva dal fatto che spesso richiede che uno dei nodi sensori per calcolare i valori dei coefficienti debba avere la conoscenza di tutte le misurazioni della rete. Questo andrebbe ad incrementare notevolmente le comunicazioni inter-nodo che si ripercuotono in alti costi di comunicazione e elevato consumo energetico.

Tuttavia, per adattare questi classici algoritmi a WSN tradizionali, in let-

¹Insieme dei punti che vengono annullati da una funzione

Introduzione

teratura ci sono state numerose varianti volte ad approssimare o modificare queste versioni. Alcuni esempi sono discussi in [4] e [5].

Se si prende come esempio la DCT, essa tende a distribuire l'energia del segnale nei primi coefficienti della trasformata. Partendo dunque da una serie temporale $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, viene calcolata la DCT corrispondente e al DCP vengono inviati i primi T' coefficienti, dove $T' < T$ determina il livello di compressione e allo stesso tempo l'accuratezza della rappresentazione.

Distributed Source Coding

DSC è un approccio molto popolare per la compressione dei dati nelle reti di sensori radio. Questa tecnica segue il noto teorema di Slepian e Wolf che dimostra che la codifica separata è efficiente tanto quanto la codifica combinata per compressioni senza perdita (*lossless*). Risultati analoghi sono stati ottenuti anche da Wyner e Ziv (1976) relativamente a codifica con perdita di informazione di sorgenti Gaussiane combinate. L'idea di fondo è che se due nodi X e Y sono correlati e c'è la necessità di comprimere i dati di X senza perdita di informazioni con Y (detto anche *side information*) noto al decodificatore, ma non al codificatore, lo si può fare usando $H(X|Y)$ bit - con H funzione entropia. Esattamente come nello caso in cui il codificatore conosca Y .

Compressive Sensing

CS è una tecnica emergente basata su una teoria del campionamento che opera la compressione senza fare riferimento a nessuna conoscenza specifica a priori o assunzione sui segnali. Questa tecnica basa la sua analisi sulla rappresentazione sparsa di un segnale n -dimensionale \mathbf{x} . Un vettore \mathbf{x} è detto p -sparso se ha al più p componenti non-nulle, dove $p < n$. Quello che viene dimostrato in CS è che questo vettore può essere ricostruito da un piccolo numero di proiezioni lineari di campioni casuali. Nel caso in cui il segnale non abbia le suddette caratteristiche di sparsità esistono tecniche, descritte per esempio in [12] e [13], che riescono a disegnare una base di sparsificazione ad-

hoc basandosi sulle statistiche spazio-temporali dei segnali. In questo modo si riescono a sfruttare comunque le tecniche di ottimizzazione convessa presenti in CS per la ricostruzione completa dei dati tramite un sottocampionamento casuale del segnale.

1.2.2 Compressione dati locale

Se si considerano gli algoritmi che fanno compressione dati locale su ogni singolo sensore senza alcun tipo di collaborazione distribuita tra gli altri nodi della rete, è intuibile che essi sfruttino solamente la correlazione temporale dei dati e non dipendano dalla loro topologia. Per questa ragione questo tipo di algoritmi funzionano bene su reti sparse che hanno una bassa correlazione spaziale. Si descrivono ora alcuni di questi algoritmi, dividendoli secondo le proprietà di perdita o meno dell'informazione.

Lossless

In questa categoria gli algoritmi più diffusi sono basati su dizionario e sono utilizzati per comprimere qualunque tipo di dato. Tali algoritmi possono essere dizionari statici con lunghezza di codice fissa o dizionari adattivi con lunghezza di codice variabile. Alcuni esempi degli algoritmi più noti sono *LZ77*, *LZ78* e *LZW*. Ciò nonostante è stato dimostrato in [1] che queste tecniche di compressione non sono adatte in contesti che prevedono nodi sensori in quanto i requisiti computazionali e di memoria eccedono quelli che sono le reali capacità dei sensori in commercio. Tuttavia, come per DTC, si è cercato di adattare questi algoritmi creando delle versioni modificate ad-hoc per reti di sensori radio. Alcuni esempi sono *S-LZW* e *miniLZO*.

Lossy

Anche se gli algoritmi lossy possono causare perdita di informazione durante i processi di compressione e decompressione dei dati, permettono al codificatore di aumentare la *compression rate* che è più alta rispetto alla

Introduzione

compressione lossless. Questi approcci sono particolarmente vantaggiosi per applicazione che non richiedono particolari requisiti di precisione. Gli esempi più noti sono LTC (*Lightweight Temporal Compression*) e DPCM (*Differential Pulse Code Modulation*).

In LTC ad esempio quello che viene fatto è approssimare una serie temporale secondo un modello lineare. Questo si ottiene approssimando più letture di un determinato dato con un singolo segmento. Essendo ogni segmento descritto da due coefficienti, la compressione si ottiene quando il numero di letture che esso copre è maggiore di 2.

Definito il contesto di azione (reti di sensori radio e algoritmi di compressione), il lavoro di questa tesi presenta uno studio della tecnica *Compressive Sensing (CS)*, scelta in quanto richiede limitate risorse computazionali ai nodi e quindi è semplicemente implementabile in modo distribuito e con bassi costi energetici. In aggiunta, i risultati in termini di ricostruzione del segnale completo al DCP sono molto buoni. Per risolvere il problema della sparsificazione sopra menzionato si è deciso di utilizzare l'analisi delle componenti principali (*PCA*), che verrà ampiamente descritta in seguito. Tutto il lavoro svolto è inserito in un framework MATLAB chiamato *SCoRe1* al quale è stato aggiunto un generatore di segnale per permettere di analizzare in dettaglio il comportamento degli algoritmi al variare delle caratteristiche statistiche del segnale quali la correlazione temporale e spaziale. Infine è stato fatto un confronto sotto il punto di vista energetico di CS con alcuni algoritmi introdotti in precedenza (*DCT*, *DSC* e *LTC*).

Il resto della tesi è così strutturato: nel capitolo 2 viene descritto come avviene la generazione del segnale utilizzato nelle simulazioni; i capitoli 3 e 4 spiegano in dettaglio come funzionano rispettivamente la *PCA* e *CS*; nel capitolo 5 viene presentato il framework *SCoRe1* ed infine nel capitolo 6 vengono mostrati i risultati ottenuti, le cui conclusioni sono esposte nel capitolo 7.

Capitolo 2

Generazione del Segnale

In questo capitolo si descrive come viene generato il segnale utilizzato nelle simulazioni.

2.1 Il segnale

Nella letteratura corrente solitamente le simulazioni di reti di sensori radio vengono eseguite su campioni di segnali reali precedentemente misurati e poi utilizzati per verificare il funzionamento dei modelli proposti. Un elemento di novità introdotto in questo lavoro è l'utilizzo di un generatore di segnali che rende disponibili insiemi di dati che rispondono a particolari caratteristiche statistiche definite a priori dall'utente. In particolare, con questo sistema è possibile imporre alcune proprietà di correlazione (spaziale e temporale) che poi vengono trasferite al segnale sintetico generato.

L'importanza di sfruttare un generatore di segnali permette di studiare in dettaglio se un dato modello statistico-matematico utilizzato per la compressione e la ricostruzione dei dati è efficace in determinate situazioni oppure tende a fallire al variare di alcune proprietà statistiche. Per di più, può essere utile per testare i limiti delle prestazioni nella raccolta dei dati e verificare l'accuratezza delle soluzioni per il processing distribuito di segnali.

La correlazione spaziale e temporale sono elementi statistici chiave per i seg-

Generazione del Segnale

nali reali e sono spesso sfruttate da molte applicazioni nel dominio delle WSN.

Nel dettaglio, la correlazione temporale cattura l'evoluzione nel tempo del segnale e, per esempio, può essere sfruttata per trovare intervalli di campionamento appropriati per una ricostruzione del segnale accurata. Durante questi intervalli i nodi sensori potrebbero andare in uno stato a basso consumo energetico e quindi risparmiare energia.

La correlazione spaziale invece può essere sfruttata nella fase di posizionamento per reti di monitoraggio ambientale nel tentativo di ottenere densità di dispiegamento appropriate.

Inoltre queste proprietà possono essere direttamente utilizzate nella progettazione di protocolli di reti e nelle tecniche per la compressione che fanno uso delle statistiche stesse del segnale. Come esempio, gli autori di [7] hanno realizzato un protocollo *Medium Access Control* (MAC) distribuito e collaborativo per le WSN che utilizza la correlazione spaziale del segnale monitorato sfruttando il fatto che un nodo sensore può agire come un rappresentante per gli altri sensori nelle vicinanze.

Infine, recenti tecniche per l'aggregazione e la compressione dei dati distribuita nelle WSN usano strumenti teorici come Compressive Sensing [8][9] da cui prende spunto questo lavoro.

2.1.1 Generazione segnale sintetico

Per le simulazioni è stato utilizzato il generatore di segnale descritto in [10]. L'idea di fondo consiste nello studiare segnali reali per ricavarne le statistiche rilevanti e successivamente sfruttarle come parametri di ingresso del generatore per ottenere segnali con caratteristiche simili. Quello che si ottiene è un modello dinamico, che evolve in istanti discreti e quindi permette la generazione di $z(\mathbf{p}, i\Delta T)$ per $i = 0, 1, 2, 3, \dots$ dove $z(\mathbf{p}, t)$ è il segnale colorato che si vuole ottenere. Questo modello cerca di conservare il più possibile le caratteristiche di correlazione che vengono imposte.

Nel lavoro di questa tesi i valori dei parametri che descrivono le correlazioni

spazio-temporali non vengono ricavati da altri segnali reali, ma vengono utilizzati come parametri liberi per studiare le prestazioni dei metodi di compressione che verranno descritti in seguito.

Nel dettaglio, il segnale viene così generato:

1. All'istante $t_0 = 0$ viene generato un campo casuale *i.i.d.* Gaussiano $w(\mathbf{p}, t_0) : \mathcal{D} \times \mathcal{T} \rightarrow \mathbb{R}$ dove ogni punto è una variabile casuale Gaussiana con media zero e varianza unitaria, $\mathcal{N}(0, 1)$. $w(\mathbf{p}, t_0)$ per costruzione è un processo stazionario.
2. $z(\mathbf{p}, t_0)$ è ottenuto colorando $w(\mathbf{p}, t_0)$ attraverso un filtro 2D. In particolare prima viene ottenuto $W(\omega, t_0)$ da $w(\mathbf{p}, t_0)$ attraverso la trasformata di Fourier 2D. Successivamente, dato un punto di riferimento $\mathbf{p}_0 = (0, 0)$, per ogni punto $\mathbf{p} \in \mathcal{D}$ viene imposta la correlazione spaziale $\rho_S(|\mathbf{p} - \mathbf{p}_0|)$ e viene calcolata la sua trasformata di Fourier, $R_S(\omega) = F[\rho_S(|\mathbf{p} - \mathbf{p}_0|)]$.
3. Dopodiché viene moltiplicata $W(\omega, t_0)$ per $R_S(\omega)^{1/2}$ e viene calcolata la trasformata di Fourier inversa del risultato così ottenuto per ricavare il campo casuale colorato voluto $z(\mathbf{p}, t_0)$, che mantiene la proprietà di stazionarietà ed è ancora Gaussiano. I dettagli che dimostrano quest'ultima osservazione si possono trovare in [11]. Il campo z così ottenuto è correlato nello spazio e la correlazione spaziale può essere controllata attraverso una valida funzione di correlazione $\rho_S(|\mathbf{p} - \mathbf{p}_0|)$.
4. Per imporre la correlazione temporale è stato adottato un filtro autoregressivo:

$$w(\mathbf{p}, t_0 + \Delta T) = \rho_T(\Delta T)w(\mathbf{p}, t_0) + \sqrt{1 - \rho_T(\Delta T)^2}\epsilon(\mathbf{p}, t_0 + \Delta T) \quad (2.1)$$

dove $\epsilon(\mathbf{p}, t_0 + \Delta T)$ è rumore bianco Gaussiano *i.i.d.*, $\mathcal{N}(0, 1)$. $\rho_T(\Delta T)$ è un coefficiente di correlazione temporale che viene imposto dall'utente nel modello. La procedura 1-4 viene ripetuta per ogni istante temporale per calcolare $w(\mathbf{p}, t_0 + i\Delta T)$ da $w(\mathbf{p}, t_0 + (i - 1)\Delta T)$ per

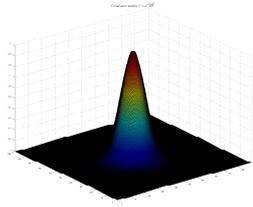


Figura 2.1: Cov. matrix

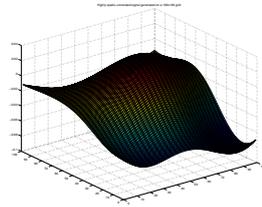


Figura 2.2: Correlato

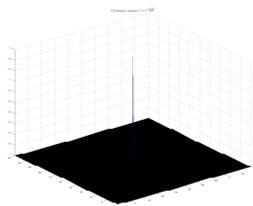


Figura 2.3: Cov. matrix

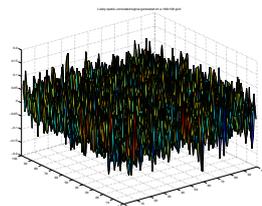


Figura 2.4: Scorrelato

$i = 0, 1, 2, 3, \dots$. Da notare che $w(\mathbf{p}, i\Delta T)$ è ancora *i.i.d.* nel dominio spaziale ed è correlato temporalmente con il coefficiente di correlazione $\rho(\Delta T)$. Come è dimostrato in [10] $w(\mathbf{p}, t)$ è di nuovo un processo casuale stazionario e Gaussiano che mantiene la stessa media e varianza del processo originale $w(\mathbf{p}, t_0)$.

Come viene dimostrato in [10] il modello auto-regressivo che viene imposto al segnale spazialmente correlato, mantiene le proprietà statistiche del campo casuale $w(\mathbf{p}, t)$ ovvero il filtraggio auto-regressivo è stabile e il segnale che genera è tempo-stazionario. Questo viene dimostrato - usando i risultati in [9] - facendo vedere che sia la media che la varianza di $w(\mathbf{p}, t)$ sono conservate. Un esempio dei segnali generati si può vedere nelle figure 2.2 e 2.4 dove viene mostrata la differenza tra due segnali che prendono come input due diverse matrici di correlazione (figure 2.1 e 2.3) che vanno ad impattare sulla correlazione spaziale.

Capitolo 3

Principal Component Analysis

La Principal Component Analysis è uno strumento della moderna analisi dei dati che viene utilizzato in molti campi delle scienze, anche diversi, come le neuroscienze e la computer grafica. Grazie alla sua semplicità è uno dei principali metodi per estrarre informazioni rilevanti da insiemi di dati variegati. Con uno piccolo sforzo, la *PCA* fornisce le linee guida per ridurre un insieme di dati complesso in uno di dimensione più piccola in cui sono evidenziate le - a volte nascoste - strutture semplificate che spesso sottendono ad esso.

L'obiettivo della *PCA* è quello di identificare la base più significativa per ri-esprimere l'insieme dei dati raccolti. La speranza è che questa nuova base filtri il rumore e riveli la struttura nascosta, permettendo di capire quali elementi del segnale sono importanti, ridondanti o rumore.

3.1 Principio di funzionamento della PCA

Stando all'algebra lineare, se il segnale viene espresso tramite una matrice $n \times m$, ogni vettore giace in uno spazio n -dimensionale generato da una qualche base ortonormale. Solitamente, la base che viene utilizzata è la matrice identità $n \times n$ dove ogni riga è un vettore canonico \mathbf{b}_i con n componenti. Tutti i dati vengono registrati in questa base e quindi possono essere espressi come una combinazione lineare dei $\{\mathbf{b}_i\}$.

Principal Component Analysis

Fatte queste assunzioni, ci si potrebbe chiedere se c'è un'altra base, combinazione lineare della base originale, che meglio rappresenti il nostro insieme di dati. Il fatto di considerare solo combinazioni lineari semplifica in modo significativo il problema restringendo l'insieme delle basi potenziali. Con questa ulteriore assunzione, la *PCA* si limita a ri-esprimere i dati come una combinazione lineare dei vettori della nuova base.

Sia \mathbf{X} l'insieme originale dei dati, dove ogni colonna è un singolo campione (o fotografia) dei nostri dati in un certo istante temporale. Sia \mathbf{Y} un'altra matrice $n \times m$ ricavata da una trasformazione lineare \mathbf{P} . \mathbf{X} è l'insieme originale dei dati e \mathbf{Y} è una nuova loro rappresentazione.

$$\mathbf{PX} = \mathbf{Y} \tag{3.1}$$

Si definiscono ora le seguenti quantità:

- \mathbf{p}_i sono le righe di \mathbf{P}
- \mathbf{x}_i sono le colonne di \mathbf{X}
- \mathbf{y}_i sono le colonne di \mathbf{Y}

L'equazione (3.1) rappresenta un cambiamento di base e quindi può avere molte interpretazioni:

- \mathbf{P} è una matrice che trasforma \mathbf{X} in \mathbf{Y}
- geometricamente, \mathbf{P} è una rotazione e uno stiramento che, di nuovo, trasforma \mathbf{X} in \mathbf{Y}
- le righe di \mathbf{P} , $\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$, sono un insieme di nuovi vettori base per esprimere le colonne di \mathbf{X}

L'ultima espressione potrebbe non essere ovvia, ma può essere vista scrivendo esplicitamente il prodotto di \mathbf{PX} .

$$\mathbf{PX} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{bmatrix} [\mathbf{x}_1 \dots \mathbf{x}_m]$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{p}_1 \cdot \mathbf{x}_1 & \dots & \mathbf{p}_1 \cdot \mathbf{x}_m \\ \vdots & \ddots & \vdots \\ \mathbf{p}_n \cdot \mathbf{x}_1 & \dots & \mathbf{p}_n \cdot \mathbf{x}_m \end{bmatrix}$$

Si può notare come ogni coefficiente di \mathbf{y}_i è il prodotto di \mathbf{x}_i con la corrispondente riga in \mathbf{P} .

$$\mathbf{y}_i = \begin{bmatrix} \mathbf{p}_1 \cdot \mathbf{x}_i \\ \vdots \\ \mathbf{p}_n \cdot \mathbf{x}_i \end{bmatrix}$$

In altre parole, il j -esimo coefficiente di \mathbf{y}_i è una proiezione sulla j -esima riga di \mathbf{P} . Questa infatti è la forma di un'equazione dove \mathbf{y}_i è una proiezione sulla base di $\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$. Quindi, le righe di \mathbf{P} sono un nuovo insieme di vettori base per rappresentare le colonne di \mathbf{X} .

3.2 Studio della varianza nella PCA

Assumendo le condizioni di linearità sopra enunciate, il problema si riduce a trovare il cambio di base appropriato. I vettori riga $\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ in questa trasformazione diventeranno le *componenti principali* di \mathbf{X} .

A questo punto ci si potrebbe chiedere qual'è il modo migliore per ri-esprimere \mathbf{X} e quale potrebbe essere una buona scelta per la base \mathbf{P} .

Per rispondere a queste domande, serve capire che proprietà si vuole che \mathbf{Y} mostri.

È evidente che sono necessarie altre assunzioni oltre la richiesta di linearità per arrivare ad un risultato ragionevole.

La questione principale però è: cosa significa “meglio ri-esprimere” i dati? Si cercherà ora di dare una risposta intuitiva a questa domanda.

In qualsiasi insieme di dati raccolti il rumore deve essere trascurabile rispetto alla misura effettuata, altrimenti, indipendentemente dalla tecnica di analisi utilizzata, nessuna informazione utile può essere dedotta. Non c'è una scala assoluta per il rumore, ma piuttosto tutto il rumore è quantificato relativamente all'intensità del segnale percepito. Un indice comune che esprime

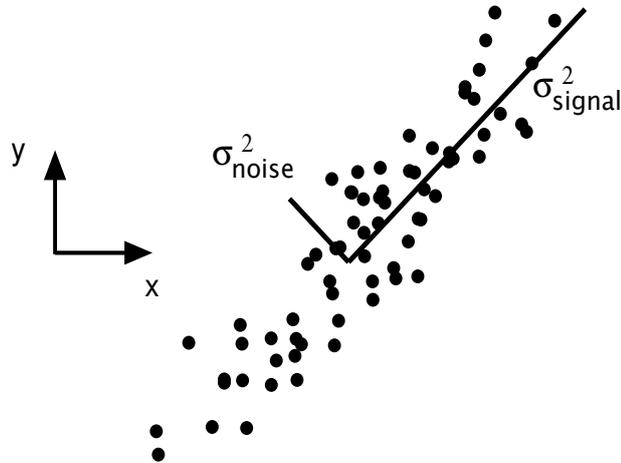


Figura 3.1: Posizioni dati (x,y) per la camera A

questo rapporto è il *signal-to-noise ratio* meglio noto come *SNR* o rapporto di varianze σ^2

$$SNR = \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2}$$

Un alto *SNR* ($\gg 1$) indica una misurazione molto precisa, mentre un basso *SNR* indica un dato molto rumoroso.

In [14] viene trattato un esempio dove 3 telecamere registrano ad una frequenza di 120Hz la posizione di una massa che oscilla attaccata ad una molla. L'output di una telecamera dopo un certo tempo T è mostrato nella figura 3.1. Se si tiene presente che la molla viaggia in linea retta, ogni telecamera dovrebbe registrare una serie di punti totalmente allineati. A questo punto, ogni deviazione dal movimento rettilineo è considerato rumore. Le varianze dovute al segnale e al rumore sono indicate da ciascuna linea nel grafico. Il rapporto delle due lunghezze rappresenta quanto schiacciata è la nuvola di punti: da una linea ($SNR \gg 1$) ad un cerchio ($SNR = 1$) o peggio.

In questo contesto si assume che le direzioni con la più ampia varianza nello spazio delle misure contengano le dinamiche di interesse. Nell'esempio in figura 3.1, la direzione con la più larga varianza non è $\hat{x}_A = (1, 0)$ né $\hat{y}_A = (0, 1)$, ma la direzione lungo l'asse maggiore della nuvola di punti. Quindi, per le considerazioni prima fatte, le dinamiche di interesse coinvolgono le direzioni

con la più ampia varianza e presumibilmente il più alto SNR .

Questa assunzione suggerisce che la base che si sta cercando non è la base canonica, perché le direzioni (x_A, y_A) non corrispondono alle direzioni dove la varianza è maggiore. Massimizzare la varianza (e quindi l' SNR) significa trovare la corretta rotazione della base canonica. Questa intuizione si traduce nel trovare nella figura 3.1 la direzione indicata dalla linea σ_{signal}^2 . Nel caso 2-dimensionale la direzione della varianza più larga corrisponde alla linea che meglio approssima la nuvola di dati. Quindi, ruotando la base canonica in modo che giaccia parallela a tale linea, porterà a rivelare la direzione della molla nel caso 2-D.

Prima di generalizzare queste considerazioni per un numero arbitrario di dimensioni, consideriamo la *ridondanza* che nell'esempio trattato in [14] viene evidenziata dal fatto che diversi sensori registrano la stessa informazione dinamica.

3.3 Matrice di covarianza

Nel caso di 2 variabili, è facile determinare casi ridondanti trovando la pendenza della linea che meglio approssima i dati e giudicando la qualità dell'approssimazione. Come è possibile quantificare e generalizzare queste nozioni per dimensioni arbitrarie? Lo si veda attraverso un esempio:

si considerino due insiemi di misure con media nulla

$$A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_n\} \quad (3.2)$$

dove il pedice denota il numero del campione. Le varianze di A e B sono definite individualmente come:

$$\sigma_A^2 = \frac{1}{n} \sum_i a_i^2, \sigma_B^2 = \frac{1}{n} \sum_i b_i^2$$

La *covarianza* tra A e B è una diretta generalizzazione:

$$\text{covarianza di A e B} \equiv \sigma_{AB}^2 = \frac{1}{n} \sum_i a_i b_i$$

Principal Component Analysis

La covarianza misura il grado della relazione lineare tra due variabili. Un valore positivo alto indica dati molto correlati, un alto valore negativo denota dati negativamente correlati. La magnitudine assoluta della covarianza misura il grado di ridondanza.

Alcune proprietà della covarianza:

1. σ_{AB} è 0 se e solo se A e B sono scorrelati
2. $\sigma_{AB}^2 = \sigma_A^2$ se $A = B$

Se si trasformano A e B nei corrispondenti vettori riga

$$\mathbf{a} = \{a_1, \dots, a_n\}$$

$$\mathbf{b} = \{b_1, \dots, b_n\}$$

si può esprimere la covarianza come un prodotto matriciale:

$$\sigma_{\mathbf{AB}}^2 \equiv \frac{1}{n} \mathbf{ab}^T \quad (3.3)$$

Infine si può generalizzare da due vettori ad un numero arbitrario di essi. Rinominando i vettori riga \mathbf{a} e \mathbf{b} come \mathbf{x}_1 e \mathbf{x}_2 , rispettivamente e considerando ulteriori vettori riga così indicizzati $\mathbf{x}_3, \dots, \mathbf{x}_n$ si definisce una nuova matrice \mathbf{X} $n \times m$.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}$$

Una possibile interpretazione di \mathbf{X} è la seguente. Ogni *riga* di \mathbf{X} corrisponde a tutte le misure nel tempo di un determinato nodo. Ogni *colonna* di \mathbf{X} corrisponde all'insieme di misure in un dato istante. Allora si può definire la *matrice di covarianza* $\Sigma_{\mathbf{X}}$ come:

$$\Sigma_{\mathbf{X}} \equiv \frac{1}{n} \mathbf{X}\mathbf{X}^T \quad (3.4)$$

Se si considera la matrice in (3.4), l' ij -esimo elemento di $\Sigma_{\mathbf{X}}$ è il prodotto riga per colonna tra il vettore dell' i -esima misurazione con il vettore della j -esima misurazione.

Alcune proprietà della matrice di covarianza $\Sigma_{\mathbf{X}}$ sono:

1. $\Sigma_{\mathbf{X}}$ è una matrice quadrata e simmetrica $n \times n$
2. I termini della diagonale di $\Sigma_{\mathbf{X}}$ rappresentano la *varianza* di una particolare misurazione
3. I termini fuori dalla diagonale di $\Sigma_{\mathbf{X}}$ rappresentano la *covarianza* tra le misurazioni

In pratica $\Sigma_{\mathbf{X}}$ cattura la covarianza tra tutte le coppie possibili di misurazioni, che riflettono il rumore e la ridondanza nelle misurazioni.

Nei termini della diagonale, per assunzione, valori alti corrispondono ad una struttura interessante.

Nei termini fuori della diagonale magnitudine alta corrisponde ad alta ridondanza.

Fingendo di avere la possibilità di manipolare $\Sigma_{\mathbf{X}}$, si potrebbe definire in modo opportuno la matrice di covarianza $\Sigma_{\mathbf{Y}}$ in modo tale da ottimizzare alcune caratteristiche interessanti di questa matrice.

3.3.1 Diagonalizzare la matrice di covarianza

Riassumendo gli ultimi due punti, si può dire che gli obiettivi sono: minimizzare la ridondanza - misurata dalla magnitudine delle covarianza - e massimizzare il segnale - misurato dalla varianza. Ci si chiede ora come dovrebbe essere la matrice di covarianza $\Sigma_{\mathbf{Y}}$ ottimizzata:

- tutti i termini fuori dalla diagonale in $\Sigma_{\mathbf{Y}}$ dovrebbero essere zero, ovvero $\Sigma_{\mathbf{Y}}$ deve essere una matrice diagonale (equivalentemente, \mathbf{Y} è decorrelata).
- ogni dimensione successiva di \mathbf{Y} dovrebbe essere ordinata secondo la varianza.

A questo punto, ci sono molti modi per diagonalizzare $\Sigma_{\mathbf{Y}}$ e si noti che la *PCA* sceglie il metodo più semplice: *PCA* assume che tutti i vettori base $\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ siano ortonormali, ovvero \mathbf{P} è un matrice ortonormale.

Principal Component Analysis

Nell'esempio proposto in [14] \mathbf{P} agisce come una rotazione generalizzata per allineare una base con l'asse di massima varianza. Nel caso di dimensioni multiple questo può essere realizzato con un semplice algoritmo:

1. Si seleziona una direzione normalizzata in uno spazio m -dimensionale lungo la quale la varianza di \mathbf{X} è massima e si salva questo valore come \mathbf{p}_1
2. Si trova un'altra direzione lungo la quale la varianza è massima. A causa della condizione di ortonormalità questa ricerca è ristretta a sole le direzioni ortogonali alle direzioni precedentemente selezionate.
3. Si salva questo vettore come \mathbf{p}_i
4. Si ripete questa procedura finché non sono stati scelti m vettori.

Ora l'insieme ordinato dei vettori risultati \mathbf{p}_i rappresenta le *componenti principali*.

Il vero beneficio di assumere l'ortonormalità della matrice è che esiste una soluzione analitica e efficiente a questo problema. Con la condizione di varianze ordinate, si ha un metodo per stabilire l'importanza delle direzioni principali. Vale a dire le varianze associate ad ogni direzione di \mathbf{p}_i quantificano quanto "principale" è la direzione ordinando ogni vettore della base \mathbf{p}_i in accordo alla corrispondente varianza.

Riassumendo, le assunzioni fatte per la *PCA* coinvolgono:

1. **Linearità:** limita il problema ad un cambio di base.
2. **Varianze ampie rispecchiano una struttura importante:** questa assunzione abbraccia anche l'idea che i dati abbiano un alto *SNR*. Quindi, le componenti principali con le varianze più larghe associate rappresentano la struttura interessante, mentre quelle con varianza più bassa rappresentano rumore. Si noti che non sempre questa assunzione è corretta.

3.4 Risolvere la PCA tramite decomposizione in autovettori

3. **Le componenti principali sono ortogonali:** questa assunzione fornisce una semplificazione intuitiva che rende la *PCA* risolvibile con tecniche di decomposizione dell'algebra lineare, come descritto nella prossima sezione.

Si è parlato di tutti gli aspetti di come derivare la *PCA*, quello che rimane ora sono le soluzioni dell'algebra lineare.

3.4 Risolvere la PCA tramite decomposizione in autovettori

La soluzione algebrica che viene proposta è basata su un'importante proprietà della decomposizione in autovettori. Di nuovo, l'insieme dei dati è \mathbf{X} , una matrice $n \times m$, dove n è il numero dei sensori e m il numero di istanti temporali. L'obiettivo è il seguente:

trovare una qualche matrice \mathbf{P} in $\mathbf{Y} = \mathbf{P}\mathbf{X}$ tale che $\Sigma_{\mathbf{Y}} \equiv \frac{1}{n}\mathbf{Y}\mathbf{Y}^T$ sia una matrice diagonale.

Le righe di \mathbf{P} sono le *componenti principali* di \mathbf{X} .

Si può riscrivere $\Sigma_{\mathbf{Y}}$ in termini di variabili non note.

$$\begin{aligned}\Sigma_{\mathbf{Y}} &= \frac{1}{n}\mathbf{Y}\mathbf{Y}^T \\ &= \frac{1}{n}(\mathbf{P}\mathbf{X})(\mathbf{P}\mathbf{X})^T \\ &= \frac{1}{n}\mathbf{P}\mathbf{X}\mathbf{X}^T\mathbf{P}^T \\ &= \mathbf{P}\left(\frac{1}{n}\mathbf{X}\mathbf{X}^T\right)\mathbf{P}^T \\ \Sigma_{\mathbf{Y}} &= \mathbf{P}\Sigma_{\mathbf{X}}\mathbf{P}^T\end{aligned}$$

Si noti che l'ultima riga rappresenta la matrice di covarianza di \mathbf{X} .

Lo scopo è riconoscere che qualunque matrice simmetrica \mathbf{A} viene diagonalizzata da una matrice ortogonale dei suoi autovalori come mostrato nell'appendice A in [14]. Per una matrice simmetrica \mathbf{A} , il *Teorema 4* fornisce

Principal Component Analysis

$\mathbf{A} = \mathbf{E}\mathbf{D}\mathbf{E}^T$, dove \mathbf{D} è una matrice diagonale e \mathbf{E} è una matrice di autovettori di \mathbf{A} disposti come colonne.

Ora, si seleziona la matrice \mathbf{P} in modo che sia una matrice dove ogni riga \mathbf{p}_i è un autovettore di $\frac{1}{n}\mathbf{X}\mathbf{X}^T$. Con questa scelta, $\mathbf{P} \equiv \mathbf{E}^T$. Con questa relazione e con il *Teorema 1* ($\mathbf{P}^{-1} = \mathbf{P}^T$) si può finire di calcolare $\Sigma_{\mathbf{Y}}$.

$$\begin{aligned}\Sigma_{\mathbf{Y}} &= \mathbf{P}\Sigma_{\mathbf{X}}\mathbf{P}^T \\ &= \mathbf{P}(\mathbf{E}^T\mathbf{D}\mathbf{E})\mathbf{P}^T \\ &= \mathbf{P}(\mathbf{P}^T\mathbf{D}\mathbf{P})\mathbf{P}^T \\ &= (\mathbf{P}\mathbf{P}^T)\mathbf{D}(\mathbf{P}\mathbf{P}^T) \\ &= (\mathbf{P}\mathbf{P}^{-1})\mathbf{D}(\mathbf{P}\mathbf{P}^{-1}) \\ \Sigma_{\mathbf{Y}} &= \mathbf{D}\end{aligned}$$

È evidente che la scelta di \mathbf{P} diagonalizza $\Sigma_{\mathbf{Y}}$. Questo era l'obiettivo della *PCA*.

Ora si possono riassumere i risultati della *PCA* nelle matrici \mathbf{P} e $\Sigma_{\mathbf{Y}}$.

- Le componenti principali di \mathbf{X} sono gli autovettori di $\Sigma_{\mathbf{X}} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- L' i -esimo elemento della diagonale di $\Sigma_{\mathbf{Y}}$ rappresenta la varianza di \mathbf{X} lungo \mathbf{p}_i .

Nella pratica dunque, il calcolo dalla *PCA* di un insieme di dati \mathbf{X} si riduce al calcolo degli autovettori di $\Sigma_{\mathbf{X}} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$

Capitolo 4

Compressive Sensing

Gli approcci convenzionali per campionare segnali o immagini seguono il noto *Teorema di Shannon* che dice che la frequenza di campionamento deve essere almeno doppia rispetto alla massima frequenza presente nel segnale (la così detta frequenza di *Nyquist*). Infatti, questo principio sottostà a tutti i protocolli di campionamento dei segnali in campi quali l'acquisizione di segnali audio-video, strumenti per elaborazioni immagini mediche, ricevitori radio e così via.

Nel campo della conversione dati, per esempio, i convertitori standard *analog-to-digital* implementano la classica rappresentazione quantizzata di Shannon: il segnale è campionato uniformemente alla frequenza di Nyquist (o superiore).

La teoria su cui si fonda Compressive Sensing sostiene invece che si possono ricostruire alcuni segnali da un numero relativamente piccolo di campioni o misurazioni rispetto a quelli usati dai metodi tradizionali. Per rendere possibile questo fatto, CS si basa su due principi: *sparsità*, che riguarda il segnale di interesse, e *incoerenza*, relativa alla modalità di campionamento.

La sparsità esprime l'idea che il tasso di informazione di un segnale continuo nel tempo potrebbe essere molto minore rispetto alla sua banda suggerita, o che un segnale discreto nel tempo dipenda da un numero di gradi di libertà che è molto più piccolo rispetto alla sua lunghezza (finita). Più precisamente

CS sfrutta il fatto che molti segnali “naturalmente” sono sparsi o comprimibili, nel senso che hanno una rappresentazione concisa se espressi in una base adeguata Ψ .

L’incoerenza invece estende la dualità tra il tempo e la frequenza e esprime l’idea che gli oggetti che hanno una rappresentazione sparsa in una base Ψ devono essere diffusi nel dominio nel quale sono stati acquisiti, come un picco nel dominio temporale è diffuso nel dominio delle frequenze. In altre parole, l’incoerenza dice che, a differenza del segnale di interesse, le forme d’onda del campionamento hanno una rappresentazione estremamente densa in Ψ .

4.1 Formulazione matematica

CS è una tecnica che viene utilizzata per ricostruire un dato segnale n -dimensionale attraverso la ricezione di un numero di campioni l che idealmente è molto più piccolo di n .

Come nel caso della *PCA*, si considerino segnali rappresentabili attraverso vettori n -dimensionali $\mathbf{x}^{(k)} \in \mathbb{R}^n$, contenenti i valori delle letture di una rete di sensori radio con n nodi dove k indica l’istante temporale nel quale il segnale è campionato. Inoltre si assuma l’esistenza di una matrice di trasformazione invertibile Ψ di dimensioni $n \times n$ tale che

$$\mathbf{x}^{(k)} = \Psi \mathbf{s}^{(k)} \tag{4.1}$$

dove $\mathbf{s}^{(k)} \in \mathbb{R}^n$ è un vettore n -dimensionale m -sparso. $\mathbf{s}^{(k)}$ si dice m -sparso quando ha solo m componenti significative, mentre le altre $n - m$ sono trascurabili rispetto all’energia media per componente, definita¹ come $E_s^{(k)} = \frac{1}{n} \sqrt{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle}$. Assumendo che Ψ sia nota, $\mathbf{s}^{(k)}$ può essere ricavato da $\mathbf{x}^{(k)}$ invertendo l’equazione sopra, ovvero $\mathbf{s}^{(k)} = \Psi^{-1} \mathbf{x}^{(k)}$.

Inoltre, $\mathbf{s}^{(k)}$ può essere ottenuto da un numero l di proiezioni casuali di $\mathbf{x}^{(k)}$,

¹Per qualsiasi coppia di vettori colonna \mathbf{a} e \mathbf{b} della stessa lunghezza, il prodotto $\langle \mathbf{a}, \mathbf{b} \rangle$ è definito come $\mathbf{a}^T \mathbf{b}$

4.1 Formulazione matematica

ossia $\mathbf{y}^{(k)} \in \mathbb{R}^l$, con $m \leq l < n$, in accordo con la seguente equazione:

$$\mathbf{y}^{(k)} = \Phi \mathbf{x}^{(k)} \quad (4.2)$$

dove Φ è chiamata *routing matrix* poiché cattura il modo in cui i dati misurati sono raccolti e trasmessi al *DCP*.

Per l'applicazione specifica di questo lavoro, Φ è una matrice $l \times n$ di tutti zero eccetto per un singolo uno in ogni riga e al più un singolo uno per ogni colonna (ovvero $\mathbf{y}^{(k)}$ è una versione campionata di $\mathbf{x}^{(k)}$). Scegliere Φ in questo modo ha due vantaggi: il primo sta nel fatto che la matrice è ortonormale e il secondo che questo tipo di matrice di routing può essere ottenuta attraverso tabelle di instradamento reali.

Ora, usando le equazioni (4.1) e (4.2) si può scrivere:

$$\mathbf{y}^{(k)} = \Phi \mathbf{x}^{(k)} = \Phi \Psi \mathbf{s}^{(k)} \triangleq \tilde{\Phi} \mathbf{s}^{(k)} \quad (4.3)$$

Questo sistema è *mal posto* poiché il numero di equazioni l è più piccolo rispetto al numero di variabili n . Potrebbe essere anche *mal condizionato* nel senso che una piccola variazione del segnale di output $\mathbf{y}^{(k)}$ può produrre una grande variazione del segnale di input. Tuttavia, se $\mathbf{s}^{(k)}$ è sparso e la matrice prodotto $\Phi \Psi$ soddisfa la condizione *RIP*² (*Restricted Isometry Property*), è dimostrato che l'equazione (4.3) può essere invertita con alta probabilità attraverso l'uso di specifiche tecniche di ottimizzazione come mostrato in [15]. Queste permettono di ricavare $\mathbf{s}^{(k)}$ da cui poi si può ricavare il segnale originale $\mathbf{x}^{(k)}$ tramite l'equazione (4.1).

4.1.1 Restricted Isometry Property

Nel tentativo di trovare qualche parametro che descriva quanto deve essere “buona” la matrice delle misure per gli scopi di CS, Candés e Tao hanno introdotto questo concetto di *restricted isometry property* che viene definito come:

²condizione sufficiente

Compressive Sensing

per tutti gli \mathbf{x} tali che $\|\mathbf{x}\|_0 \leq K$, \mathbf{A} soddisfa la *RIP* con parametro δ_K , se δ_K è il più piccolo valore che soddisfa:³

$$(1 - \delta_K)\|\mathbf{x}\|_2^2 \leq \|\mathbf{Ax}\|_2^2 \leq (1 + \delta_K)\|\mathbf{x}\|_2^2$$

La matrice delle misure \mathbf{A} può essere vista come una trasformazione del segnale dallo spazio del segnale allo spazio delle misure, dove lo spazio delle misure è più piccolo dello spazio del segnale. Per la variabile \mathbf{x} quindi, la *RIP* è una caratterizzazione di come la sua norma Euclidea cambia rispetto alla trasformazione descritta da \mathbf{A} . D'altro canto, considerando ad esempio due variabili \mathbf{x}_1 e \mathbf{x}_2 , la *RIP* può anche essere interpretata come un indice di quanto la distanza tra \mathbf{x}_1 e \mathbf{x}_2 cambi dopo aver applicato la trasformazione \mathbf{A} .

4.2 Soluzioni sparse di sistemi mal posti

L'idea è la seguente: data una soluzione \mathbf{s}_p di (4.3) tale che $\tilde{\Phi}\mathbf{s}_p = \mathbf{y}$ e dato il *kernel* della matrice $\tilde{\Phi}$, $\mathcal{N}(\tilde{\Phi})$ di dimensione $n - m$, qualsiasi vettore $\mathbf{s}' = \mathbf{s}_p + \mathbf{r}$, dove $\mathbf{r} \in \mathcal{N}(\tilde{\Phi})$ è anch'esso una soluzione di (4.3).

Tuttavia, in [9] si dimostra che:

1. se la matrice $\tilde{\Phi}$ ha certe proprietà (*RIP*)
2. se \mathbf{s} è p -sparso con p più piccolo di una certa soglia

allora \mathbf{s} è la soluzione ammissibile più sparsa di (4.3). La soluzione che si trova in questo modo, che viene chiamata \mathbf{s}^* , è uguale all' \mathbf{s} originale se le condizioni 1) e 2) sono verificate. Altrimenti, ci sarà un errore di ricostruzione che diminuisce all'aumentare di m e chiaramente quando $m = n$ e $\tilde{\Phi}$ ha rango pieno, l'unica soluzione del sistema è \mathbf{s} e può essere ottenuta tramite una classica inversione di matrice.

³ $\|\cdot\|_0$ viene definita con un abuso di notazione "Norma 0" e rappresenta il numero di componenti nulle del vettore

Capitolo 5

Framework SCoRe1

Ora che sono stati presentati gli strumenti matematici utilizzati per ricostruire il segnale, si passa alla descrizione del framework impiegato per l'analisi dei dati, i cui risultati saranno poi presentati nel capitolo 6.

5.1 Descrizione

L'intero lavoro di questa tesi si appoggia su *SCoRe1* [16]: *Sensing, Compression and Recovery through ON-line Estimation for WSNs*: un framework scritto in MATLAB in grado di simulare il comportamento delle reti di sensori radio e i relativi meccanismi di ricostruzione del segnale con tecniche quali *CS – PCA* e derivanti da un'analisi Bayesiana del contesto. Inoltre è capace di auto-adattarsi ai cambiamenti delle statistiche del segnale grazie ad un sistema di retroazione che è in grado di stimare, in tempo reale, l'errore di ricostruzione.

Nella WSN realizzata da *SCoRe1* per questo lavoro, vengono simulati 50 sensori posizionati su un'ipotetica griglia di dimensioni 100×100 . Il posizionamento di questi sensori avviene in modo aleatorio utilizzando due variabili casuali uniformi $\mathcal{U}(1, 100)$ *i.i.d.* per la scelta delle coordinate x e y di ogni nodo. Il punto di raccolta dati (*DCP*) invece è posizionato al centro di tale griglia alle coordinate $(x = 50, y = 50)$.

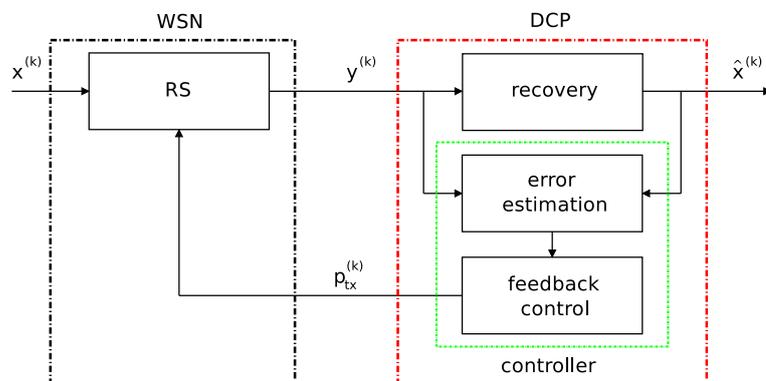


Figura 5.1: Schema a blocchi del framework SCoRe1

Il *DCP* si assume essere un calcolatore con elevate prestazioni computazionali, alimentato autonomamente da una fonte di energia esterna sempre disponibile e i cui costi energetici non sono di interesse per questa analisi. Le sue funzioni sono quelle di ricevere un vettore di dati in ingresso $\mathbf{y}^{(k)}$ e restituire il corrispondente segnale stimato $\hat{\mathbf{x}}^{(k)}$.

Oltre al blocco per fornire la ricostruzione dei dati, il *DCP* gestisce anche un *training set* $\hat{\mathcal{T}}_K$ che è utilizzato per dedurre la struttura statistica del segnale necessaria poi all'algoritmo di ricostruzione, come sarà chiaro più avanti.

Il training set è formato da tutti (o una parte) i segnali ricostruiti negli istanti precedenti all'istante di interesse. All'avvio dell'applicazione, non essendo chiaramente possibile ricavare alcun training set precedente all'istante 0, tutti i nodi trasmettono il proprio valore misurato per un numero di istanti q fissato.

In questo lavoro è stato scelto di usare $|\hat{\mathcal{T}}_K| = 2$. Tale scelta viene motivata da un precedente lavoro [16] dove viene dimostrato attraverso un'analisi mirata che $|\hat{\mathcal{T}}_K| = 2$ (ovvero $\hat{\mathcal{T}}_K = \{\mathbf{x}^{(k-2)}, \mathbf{x}^{(k-1)}\}$) conviene in termini di costi energetici. In generale quello che si può dire relativamente al training set è che meno il segnale è correlato temporalmente, più $|\hat{\mathcal{T}}_K|$ grande aiuta a ridurre l'errore di stima del segnale. Viceversa, per un segnale molto correlato temporalmente, training set di piccole dimensioni aiutano ad approssimare meglio il segnale che viene ricostruito.

5.2 Utilizzo congiunto di CS e PCA

Relativamente al campionamento, viene utilizzata una tecnica di *Random Sampling*: una strategia totalmente distribuita nella quale ad ogni istante k ogni nodo o trasmette i propri dati misurati al *DCP* oppure rimane silente. Tale scelta rappresenta una soluzione di raccolta dati semplice e generale, facile da implementare e che ha un basso sovraccarico relativo alla comunicazione per la sincronizzazione dei nodi che devono trasmettere.

In dettaglio, ad ogni istante k ogni sensore decide, con probabilità $p_{tx}^{(k)}$, se trasmettere o meno i suoi dati al *DCP*. Questa decisione è presa indipendentemente dal passato e dal comportamento del resto degli altri nodi. In questa implementazione si è deciso di mantenere $p_{tx}^{(k)}$ fissata e costante nel tempo. Nulla però vieta di variare $p_{tx}^{(k)}$ in funzione dell'errore di ricostruzione tramite un meccanismo di *feedback*, nel qual caso il valore aggiornato ad ogni istante temporale viene inviato in broadcast dal *DCP* come descritto in [16]. C'è da dire che quest'ultimo approccio non è stato preso in considerazione nell'implementazione corrente perché non ritenuto necessario per i fini dello studio realizzato.

In termini pratici questo si traduce nel far generare ad ogni sensore in ogni istante un numero arbitrario da una variabile causale uniforme $\mathcal{U}(0, 1)$ e farlo confrontare con la probabilità di trasmissione impostata: se il numero è maggiore, il nodo trasmette, se è minore, il nodo si porta nello stato di *stand-by*.

L'errore considerato nelle simulazioni è l'errore assoluto calcolato come la media del valore assoluto della differenza tra il segnale ricostruito e il segnale reale: $\epsilon^{(k)} = E[\|\hat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|_1]$.

5.2 Utilizzo congiunto di CS e PCA

Nei capitoli 3 e 4 sono stati introdotti separatamente Compressive Sensing e l'analisi delle componenti principali. Ora si illustra come combinando questi due approcci è possibile andare a ricostruire in modo efficace alcune classi di segnali.

Si è visto che la *PCA* è un metodo per rappresentare un generico segnale n -dimensionale attraverso la miglior approssimazione a m -termini, dove $n > m$ mentre *CS* è una tecnica per ricostruire un segnale n -dimensionale attraverso la ricezione di un piccolo numero di campioni l , con $l < n$. In questa sezione si descrive un sistema che ricostruisce ad ogni istante k un segnale $\mathbf{x}^{(k)}$ applicando insieme *PCA* e *CS*.

La prima assunzione è che il segnale sia correlato sia nello spazio che nel tempo, ma che in generale non sia stazionario. Questo significa che le statistiche da utilizzare nella soluzione (ovvero la media del segnale e la sua matrice di covarianza) devono essere ricavate *runtime* e potrebbero non essere valide per tutto l'intervallo temporale entro cui si vuole ricostruire il segnale.

Usando la *PCA*, dall'equazione (4.1) si può mappare in ogni istante k il segnale $\mathbf{x}^{(k)}$ in un vettore sparso $\mathbf{s}^{(k)}$. La matrice \mathbf{U} e la media $\bar{\mathbf{x}} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}^{(k)}$ possono essere pensate come calcolate iterativamente dal training set $\mathcal{X}^{(k)}$, ad ogni istante k .

Come descritto nel capitolo 3, viene calcolata la matrice di covarianza in questo modo:

$$\hat{\Sigma} = \frac{1}{K} \sum_{k=1}^K (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \quad (5.1)$$

Data l'equazione (5.1), si consideri la matrice \mathbf{U} le cui colonne sono gli autovettori di $\hat{\Sigma}$, ordinati secondo l'ordine decrescente dei rispettivi autovalori. In questo modo è possibile proiettare un vettore di misurazioni $\mathbf{x}^{(k)}$ su uno spazio vettoriale coperto dalle colonne di \mathbf{U} . Da qui si può definire $\mathbf{s}^{(k)} \triangleq \mathbf{U}^T(\mathbf{x}^{(k)} - \bar{\mathbf{x}})$.

Ad ogni istante k la matrice \mathbf{U} viene indicata come $\mathbf{U}^{(k)}$ e fa riferimento alla media temporale e alla varianza di $\mathcal{X}^{(k)}$ come $\bar{\mathbf{x}}^{(k)}$ e $\hat{\Sigma}^{(k)}$ rispettivamente.

Quindi si può scrivere:

$$\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)} = \mathbf{U}^{(k)} \mathbf{s}^{(k)} \quad (5.2)$$

Ora, usando l'equazione (4.2) e l'equazione (5.2), possiamo scrivere:

$$\mathbf{y}^{(k)} - \Phi^{(k)} \bar{\mathbf{x}}^{(k)} = \Phi^{(k)} (\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}) = \Phi^{(k)} \mathbf{U}^{(k)} \mathbf{s}^{(k)} \quad (5.3)$$

dove con la variabile $\Phi^{(k)}$ si esplicita che anche la matrice di instradamento Φ può cambiare nel tempo. Il segnale originale $\mathbf{x}^{(k)}$ è approssimato come segue:

1. si trova una buona stima¹ di $\mathbf{s}^{(k)}$, $\hat{\mathbf{s}}^{(k)}$
2. si esegue il seguente calcolo: $\hat{\mathbf{x}}^{(k)} = \bar{\mathbf{x}}^{(k)} + \mathbf{U}^{(k)}\hat{\mathbf{s}}^{(k)}$

In SCoRe1 la stima per la soluzione più sparsa del sistema sottodeterminato viene effettuata tramite l'algoritmo *SL0*[17]. Al contrario dei metodi classici che tipicamente cercano di risolvere questo problema minimizzando la norma- l^1 usando tecniche di programmazione lineare (LP), *SL0* cerca direttamente di minimizzare la norma- l^0 . È dimostrato sperimentalmente che *SL0* è due o tre ordini di grandezza più veloce rispetto ai solver LP più recenti, fornendo la stessa - se non migliore - accuratezza.

5.3 Approccio Bayesiano

Rivisitando il problema da risolvere, si assuma che il *DCP* sia posizionato al centro di una *WSN* con n nodi e si voglia determinare ad ogni istante k il valore di tutti gli n sensori raccogliendo però al *DCP* solo una piccola frazione di essi. A questo fine si può sfruttare l'approccio combinato di *PCA* e *CS* come visto nella sezione precedente. L'equazione (5.2) mostra che il framework utilizzato non dipende dalla particolare topologia considerata, l'unico requisito è che i nodi siano ordinati (ad esempio secondo il loro ID). Tale applicazione di monitoraggio può essere vista ad ogni istante k come un problema di interpolazione: da un vettore di campioni m -dimensionale $\mathbf{y}^{(k)} = \Phi\mathbf{x}^{(k)} \in \mathbb{R}^m$ si è interessati a ricostruire, attraverso un'interpolazione, il segnale $\mathbf{x}^{(k)} \in \mathbb{R}^n$. Solitamente questo problema può essere risolto attraverso un'interpolazione lineare, dove la funziona di interpolazione in questo caso ha la forma:

$$\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)} = \sum_{i=1}^n s_i^{(k)} \mathbf{u}_i^{(k)} \quad (5.4)$$

¹Ad esempio usando le tecniche in [9]

Un approccio *Bayesiano* stimerebbe il valore più probabile di $\mathbf{s}^{(k)} = (s_1^{(k)}, \dots, s_n^{(k)})^T$ massimizzando una *pdf* per il *posterior* di forma $p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)}, \mathbf{U}^{(k)}, \mathcal{M})$, dove \mathcal{M} è un modello possibile per il vettore $\mathbf{s}^{(k)}$. Due alternative sono considerare \mathbf{s} come un processo multivariato Gaussiano oppure un processo vettoriale Laplaciano con componenti *i.i.d.*

Quando si assegna a $\mathbf{s}^{(k)}$ un *prior* Gaussiano si può risolvere il problema via *Ordinary Least Square*.

Partendo dall'equazione:

$$\hat{\mathbf{s}}^{(k)} = (\mathbf{\Phi}^{(k)}\mathbf{U}^{(k)})^\dagger(\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)}\bar{\mathbf{x}}^{(k)}) \quad (5.5)$$

si ha che la dimensione di $\mathbf{y}^{(k)}$ è minore della dimensione di $\mathbf{s}^{(k)}$ ovvero $l < n$. Per cui l'equazione sopra è la soluzione di un sistema mal posto, che teoricamente ammette un numero infinito di soluzioni. Tuttavia, un prior Gaussiano multivariato $\mathbf{s}^{(k)}$ con media zero e componenti indipendenti, ovvero $p(\mathbf{s}^{(k)}) \sim \mathcal{N}(0, \mathbf{\Sigma}_s)$ dove $\mathbf{\Sigma}_s$ è una matrice diagonale, rende possibile scegliere, tra tutte le soluzioni possibili, quella stimata come:

$$\begin{aligned} \hat{\mathbf{s}}^{(k)} &= \arg \max_{\mathbf{s}^{(k)}} p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)}) \\ &= \arg \max_{\mathbf{s}^{(k)}} p(\mathbf{y}^{(k)}|\mathbf{s}^{(k)})p(\mathbf{s}^{(k)}) \\ &= \arg \max_{\mathbf{s}^{(k)}} \delta(\mathbf{y}^{(k)}, \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}) \frac{1}{(2\pi)^{\frac{l}{2}} \det(\mathbf{\Sigma}_s)^{\frac{l}{2}}} \exp -\frac{\|\mathbf{\Sigma}_s\mathbf{s}^{(k)}\|_2^2}{2} \\ &= \arg \min_{\mathbf{s}^{(k)}} \|\mathbf{\Sigma}_s\mathbf{s}^{(k)}\|, \text{ dato } \mathbf{y}^{(k)} \\ &= \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)} \end{aligned}$$

che corrisponde alla soluzione dell'equazione (5.5), ovvero $\min \|\mathbf{s}^{(k)}\|_2$ dato $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}$

Nel caso di un prior Laplaciano per $\mathbf{s}^{(k)}$ quello che si ottiene massimizzando il posterior corrisponde a calcolare $\min \|\mathbf{s}^{(k)}\|_1$ dato $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}$ che è il problema di ottimizzazione risolto dall'algoritmo di ricostruzione di *CS*, come viene mostrato anche in [9] e [15].

Capitolo 6

Risultati

Per studiare come varia l'errore di ricostruzione del segnale al variare delle sue proprietà statistiche sono state effettuate delle simulazioni i cui risultati sono riportati in questo capitolo.

6.1 Dimensione Training Set ottima

Si è visto come nel Compressive Sensing supportato dalla Principal Component Analysis siano di fondamentale importanza le caratteristiche statistiche del segnale, utili nel suo processo di ricostruzione. Dal capitolo 5 si è visto come tali informazioni - rappresentate dalla matrice di covarianza - vengano estratte dal training set.

Il training set è la matrice che ha per colonne i vettori $\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^N$, $k < i$ contenenti i valori del segnale stimati negli istanti precedenti a quello preso in considerazione (i). Dalle simulazioni effettuate si è notato come la scelta della finestra di vettori considerati, influenzi in modo significativo l'errore di ricostruzione.

Nella figura 6.1 vengono presentati i grafici di diversi esperimenti che mostrano come varia l'errore in funzione della dimensione del training set per diversi valori del coefficiente di correlazione temporale ($\rho_t \in \{0, 0.2, 0.4, 0.6, 0.8, 0.9\}$) e spaziale ($\rho_s \in \{0.001, 0.1, 1, 5, 10\}$). Quello che si osserva dai grafici è che,

Risultati

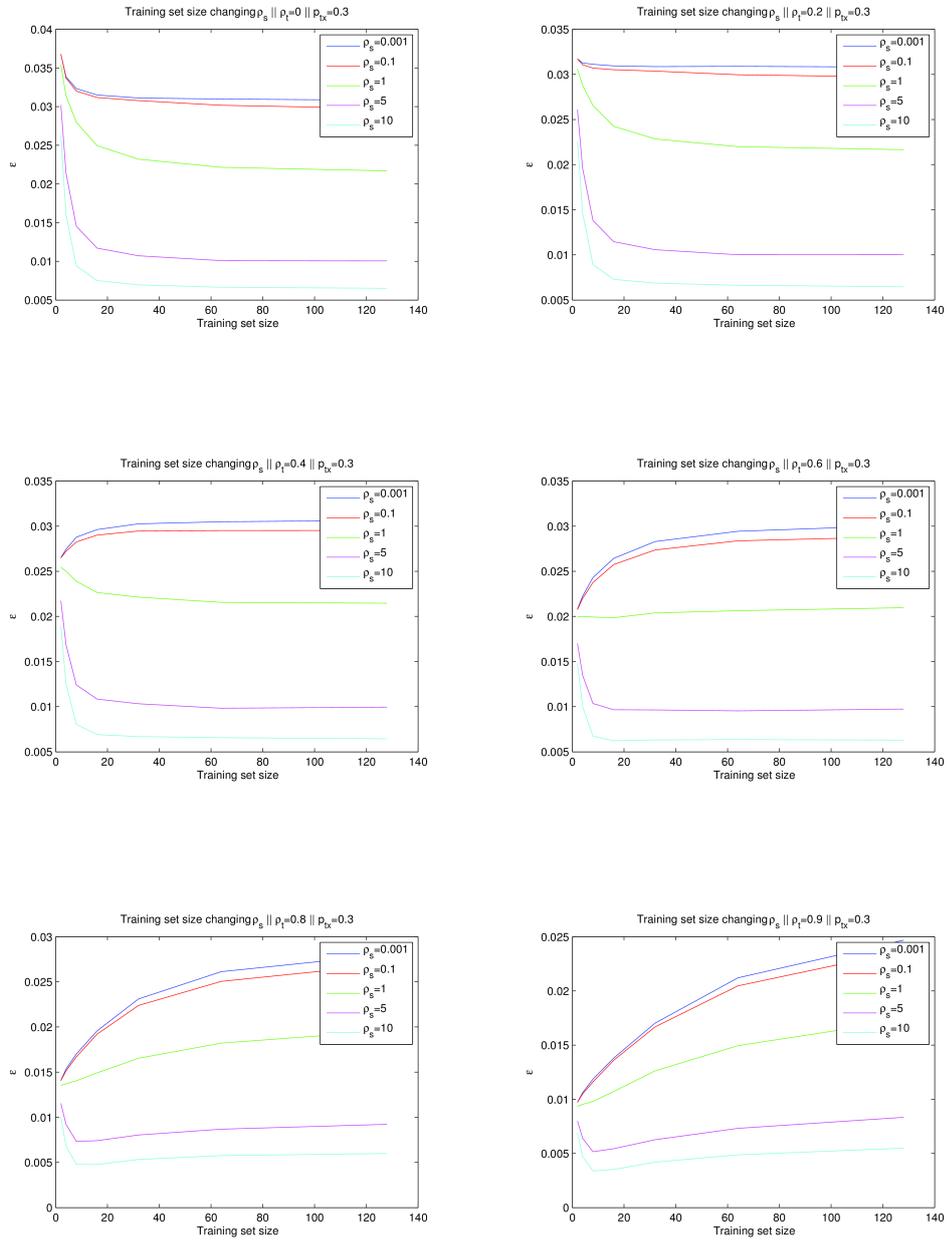


Figura 6.1: Dimensione Training Set

6.2 Correlazione spaziale e temporale

all'aumentare della correlazione temporale, dimensioni del training set piccole forniscono risultati migliori in termini di errore assoluto. Questo rimane vero per correlazioni spaziali medio basse, in quanto per segnali fortemente correlati nello spazio invece la dimensione ottima si scosta leggermente dal minimo. In particolare con un segnale di lunghezza pari a 128 istanti temporali si ha una dimensione ottima della finestra per $\rho_t = 0.9$ e $\rho_s = 5$ di 8. Questo riflette la struttura della matrice di covarianza, la quale è tempo-variante per cui se il segnale è molto correlato prendere campioni lontani nel tempo da origine a una stima errata della stessa. Quando invece la correlazione è bassa, anche i campioni lontani sono significativi e quindi la dimensione della finestra per la stima cresce.

6.2 Correlazione spaziale e temporale

Studiando la correlazione temporale si è visto che - da alcuni risultati qui non riportati - la maggior variazione dell'errore si ha per valori alti di ρ_t . Infatti, come viene evidenziato dalla figura 6.2, a partire da un valore di ρ_t pari a 0.90 si ha una decrescita esponenziale dell'errore. Inoltre si osserva come segnali poco correlati nello spazio ($\rho_s \in \{0.001, 0.01, 0.1\}$) hanno un andamento simile, mentre per quelli soggetti ad una forte correlazione spaziale ($\rho_s \in \{1, 5, 10\}$) la differenza d'errore è più marcata. Il risultato viene confermato e reso maggiormente evidente se si considera l'approccio teorico mostrato in figura 6.3, dove viene utilizzato per il calcolo della base per la PCA un training set composto da valori del segnale originale e non più stimati.

Analizzando invece il comportamento dell'errore al variare delle statistiche spaziali del segnale si è notato un comportamento ambiguo per bassi valori di ρ_s . Questo fatto probabilmente è legato agli elementi affetti da errore che si trovano nel training set e che non riescono ad essere compensati quando la correlazione è bassa. Infatti, essendo la PCA stimata da un modello incompleto del segnale, viene introdotto un errore di stima nel calcolo della matrice

Risultati

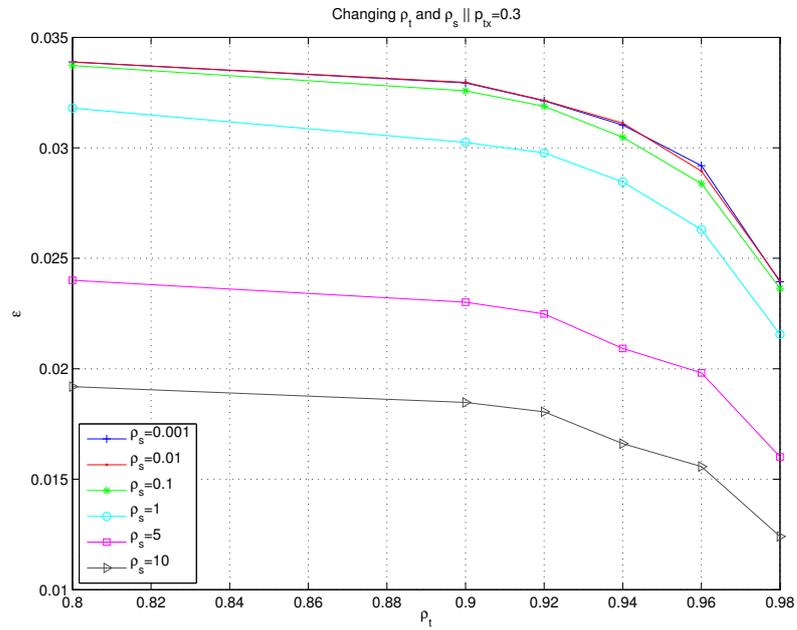


Figura 6.2: Errore in funzione della correlazione temporale

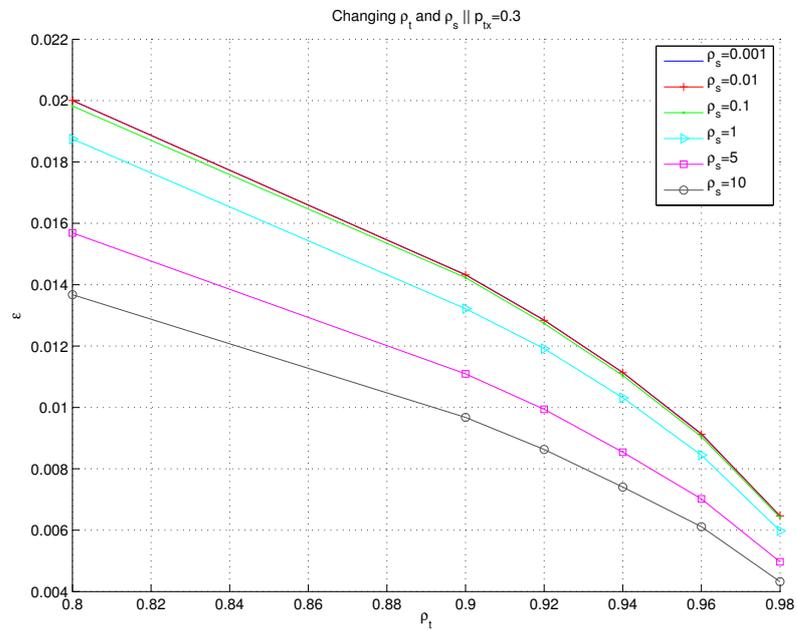


Figura 6.3: Errore in funzione della correlazione temporale con TS ottimo

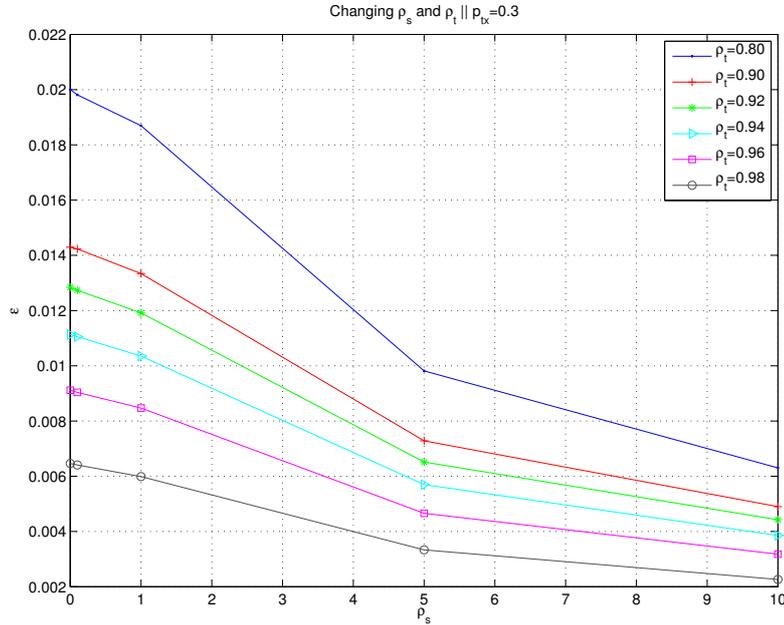


Figura 6.4: Errore in funzione della correlazione spaziale con TS ottimo

di covarianza e quando la correlazione è bassa, questo errore di stima risulta essere dominante. Questa affermazione è dimostrata se si considera il caso teorico in cui nel training set vengono inseriti solamente valori provenienti dal segnale originale, e quindi per definizione privi di errore: all'aumentare della correlazione spaziale - come ci si aspetterebbe - si nota un comportamento monotono decrescente dell'errore (figura 6.4). L'effetto della correlazione temporale invece si riflette nelle curve in una loro traslazione verso il basso proporzionale al valore di ρ_t .

6.3 Consumo energetico

Considerando ora l'analisi da un punto di vista energetico, si ricorda come in questo contesto le politiche di risparmio si applicano impostando in modo appropriato la probabilità di trasmissione nei singoli nodi. Sussiste infatti un rapporto di proporzionalità diretta tra la probabilità di trasmissione e

Risultati

l'ammontare di energia totale richiesta dalla rete di sensori.

Il calcolo dell'energia spesa all'istante k è definito dall'equazione (6.1) e viene effettuato facendo uso di una *routing table* R che per ogni nodo i fornisce il numero di *hop* necessari a raggiungere il punto di raccolta dei dati, ovvero il numero di trasmissioni effettuate dal sistema per trasmettere un dato dal nodo al DCP. Per ogni hop viene tenuto conto di una trasmissione e una ricezione, il cui costo energetico dipende dall'architettura dei nodi.

In questo caso specifico, lo studio del consumo energetico è stato fatto considerando un header di 13 byte e un payload di 2 byte per un totale di 120bit per pacchetto. I costi di trasmissione e ricezione per bit sono stati considerati in entrambi i casi pari a 430nJ/bit¹. L'energia invece necessaria al nodo per generare il numero casuale che determina la trasmissione è stata trascurata in quanto poco significativa rispetto a quella spesa per le operazioni compiute dal modulo radio.

$$E^{(k)} = \sum_{i \in tx \text{ nodes in } k} (R(i) * 430 + (R(i) - 1) * 430) \quad (6.1)$$

Si noti che per l'ultimo hop, non viene contato il consumo energetico relativo alla ricezione dei dati da parte del DCP perché per ipotesi è stato assunto connesso ad una fonte non esauribile di energia. Ecco il motivo per cui nella formula compare il termine $R(i) - 1$.

La tabella R invece viene calcolata partendo dal DCP e andando a verificare quali sono i suoi "vicini" ossia i nodi che sono ad una distanza inferiore a TR dove TR è il *Transmission Range*: raggio di copertura di ogni singolo nodo. Tali sensori avranno $R(i)$ corrispondente a 1 e vengono marcati come selezionati. Successivamente si vanno a controllare i restanti nodi non selezionati e per ognuno di essi si impone $R(i) = \min_{j \in \text{nodi vicini a } i} \{R(j)\} + 1$. Chiaramente questo algoritmo funziona solo nel caso in cui il grafo che rappresenta la matrice di adiacenza dei nodi sia connesso.

In questa analisi si è fatto un confronto con un altro metodo - POLS (*Prob-*

¹Questo valore è stato calcolato dalle specifiche del modulo radio CC2420 presente nei sensori TMote Sky considerati in questo lavoro

6.3 Consumo energetico

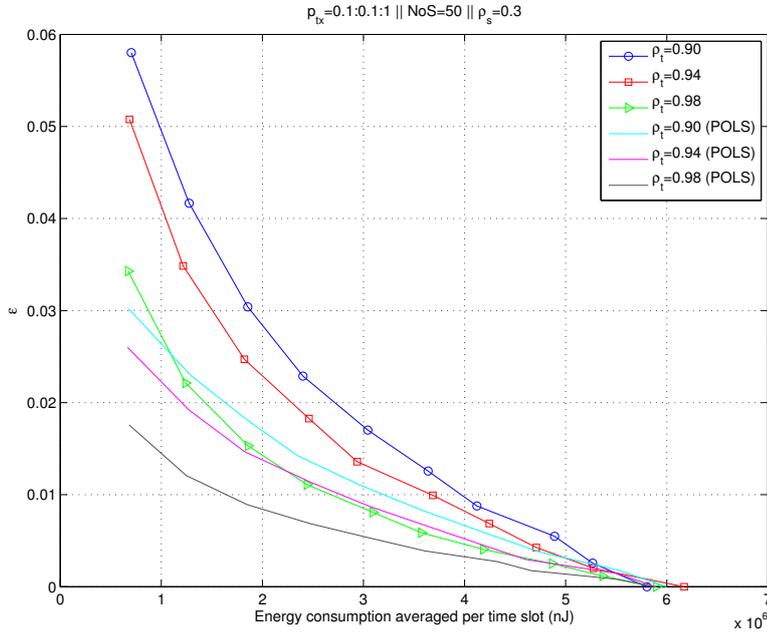


Figura 6.5: Errore in funzione dell'energia e ρ_t

abilistic Ordinary Least Square) - che si differenzia per la strategia utilizzata nel calcolo del vettore sparso \mathbf{s} . I risultati sono rappresentati nelle figure 6.5 e 6.6 rispettivamente con $p_{tx} = 0.3$ e $\rho_t = 0.3$ fissate nel primo caso e $p_{tx} = 0.3$ e $\rho_t = 0.98$ nel secondo.

Ancora una volta i risultati confermano il comportamento atteso in quanto si vede come all'aumentare del coefficiente che esprime la correlazione temporale si ha un miglior rapporto energia/errore che per basse probabilità porta anche a dimezzare l'errore.

Dal punto di vista energetico è interessante notare che riducendo al 30% il numero dei nodi sensore che trasmettono, si riesce a ridurre ad un quarto il consumo energetico ottenendo un errore assoluto pari a 0.025 che si traduce in un errore relativo vicino all' 8%.

Nella figura 6.7 invece si può vedere da un'altra prospettiva quanto il training set scelto possa influenzare l'errore e che benefici questo può portare in termini di energia spesa dal sistema. La prima curva (*TS full*) mostra l'andamento

Risultati

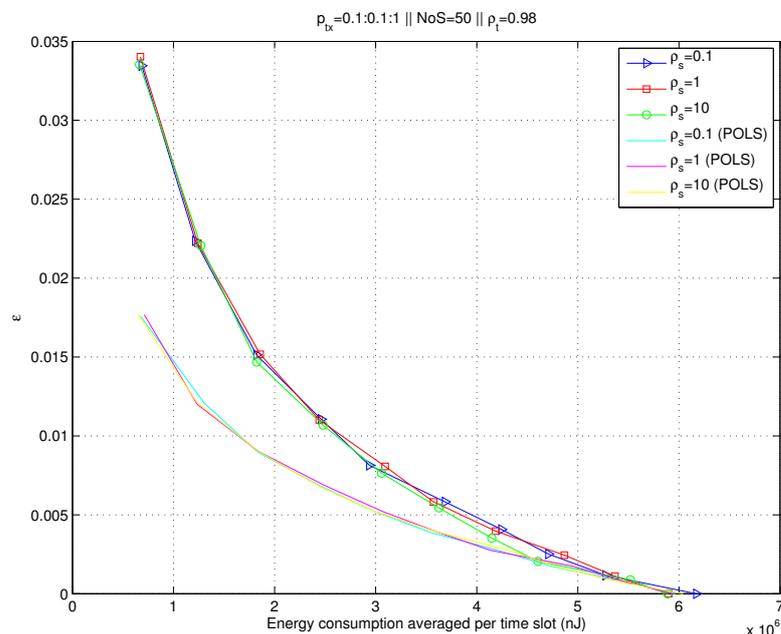


Figura 6.6: Errore in funzione dell'energia e ρ_s

dell'errore quando si inseriscono nel training set, per il calcolo della matrice di covarianza, tutti i segnali ricostruiti dall'inizio del campionamento all'istante precedente a quello considerato (k): $TS = \{\hat{\mathbf{x}}^{(1)}, \dots, \hat{\mathbf{x}}^{(k-1)}\}$. Nella seconda (*TS short*), il training set considerato invece è formato solamente dagli ultimi due campioni ricostruiti: $TS = \{\hat{\mathbf{x}}^{(k-2)}, \hat{\mathbf{x}}^{(k-1)}\}$. Infine, per il training set utilizzato nella terza curva (*TS theo*) sono sempre stati selezionati gli ultimi due campioni, questa volta però provenienti dal segnale originale: $TS = \{\mathbf{x}^{(k-2)}, \mathbf{x}^{(k-1)}\}$, simulando il caso in cui l'errore di ricostruzione sia nullo.

La figura 6.7 è particolarmente importante perché fornisce un *lower bound* per le prestazioni, rappresentato dalla linea *TS Theo*, che evidenzia quali risultati in termine di errore potrebbe ottenere CS con l'utilizzo di tecniche in grado di ricostruire il segnale in modo più preciso.

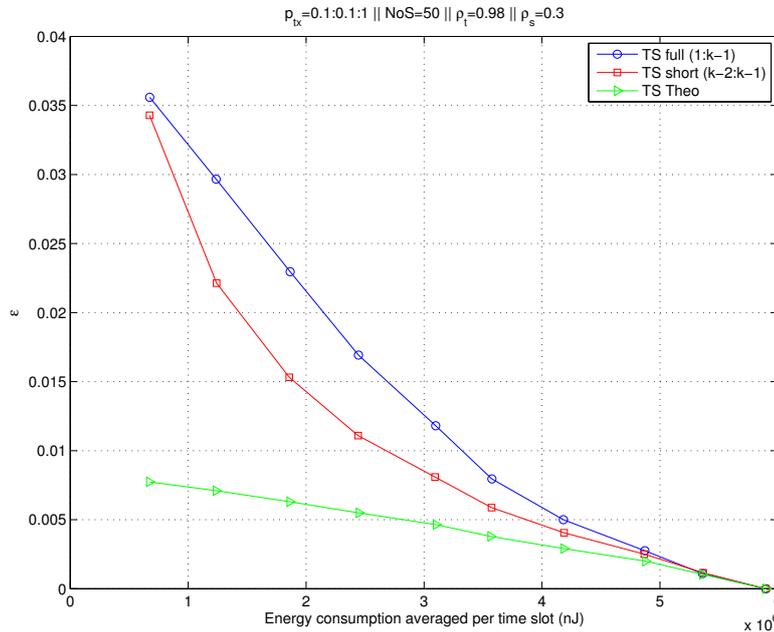


Figura 6.7: Errore in funzione dell'energia e TS

6.4 Clustering

Per completare lo studio, sono state fatte delle simulazioni che verificano le prestazioni di ricostruzione del segnale utilizzando un approccio che suddivide la rete in *cluster* e applica CS-PCA localmente, in ogni gruppo di sensori.

L'algoritmo utilizzato per il clustering opera in due fasi: nella prima seleziona i *cluster head* grazie alla matrice di adiacenza; nella seconda assegna ad ogni nodo della rete il proprio cluster di appartenenza per mezzo della matrice di distanza. Infatti, una volta ordinati i nodi in ordine decrescente secondo il loro numero di vicini, viene selezionato il primo nodo della lista e gli viene assegnato il ruolo di *cluster head*. Il successivo cluster head viene selezionato in modo che la sua distanza da quelli precedenti sia maggiore del suo raggio di trasmissione. Terminata questa procedura e avendo così selezionato tutti i cluster head, i restanti nodi vengono assegnati al cluster controllato dal

Risultati

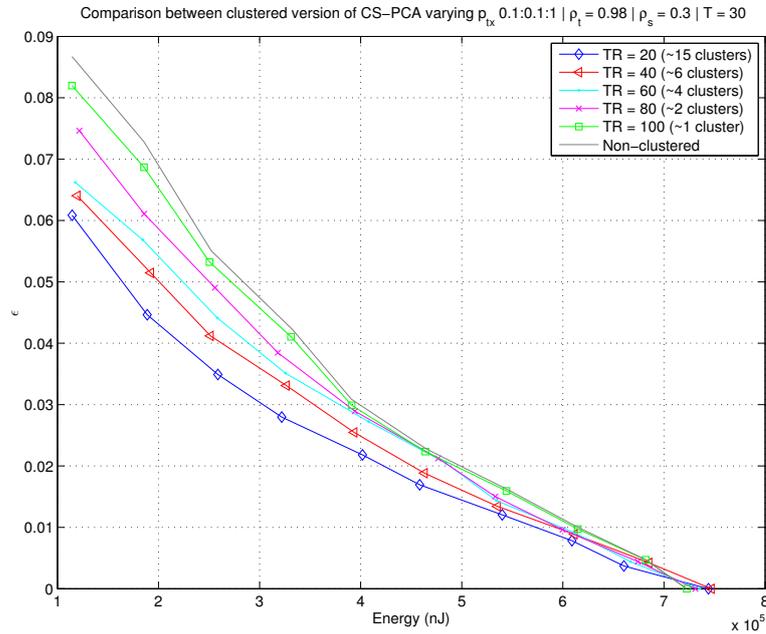


Figura 6.8: Errore in funzione della dimensione dei cluster

cluster head più vicino, secondo quanto definito nella matrice delle distanze. Le politiche di instradamento sono invece rimaste invariate, ossia il calcolo dell'energia spesa viene effettuato come nella sezione 6.3 senza tenere in considerazione il cluster di appartenenza o eventuali algoritmi di aggregazione presso il cluster head.

Dai risultati ottenuti (figura 6.8) emerge che maggiore è il numero di cluster, migliore è l'accuratezza con cui viene ricostruito il segnale. Il fatto che il clustering ottenga prestazioni migliori è segnale che una distribuzione più omogenea dei dati campionati dalla rete aiuta la ricostruzione. Quindi, con un algoritmo di campionamento o scheduling che tenga in considerazione la topologia della rete probabilmente si riescono a migliorare ulteriormente le prestazioni.

Si sottolinea come riuscire a migliorare la qualità della stima del segnale in un istante temporale, non solo implica la riduzione dell'errore sul singolo istante temporale, ma a sua volta si ripercuote in una migliore stima della matrice

di covarianza.

6.5 Confronto tra i vari algoritmi

Si propone infine un confronto fra diversi algoritmi di compressione utilizzati nelle reti di sensori radio. In particolare nella figura sono mostrati *LTC*, *DCT*, *DSC* - descritti nella sezione 1.2.1 - *CS-RND* (*Random Node Selection*), *CS-DNS* (*Deterministic Node Selection*) e *CS Theo*.

CS-DNS è una versione modificata di CS-RND (realizzazione del Compressive Sensing descritto nel capitolo 4) dove vengono attuate delle politiche di scheduling deterministiche. Nella fattispecie, in funzione della probabilità di trasmissione (p_{tx}) imposta dal sistema, i sensori vengono equamente distribuiti in $\lceil N/M \rceil$ sottoinsiemi, dove N è il numero totale di nodi nella rete e $M = Np_{tx}$. Il tempo viene diviso in round formati da $T = \lceil N/M \rceil$ istanti temporali ciascuno. A rotazione, in ogni istante temporale lo stesso numero di nodi per sottoinsieme campiona il segnale. Questo viene fatto in modo tale che alla fine del round tutti i nodi abbiano campionato il segnale almeno una volta. Terminato il k -esimo round, con k intero predefinito, ogni sensore aggrega i propri dati raccolti in un uno (o più) pacchetti che vengono trasmessi al DCP. Il DCP, conoscendo a priori lo scheduling imposto ai nodi e il valore di k , applica gli algoritmi per la ricostruzione dei dati e fornisce la stima del segnale nei kT istanti temporali del round.

La sostanziale differenza tra il consumo energetico di CS-RND rispetto agli altri algoritmi è dovuta al fatto che in CS-RND non si opera nessuna azione di aggregazione dei dati. Infatti i pacchetti dei nodi selezionati vengono inviati ad ogni istante temporale e non ogni T (o kT) istanti come accade per gli altri, facendo così incrementare i consumi. Questo fatto è aggravato dalla grossa inefficienza che si ha con CS-RND se si considera il throughput del sistema in quanto si trasmettono pacchetti con payload molto piccolo (2 byte) rispetto all'header (13 byte), andando a sprecare la maggior parte del traffico utile.

Risultati

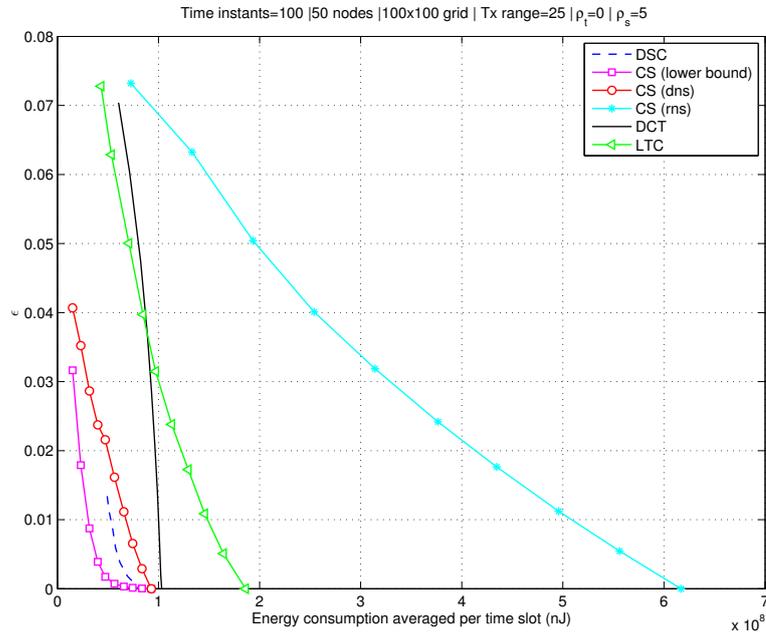


Figura 6.9: Confronto tra vari algoritmi di compressione

In generale è emerso che LTC, DCT, DSC sono tutte valide alternative e la loro scelta dipende fortemente dalle statistiche del segnale. In particolare per segnali correlati nello spazio un approccio DSC conviene, altrimenti, se si considera il tempo, DCT e LTC hanno prestazioni migliori dove DCT è preferibile in termini di consumo energetico in quanto evita di inviare per ogni dato due coefficienti come fa LTC.

Si pone infine l'attenzione sulla riduzione del gap tra CS-Theo e CS-DNS, la quale è attribuibile al miglioramento della qualità della stima della matrice di covarianza ottenuta grazie ad una migliore distribuzione del campionamento, insito nella selezione deterministica dei nodi che trasmettono.

Capitolo 7

Conclusioni

In questa tesi si è affrontato il problema della raccolta dei dati in reti di sensori radio andando ad analizzare in dettaglio uno degli algoritmi che viene utilizzato per ricostruire il segnale originale a partire da una sua versione incompleta. La ricostruzione viene effettuata al punto di raccolta e non comporta nessun aumento della complessità delle operazioni svolte dai sensori. Questo comporta un duplice vantaggio, sia energetico che economico. Infatti da una parte diminuisce il numero di nodi che trasmettono con un conseguente risparmio energetico e, allo stesso tempo, le capacità computazionali necessarie al sensore sono molto limitate, per cui il processing può essere effettuato con hardware a basso costo.

La tecnica di compressione e ricostruzione proposta è il Compressive Sensing, affiancata alla Principal Component Analysis necessaria per trovare una matrice di trasformazione che renda sparso il segnale in ingresso. Oltre allo studio teorico, queste tecniche sono state utilizzate nel framework *SCoRe1* - simulatore di reti di sensori radio - in combinazione con un generatore di segnale, creato ad-hoc per studiare il loro comportamento al variare delle caratteristiche statistiche del segnale.

Dai risultati ottenuti dalle simulazioni effettuate si è potuto osservare come la scelta del training set giochi un ruolo di fondamentale importanza per una ricostruzione accurata del segnale. In particolare si è visto come training set

Conclusioni

di piccole dimensioni e accurati sono preferibili nella ricostruzione di segnali che sono molto correlati nel tempo. Viceversa, per segnali scarsamente correlati training set grandi ottengono risultati migliori.

In generale si è stato verificato che più i segnali sono correlati, nello spazio e nel tempo, minore è l'errore che si compie nella ricostruzione del segnale.

Inoltre, l'utilizzo di un training set formato dai valori estratti direttamente del segnale originale, ha permesso di individuare un lower bound teorico che mostra quali sono le reali potenzialità di CS e qual'è il gap ancora da colmare.

Infine si è visto come gli approcci che sfruttano l'aggregazione hanno prestazioni significativamente migliori rispetto al CS standard. Questo al prezzo di un'applicazione che deve essere necessariamente *delay tolerant*. Altrimenti, se è importante un controllo real time, la soluzione ideale è la versione di CS così come descritta nei capitoli 4 e 5 di questa tesi.

7.1 Sviluppi futuri

La tecnica del Compressive Sensing appare molto promettente, soprattutto per i risparmi in termini economici ed energetici sopracitati. Si è visto come in CS l'approccio che fa utilizzo dei cluster ha prestazioni migliori, per questo motivo potrebbe essere interessante investigare una soluzione che tenga conto delle statistiche del segnale per suddividere in modo opportuno l'area di interesse e sfruttare al meglio queste informazioni. Alternativamente si potrebbero affinare le tecniche di scheduling adottate nella versione di CS-DNS (*Deterministic Node Selection*) che si è visto avere prestazioni migliori in un ambito in cui è permessa l'aggregazione dei dati.

L'altro aspetto su cui si può portare un contributo migliorativo è la scelta di una diversa tecnica per la costruzione delle matrici di trasformazione, che in questo lavoro è svolta dalla Principal Component Analysis, ma che è possibile effettuare in numerosi modi diversi. Una matrice di trasformazione diversa si traduce in un diversa rappresentazione sparsa del segnale, fatto che può fare la differenza nell'errore commesso durante la ricostruzione delle misure.

Concludendo, Compressive Sensing si è rivelata essere una tecnica estremamente interessante anche se forse ancora non del tutto matura e l'ampio gap che c'è dal limite teorico fa intuire quanto lavoro sia ancora possibile svolgere per migliorarla.

Bibliografia

- [1] **KC. Barr, K. Asanovic**, “Energy-aware lossless data compression”, *ACM Trans. Comput Syst*, vol. 24, pp. 250-291, 2006
- [2] **E. Fasolo, M. Rossi, J. Widmer, M. Zorzi**, “In network aggregation techniques for wireless sensor networks: a survey”, *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70-87
- [3] **C. Guestring, P. Bodik, R. Thibaux, M. Paskin, S. Madden**, “Distributed regression: an efficient framework for modeling sensor network data”, *International Conference on Information Processing in Sensor Network (IPSN)*, 2004
- [4] **M. Gastpar, P. Dragotti, M. Vetterli**, “The distributed Karhunen-Loève transform”, *IEEE Transaction Information Theory*, vol. 52, no. 12, pp. 5177-5196, 2006
- [5] **A. Amar, A. Leshem, M. Gastpar**, “Recursive implementation of the distributed Karhunen-Loève transform”, *Transaction Signal Process*, vol. 58, pp. 5320-5330, 2010
- [6] **T. Srisooksai, K. Keamarungsi, P. Lamsrichan, K. Araki**, “Practical data compression in wireless sensor networks: A survey”, *Journal of Network and Computer Applications*, vol. 35, pp. 37-59, 2012
- [7] **M.C. Varuan and I.F. Akyildiz**, “Spatial-Correlation-Based Collaborative Medium Access Control in Wireless Sensor Networks”,

BIBLIOGRAFIA

- IEEE/ACM Trans. on Networking*, vol. 14 no. 2, pp. 316-329, Apr. 2006.
- [8] **D. Donoho**, “Compressive Sensing”, *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 4036-4048, Apr. 2006.
- [9] **E. Candès, J. Romberg and T. Tao**, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”, *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489-509, Feb. 2006.
- [10] **D. Zordan, G. Quer, M. Zorzi and M. Rossi** “Modeling and Generation of Space-Time Correlated Signals for Sensor Network Fields”
- [11] **A. M. Yaglom**, “An Introduction to the Theory of Stationary Random Functions”, Dover Phoenix Editions, 1962.
- [12] **R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer and M. Zorzi** “Data Acquisition through joint Compressive Sensing and Principal Component Analysis”, in *IEEE SenseApp Workshop*, Denver, CO, US, Dec. 2010, pp. 937-944.
- [13] **G. Quer, D. Zordan, R. Masiero, M. Zorzi and M. Rossi** “WSN-Control: Signal Reconstruction through Compressive Sensing in Wireless Sensor Networks”, in *IEEE Globecom 2009*, Honolulu, Hawaii, US, Nov. - Dec. 2009.
- [14] **J. Shlens**, “A tutorial on Principal Component Analysis”, Apr. 2009
- [15] **S. Berker, J. Bobin and E. Candès**, “NESTA: a fast and accurate first order method for sparse recovery” *SIAM J Imaging Sciences* vol. 4, no. 1, 2011

- [16] **G. Quer, R. Masiero, G. Pilonetto, M. Rossi and M. Zorzi** “Sensing, Compression and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework”
- [17] **G. Hosein Mohimani, M. Babaie-Zadeh and C. Jutten**, “A fast approach for overcomplete sparse decomposition based on smoothed l^0 norm”, *IEEE Transaction on Signal Processing* vol. 57, no. 1, pp. 289-301, 2009