Algoritmi end-of-line per la diagnosi di processo nella fabbricazione di dispositivi a semiconduttore

Federico Di Palma

3 Novembre 2005
Il secondo passo per compiere una magia è che qualcuno creda in te.
Indice

Introduzione 11

1 La fabbricazione di dispositivi a semiconduttore 13
   1.1 I dispositivi a semiconduttore ................................. 13
   1.2 Il processo produttivo ........................................ 15
      1.2.1 La lavorazione del wafer (Front-End) .................... 16
      1.2.2 L’assemblaggio (Back-End) ............................... 18
   1.3 I controlli ....................................................... 19
      1.3.1 I controlli in linea ......................................... 19
      1.3.2 I controlli fuori linea .................................... 20

2 La diagnosi end-of-line 23
   2.1 Gli obiettivi della diagnosi di processo ..................... 24
      2.1.1 La produttività di un impianto ............................ 24
      2.1.2 Le possibili cause di scarti sistematici .................. 25
      2.1.3 Le principali azioni di una diagnosi di processo ........ 25
   2.2 I dati end-of-line .............................................. 27
      2.2.1 I test parametrici ........................................ 27
      2.2.2 I test elettrici ........................................... 27
   2.3 Gli strumenti di diagnosi esistenti ............................ 30
      2.3.1 Commonality Analysis .................................... 31
3 La classificazione delle Mappe Binarie

3.1 L’identificazione dei fallimenti

3.1.1 La formulazione del problema di Clustering

3.1.2 Possibili approcci di clustering

3.2 K-Means

3.3 Reti Neurali

3.3.1 Adaptive Resonance Theory (ART)

3.3.2 Mappe di Kohonen

3.3.3 Neural Gas

3.4 Mistura di Bernoulliane

4 Confronto dei classificatori per mappe binarie
5 La diagnosi di processo

5.1 Diagnosi mediante Data Mining statistico

5.1.1 L’analisi di una tabella di contingenza

5.1.2 L’identificazione delle possibili cause

5.1.3 Le limitazioni del metodo

5.2 Approccio Model-Based

5.2.1 Il Modello

5.2.2 La formulazione del problema

5.2.3 Criteri di esistenza ed unicità

5.2.4 Il primo algoritmo risolutivo

5.2.5 Il secondo algoritmo risolutivo

5.3 Approccio Probabilistico

5.3.1 La determinazione del prior

5.3.2 Riformulazione del problema

5.4 Confronto su casi reali

5.4.1 I casi testati

5.4.2 I risultati dell’analisi esaustiva

5.4.3 I risultati della stepwise regression
<table>
<thead>
<tr>
<th>5.4.4</th>
<th>I risultati dell’approccio semi-bayesiano</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.5</td>
<td>Approccio Data Mining</td>
<td>106</td>
</tr>
<tr>
<td>5.4.6</td>
<td>Il risultato del confronto</td>
<td>106</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>Il software ACID</th>
<th>107</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Le tecniche scelte</td>
<td>108</td>
</tr>
<tr>
<td>6.2</td>
<td>La struttura del Software</td>
<td>108</td>
</tr>
<tr>
<td>6.3</td>
<td>Le funzionalità aggiuntive</td>
<td>110</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Gli strumenti per la selezione del calo di resa</td>
<td>111</td>
</tr>
<tr>
<td>6.3.2</td>
<td>I filtri di processo</td>
<td>112</td>
</tr>
<tr>
<td>6.3.3</td>
<td>La valutazione dell’analisi</td>
<td>113</td>
</tr>
<tr>
<td>6.4</td>
<td>I risultati ottenuti</td>
<td>114</td>
</tr>
</tbody>
</table>

**Conclusioni**

<table>
<thead>
<tr>
<th>7</th>
<th>Appendici</th>
<th>117</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Derivazione dell’algoritmo EM</td>
<td>117</td>
</tr>
<tr>
<td>7.2</td>
<td>Utilizzo pratico dei criteri di esistenza ed unicità</td>
<td>121</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Verifica di appartenenza ad $O^1$</td>
<td>121</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Verifica di appartenenza ad $O^2$ ed $O^3$</td>
<td>121</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Riduzione della tabella</td>
<td>122</td>
</tr>
<tr>
<td>7.3</td>
<td>Catalogazione di dati reali</td>
<td>124</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Descrizione degli indici</td>
<td>124</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Definizione dell’esperimento</td>
<td>127</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Confronto fra gli indici</td>
<td>128</td>
</tr>
</tbody>
</table>

**Bibliografia**

<table>
<thead>
<tr>
<th>7.3</th>
<th>Catalogazione di dati reali</th>
<th>124</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.3.1</td>
<td>Descrizione degli indici</td>
<td>124</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Definizione dell’esperimento</td>
<td>127</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Confronto fra gli indici</td>
<td>128</td>
</tr>
</tbody>
</table>
Indice delle Figure

1.1 Schema di un semplice circuito elettrico e sua realizzazione ....... 14
1.2 Wafer semiconduttore .................................................. 15
1.3 Processi di impiantazione e diffusione ionica ....................... 18
1.4 Marcatura dei dies difettosi ai test elettrici ........................ 20

2.1 Esempio di Storia Lotto - (i nomi sono stati cambiati per motivi di
confidenzialità) ............................................................. 26
2.2 Mappa $EWS$ completa .................................................. 29
2.3 Mappa $EWS$ binaria - rappresentazione esadecimale ............... 30
2.4 Mappa $EWS$ binaria - rappresentazione grafica .................... 31

3.1 Mappe $EWS$ binarie ..................................................... 38
3.2 Pattern delle mappe c,e,f,g di Figura 3.1 ............................. 39
3.3 Rielaborazione della mappa $EWS$ binaria in forma vettoriale .... 40
3.4 Neurone Artificiale ....................................................... 44
3.5 Schema Funzionale di un Classificatore Neurale .................... 45
3.6 Griglia bidimensionale di una mappa di Kohonen .................. 48

4.1 Funzioni di probabilità utilizzate nei dati simulati .................. 62
4.2 Classi identificate dal metodo $ART1$ al variare della verosimiglianza $\rho$ 65
4.3 Taratura del numero dei parametri per l’algoritmo $SOM$ .......... 66
4.4 Taratura dei parametri per l’algoritmo Neural Gas ................. 67
### INDICE DELLE FIGURE

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Confronto fra le tecniche di inizializzazione per l’algoritmo SOM</td>
<td>71</td>
</tr>
<tr>
<td>4.6</td>
<td>Confronto fra le tecniche di inizializzazione per l’algoritmo Neural Gas</td>
<td>73</td>
</tr>
<tr>
<td>4.7</td>
<td>Confronto fra le tecniche di inizializzazione per l’algoritmo K-Means</td>
<td>74</td>
</tr>
<tr>
<td>4.8</td>
<td>Confronto fra le tecniche di inizializzazione per l’algoritmo EM</td>
<td>76</td>
</tr>
<tr>
<td>4.9</td>
<td>Confronto dei metodi di classificazione</td>
<td>77</td>
</tr>
<tr>
<td>5.1</td>
<td>d.d.p. di $\eta$ in funzione del parametro $\rho$</td>
<td>95</td>
</tr>
<tr>
<td>5.2</td>
<td>Malfunzionamenti identificati per il primo processo produttivo</td>
<td>99</td>
</tr>
<tr>
<td>5.3</td>
<td>Suddivisione dei wafer nelle 9 classi identificate</td>
<td>99</td>
</tr>
<tr>
<td>5.4</td>
<td>Malfunzionamenti identificati per il secondo processo produttivo</td>
<td>100</td>
</tr>
<tr>
<td>5.5</td>
<td>Andamento dell’FPE in funzione del numero di regressori del modello identificati mediante stepwise regression</td>
<td>104</td>
</tr>
<tr>
<td>6.1</td>
<td>Architettura del software ACID</td>
<td>110</td>
</tr>
<tr>
<td>6.2</td>
<td>Andamento dei bad wafer presenti nei lotti di una selezionata classe</td>
<td>111</td>
</tr>
<tr>
<td>6.3</td>
<td>Andamento temporale dei bad wafer in una selezionata classe</td>
<td>112</td>
</tr>
<tr>
<td>6.4</td>
<td>Distribuzione dei bad wafer su macchine che eseguono la stessa operazione</td>
<td>113</td>
</tr>
<tr>
<td>7.1</td>
<td>Confronto fra le valutazioni degli indici proposti con quelle dell’indice $F$</td>
<td>129</td>
</tr>
</tbody>
</table>
Indice dei Simboli

- **Costanti**
  
  \(N_d\) # dei dies per singola mappa.
  
  \(N_w\) # delle mappe EWS.
  
  \(N_c\) # delle classi identificate.
  
  \(N_l\) # dei lotti.
  
  \(N_m\) # delle lavorazioni.

- **Indici**
  
  \(d\) Die \(i \in [1, N_d]\).
  
  \(w\) Wafer \(w \in [1, N_w]\).
  
  \(c\) Classe o cluster \(w \in [1, N_c]\).
  
  \(i\) Lavorazione \(i \in [1, N_l]\).
  
  \(l\) Lotto \(i \in [1, N_m]\).

- **Grandezze vettoriali**
  
  \(x^w(d)\) : Valore dell \(d\)-simo die nella \(w\)-sima mappa.
  
  \(c^c(d)\) : Valore dell \(d\)-simo die nel pattern della \(c\)-sima classe.
  
  \(\pi(c)\) : Probabilità di ottenere un wafer appartente alla \(c\)-sima classe.
  
  \(\overline{r^w(c)}\) : Probabilità che il wafer \(w\) appartenga alla classe \(c\).
  
  \(R(j)\) : Rendimento in termini di good wafer dell \(j\)-simo lotto.
  
  \(\eta(i)\) : Probabilità di produrre good wafer della lavorazione \(i\).
$g(w)$ : Classe in cui è stata catalogata la mappa $w$.

- **Grandezze matriciali**

  $T(i,j)$ Matrice di processo.

- **Insiemi**

  $I_c$ Insieme contenente le ordinalità dei wafer appartenenti alla classe $c$.

  $W_{good}$ Insieme dei good wafer.

  $W_{bad}$ Insieme dei bad wafer.
Introduzione

Oggi la produzione di wafer a semiconduttore rappresenta uno dei processi più complessi e delicati presenti in ambito industriale. Si pensi che nella produzione di poco più di una decina di lotti sono coinvolte oltre mille lavorazioni differenti. La precisione richiesta impone che molte lavorazioni siano costantemente monitorate e, qualora non risultassero idonee, rimosse dal processo produttivo.

Anche quando tutte le attrezzature coinvolte nella lavorazione di un wafer a semiconduttore soddisfano i criteri di test, non è raro osservare come molti dispositivi risultino difettosi al controllo finale, generando quindi scarti di produzione. La diagnosi di processo è volta all’eliminazione di queste problematiche. Più precisamente, l’obiettivo che ci si pone è quello di individuare la principale causa di un calo di produttività dell’impianto, considerando l’intero processo produttivo od un suo vasto sottoinsieme.

In quest’ambito la tecnica di diagnosi maggiormente diffusa è la commonality analysis. Questa metodologia di analisi presenta molte lacune dovute sia al mancato utilizzo di molte informazioni disponibili sia all’eccessiva semplificazione di altre.

Nella presente Tesi viene proposta una nuova metodologia di analisi denominata AC/ID in grado di sopperire alle lacune riscontrate nella commonality analysis. La metodologia prevede due fasi principali: una legata alla classificazione dei wafer nelle differenti classi di fallimento (AC- Automatic Classification), l’altra relativa alla diagnosi di processo (ID- Interactive Diagnosis).

Nel corso del presente lavoro si sono sviluppate differenti soluzioni per svolgere le due fasi della metodologia AC/ID. Al fine di scegliere il migliore algoritmo, ogni soluzione è stata confrontata con le altre relative alla stessa fase.

La metodologia complessiva è stata sviluppata in collaborazione con la multi-
nazionale STMicroelectronics. L’azienda ha trovato le soluzioni proposte interessanti al punto da finanziare lo sviluppo di un prototipo o test vehicle per sondarne efficacia.

Il buon esito delle prove effettuate su diversi casi reali e l’interesse mostrato dalle persone che hanno utilizzato il test vehicle, ha spinto il reparto di R&D (Ricerca e Sviluppo) della sede di Agrate a sviluppare, in stretta collaborazione con l’autore, un software di diagnosi da integrare con i sistemi aziendali.

Il software di diagnosi ACID (Automatic Classification for Interactive Diagnosis), attualmente in uso presso il sito di Agrate ed in via di installazione presso quello di Castelletto, rappresenta la realizzazione pratica della metodologia descritta nella presente Tesi di dottorato.

La tesi è così strutturata:

**Capitolo 1** : fornisce una breve introduzione al processo di produzione dei dispositivi a semiconduttore.

**Capitolo 2** : dopo aver presentato i requisiti e le principali problematiche legate ad una diagnosi di processo, con particolare attenzione alla Commonality Analysis, descrive la metodologia AC/ID.

**Capitolo 3** : in questa sede vengono illustrate le diverse tipologie di algoritmo proposte per la fase di classificazione. Si presenta inoltre un nuovo modello probabilistico in grado di descrivere le mappe di difettosità dei wafer.

**Capitolo 4** : effettua il confronto fra le tecniche di classificazione. Per ogni soluzione proposta si affronta il problema di effettuare una taratura automatica dei parametri. La scelta viene fatta in base ad un indice appositamente creato per questo tipo di problematiche.

**Capitolo 5** : descrive le soluzioni proposte per la fase di diagnosi. Vengono presentate quattro diverse soluzioni basate su tre differenti approcci. Gli algoritmi proposti sono stati confrontati su casi reali.

**Capitolo 6** : introduce brevemente il software ACID. Particolare risalto viene dato all’architettura ed alle scelte tecniche adottate per rendere la metodologia più efficace.
Capitolo 1

La fabbricazione di dispositivi a semiconduttore

*Less is more*
*Robert Browning*

Negli anni d’oro della fantascienza, gli scrittori immaginavano congegni dalle capacità incredibili, come una piccola stazione di monitoraggio corporea, aventi dimensioni così ridotte da rendere possibile la loro installazione in un anello [1]. Ispirati anche da quei sogni, generazioni di scienziati e ingegneri hanno realizzato dispositivi sempre più potenti e sempre più piccoli sino a giungere a progettare e realizzare una parte di ciò che i nostri nonni potevano solo immaginare [2] [3].

Un ruolo chiave in queste scoperte è stato ed è tutt’ora interpretato dalla tecnologia a semiconduttore che vede nei circuiti integrati il suo principale strumento di diffusione nella vita quotidiana.

1.1 I dispositivi a semiconduttore

Un semiconduttore è un materiale che possiede proprietà elettriche intermedie tra quelle di un conduttore e quelle di un isolante [4]. Il silicio gode di questa proprietà
CAPITOLO 1. LA FABBRICAZIONE DI DISPOSITIVI A SEMICONDUTTORE

Figura 1.1: Schema di un semplice circuito elettrico e sua realizzazione

e mediante l’inserimento di alcune impurità come atomi di boro o fosforo, è possibile alterarne le caratteristiche chimico-fisiche al fine di esaltare le proprietà d’interesse. Questo processo è detto di drogaggio, ed è la base della costruzione dei moderni circuiti integrati.

Un circuito integrato è un piccolo dispositivo che può implementare complesse funzioni elettroniche. Il prodotto finito è costituito sostanzialmente da due parti: un chip a base di silicio che identifica il dispositivo vero e proprio ed un contenitore, detto package, che svolge sia la funzione di proteggere il chip, sia quella di permettere all’utente di trattare il dispositivo in modo pratico nel contesto in cui verrà inserito collegandoelo elettricamente ad altre strutture attraverso connettori detti piedini (o pin).

In Figura 1.1 viene riportato lo schema di un semplice circuito ed una rappresentazione della sezione trasversale del chip che lo implementa. Come si può notare il chip è costituito da due differenti strati. Il primo strato è rappresentato da silicio drogato positivamente definito substrato. In questo strato sono presenti alcune zone con differenti drogaggi che realizzano buona parte dei componenti di base del circuito integrato. Il secondo strato è formato da un ossido di silicio, ove son costruite, disposte su più livelli, le strutture di collegamento fra le varie giunzioni dello strato sottostante.
1.2 Il processo produttivo

Il processo produttivo attualmente utilizzato per la costruzione di circuiti integrati viene definito monolitico [5]. Il termine monolitico deriva dall’unione delle parole greche *monos*, singola, e *lithos*, pietra; infatti la produzione di circuiti integrati avviene a partire da un singolo cristallo di silicio ad elevata purezza. Questo cristallo, dalla forma cilindrica avente circa un metro di lunghezza e un diametro che può variare da 125 fino a 300 mm, viene successivamente "affettato" per ottenere dei sottili dischi dello spessore di circa 650 µm conosciuti più comunemente come *wafer*. In un unico wafer vengono contestualmente realizzati più circuiti integrati con le stesse funzioni elettroniche, detti *dies*, visualizzabili in Figura 1.2. Per ottenere i vari chip, il wafer viene tagliato separando i vari dies che verranno successivamente inseriti nei rispettivi pakages.

In ambito industriale la realizzazione di un circuito integrato viene solitamente suddivisa in due fasi principali. La prima si occupa della lavorazione del wafer di silicio, mentre la seconda è caratterizzata da tutte quelle attività rivolte alla suddivisione dei chips ed ai loro packaging. Queste due fasi sono comunemente conosciute come *Front-End* e *Back-End*.
1.2.1 La lavorazione del wafer (Front-End)

La lavorazione dei wafer è tutt’ora oggetto di mutamenti volti ad ridurre le dimensioni dei dispositivi elementari contenuti all’interno del singolo chip. Nella moderna tecnologia [6], la corrente scala di integrazione viene denominata Ultra Large Scale of Integration (ULSI) e richiede che in un singolo chip possano essere presenti oltre 10 milioni di dispositivi elementari. La realizzazione di dispositivi così complessi avviene mediante il susseguirsi ed il ripetersi di diverse operazioni elementari, svolte da diverse attrezzature. I wafer, per motivi economici e funzionali, attraversano questa catena in gruppi di più unità, definiti lotto di produzione. Il lotto viene formato all’inizio del front-end e ne rappresenta l’unità di lavoro.

Le diverse operazioni necessarie a produrre lotto con tecnologia ULSI possono essere raggruppate nelle macro operazioni riassunte nel seguito. L’ordine in cui sono presentate non riflette la sequenza temporale con la quale vengono effettuate nella realizzazione del singolo wafer.

**Litografia:** In un circuito elettronico, sia all’interno dello strato di silicio sia in quello di ossido, sono presenti diversi sottostrati, zone con differenti drogature e piste di metal. In molte delle operazioni che consentono la creazione di queste zone le parti da trattare o da non trattare debbono essere indicate o protette mediante uno strato di una resina chimica chiamata fotoresist. La creazione di queste zone avviene sfruttando una particolare proprietà del fotoresist. Questa sostanza, se esposta ad un fascio di luce ad una determinata lunghezza d’onda (365 nm per una lampada di Mercurio o 248,193 nm per un laser ad ultravioletti), modifica le proprie caratteristiche chimiche, rendendo facile la propria rimozione. Per definire le varie zone sarà dunque sufficiente creare uno strato di resina uniforme su tutto il wafer ed irraggiare solo le parti interessate. Il procedimento litografico realizza queste operazioni. La selezione delle zone da irradiare avviene mediante il posizionamento di un reticolo opportunamente disegnato, chiamato maschera, fra l’emettitore ed il wafer. Le zone di resina esposta verranno poi rimosse con delle operazioni di attacco che sfruttano sostanze chimiche dette developer. Il fotoresist non esposto viene lasciato sul wafer in modo da creare una strato protettivo per il materiale sottostante per il successivo step di processo.

**Attacco:** successivamente ad ogni step litografico è effettuata un’operazione cosiddetta d’attacco. Si tratta di uno step di processo che permette la rimozione di
sottili strati di materiale, definiti durante il processo di litografia. Quest’operazione può essere realizzata secondo diverse metodologie: attraverso l’utilizzo di composti gassosi, che vengono ionizzati formando un plasma (tecniche dette quindi di plasma etching o dry) o attraverso l’utilizzo di composti liquidi o solubili (attacco wet). Questi ultimi sono utilizzati soprattutto per la rimozione del resist.

**Planarizzazione:** questo particolare step di processo utilizza tecniche avanzate come il Chemical Mechanical Polishing (CMP). Si tratta di effettuare una deposizione, principalmente di strati di ossido, ma anche di polisilicio o di rame dopo la quale viene utilizzata una speciale attrezzatura a più teste che riduce lo spessore utilizzando alcune sostanze, dette slurry, e contemporaneamente un processo meccanico che porta dei dischi rotanti, detti *pad*, a contatto con il wafer.

**Diffusione:** questo passo è utilizzato sia per introdurre dopanti nei materiali sia per far crescere strati sottili di ossido sul wafer. Per depositare strati di materiale i wafer vengono inseriti in forni ad alta temperatura, tipicamente da 500 a 1000°C, mentre i gas dopanti penetranano nel silicio o reagiscono per costituire strati di silicio ossidato.

**Impiantazione ionica:** questa fase permette di introdurre droganti ad una specifica profondità nel materiale mediante radiazioni ad alta energia (vedi Figura 1.3). Contrariamente alla tecnica di diffusione, l’impiantazione permette un miglior controllo di tutti i parametri coinvolti durante il processo. La differenza rispetto al dopaggio attraverso il processo di diffusione, risiede nel fatto che l’impiantazione è selettiva e permette quindi di introdurre droganti in zone ben precise del wafer definite attraverso un processo litografico.

**Deposizione metallica:** questa operazione ha il compito di realizzare le connessioni elettriche tra diverse celle del circuito integrato e la parte esterna. La deposizione di materiale conduttivo (come Alluminio, Rame e loro composti) avviene tramite tecniche di Chemical Vapour Deposition (CVD) o Physical Vapour Deposition. La deposizione può avvenire in alcuni casi tramite una tecnica chiamata *sputtering*. Quest’ultima consiste nel creare un plasma di ioni di argon che colpiscono la superficie metallica. Gli ioni, urtando il metallo, strappano degli atomi che vengono così proiettati in tutte le direzioni in modo che la maggior parte di essi possano condensare sulla superficie del substrato.

**Passivazione:** i wafer vengono ”sigillati” con uno strato di passivazione per
proteggere i dispositivi da elementi contaminanti e dall’umidità. Il film è solitamente costituito da nitruro di sodio o da ossido di silicio.

**Lappatura:** Questa tipologia di operazioni è solitamente l’ultimo passo della fabbricazione di una wafer e ne comporta la riduzione dello spessore da 650 a 380 $\mu m$. In questa fase viene solitamente deposto un sottile strato d’oro sulla parte posteriore della fetta.

Terminata la fase di Front-End i chip, costruiti su un unico wafer, sono pronti per essere separati. Prima di procedere alle operazioni di Back-End vengono eseguiti dei controlli sul wafer processato.

### 1.2.2 L’assemblaggio (Back-End)

Il primo passo dell’assemblaggio consiste nella separazione dei chip. Al fine di favorire l’operazione di taglio, chiamata *die cutting*, tra un dispositivo e i vicini sono state ricavate delle, *scribe lines* che possono essere considerate come libere. Una volta separati, i dispositivi vengono posizionati su un telaio e connessi alla piedinatura esterna tramite tecniche di *wirebonding*.

Al termine di questa fase il dispositivo è già in grado di eseguire le funzioni elettroniche per le quali è stato progettato ma la mancanza di un adeguato sistema di supporto e protezione ne impedisce ancora il normale utilizzo. A tale scopo, il dispositivo viene inserito in nel proprio package plastico o ceramico. Le caratteristiche del package dipendono sia dal tipo di chip, sia dall’applicazione in cui questo verrà utilizzato.
1.3 I controlli

Il processo di lavorazione di un wafer a semiconduttore sebbene composto da otto diverse tipologie di lavorazione risulta essere fra i più complessi attualmente presenti in ambito industriale. Un buon indicatore di questa complessità è dato dal numero delle operazioni elementari che vengono svolte durante il solo Front-End. Per trasformare un cristallo di silicio in wafer di memorie flash sono infatti richiesti oltre 150 differenti passi di processo.

Non stupisce quindi che, per mantenere una alta produttività, siano presenti in ogni impianto di produzione numerose forme di controllo [7]. Queste tipologie di monitoraggio possono essere riassunte fondamentalmente in due grandi categorie: i controlli in linea ed i controlli fuori linea.

1.3.1 I controlli in linea

A questa tipologia di controllo appartengono tutte le misurazioni atte a verificare il buon funzionamento dell’apparecchiatura che gestisce il processo elementare. Il singolo passo di processo è a tutti gli effetti un processo con un proprio ingresso ed una uscita. Per tenere sotto controllo questa tipologia di processi si deve verificarne sia la stabilità, sia l’effettiva aderenza di quanto prodotto alle specifiche di progetto. A questo scopo vengono fatte delle misurazioni di parametri fisici quali spessori di ossidi, distribuzioni di temperatura, allineamenti di maschere ed altro [8]. Questo tipo di misurazioni vengono detti controlli in linea perché avvengono durante la lavorazione del chip, quando quest’ultimo non è ancora funzionante.

E’ importante rilevare che un controllo in linea può determinare l’arresto di un’apparecchiatura per manutenzione non programmata o l’interruzione della lavorazione di alcuni wafers [9]. In altri casi, se un test in linea dovesse fornire un risultato negativo è possibile, solo per talune operazioni, effettuare una operazione di annullamento della lavorazione difettosa ed immettere nuovamente i wafer malfunzionati nel flusso di produzione. Questa operazione viene detta operazione di rework.
1.3.2 I controlli fuori linea

I controlli fuori linea sono quei controlli che possono essere effettuati solo sul dispositivo funzionante, quindi al termine della fase di lavorazione del wafer; per questo motivo vengono denominati controlli end-of-line. Sebbene concettualmente non siano strettamente legati alla lavorazione del wafer di silicio, questi controlli vengono inclusi nella fase di Front-end. I principali controlli end-of-line vengono divisi in due tipologie: test parametrici e test elettrici.

**Test Parametrico:** Per monitorare l’efficienza dei vari passi di processo, l’eventuale interazione fra più passi o la bontà del progetto di un dispositivo vengono costruite, contestualmente al dispositivo, delle strutture elementari progettate ad hoc. Queste particolari strutture, chiamate TAG, sono posizionate all’esterno del dispositivo nelle scribe lines. Su di un singolo wafer vengono generalmente costruite e testate meno di una decina di TAG. Nel test parametrico si verificano le funzionalità elettriche dei singoli TAG attraverso una serie di misurazioni di grandezze fisiche (resistenze, capacità, correnti ecc.). In alcuni impianti questo tipo di testing viene denominato T84.

**Test Elettrici:** Questa seconda tipologia di test rappresenta di fatto l’ultima fase relativa alla parte di Front-end ed è volta alla verifica delle reali funzionalità di tutti i dispositivi presenti sul wafer di silicio. Vengono infatti effettuati test e misure di natura elettrica sequenzialmente su ogni dispositivo presente sul wafer. Queste misure analogiche prendono il nome di test ingegneristici. Se una misurazione esula dalle specifiche di progetto il dispositivo viene segnalato come guasto mediante una macchia d’inchiostro apposta sul die, come mostrato in Figura 1.4.
1.3. I CONTROLLI

Queste tipologie di test sono molto costose in termini di tempo: si pensi che un chip definito funzionante ha superato oltre 900 diverse misurazioni. Per ottenere un testing efficiente le prove vengono fatte in modalità first fail: la sequenza di test viene interrotta alla prima occorrenza non conforme alle specifiche di progetto. Questa fase del flusso di produzione assume il nome di *Electrical Wafer Sorting (EWS)* o più semplicemente *Sort*.

Nel prossimo capitolo vedremo come i controlli end-of-line possano essere utilizzati per la diagnostica di un impianto di produzione.
CAPITOLO 1. LA FABBRICAZIONE DI DISPOSITIVI A SEMICONDUTTORE
Capitolo 2

La diagnosi end-of-line

La fine è solo l’inizio
San Paolo

Come è facilmente intuibile, il primo obiettivo del reparto produttivo di un’azienda di semiconduttori è quello di garantire un alto numero di dispositivi funzionanti per ogni wafer prodotto. Il verificarsi di una qualunque anomalia in un dispositivo può essere attribuita a molti fattori: un errore in un particolare step di processo, problemi dovuti all’integrazione tra più processi, l’interazione di più marginalità di progetto, il mancato controllo di un parametro di lavorazione ed altro ancora. Non stupisce quindi che in una catena di produzione così lunga e composta di lavorazioni tanto delicate possa accadere che alcuni chips di un wafer siano difettosi, nonostante le attrezzature siano risultate idonee durante i relativi test in-line.

In questi casi, per garantire una buona efficienza dell’impianto produttivo, diventa cruciale l’utilizzo di tecniche che consentano l’individuazione tempestiva della causa del calo di produttività. Ovviamente, queste tecniche non potranno basarsi sui controlli in-line, che non si sono rivelati sufficienti ad individuare il problema, ma dovranno basarsi sui test end-of-line. Queste procedure di vengono denominate procedure di diagnosi di processo o diagnosi end-of-line.

In questo capitolo verranno presentate le principali e più diffuse tecniche di
diagnosi di processo, mostrandone i loro pregi e difetti. Verranno inoltre tratteggiati i fondamenti della metodologia proposta in questa sede.

2.1 Gli obiettivi della diagnosi di processo

Sebbene tutte le attrezzature coinvolte nella lavorazione di un wafer a semiconduttore soddisfino i criteri di test in-line, non è raro osservare come molti dispositivi risultino difettosi al controllo EWS, generando quindi scarti di produzione. La diagnosi di processo è volta all’eliminazione di queste problematiche. Più precisamente, l’obiettivo che ci si pone è quello di individuare la principale causa di un calo di produttività dell’impianto, considerando l’intero processo produttivo od un suo vasto sottoinsieme. La precedente formulazione racchiude al suo interno due differenti concetti: quello di causa principale e quello di calo di produttività.

2.1.1 La produttività di un impianto

Per calo di produttività si intende il manifestarsi di un’abbassamento della produzione di dispositivi funzionanti rispetto alla quantità prevista. In una fabbrica di semiconduttori, definire in maniera precisa e statica la quantità di scarti attesa in una linea di produzione non è facile.Questo perché i vari prodotti sono caratterizzati, oltre che dalle peculiari caratteristiche fisiche, da diversi fattori, fra cui la tipologia del processo costruttivo, lo stato di maturità di un prodotto, il volume di produzione fissato. Questi ed altri fattori concorrono alla definizione quantitativa di produttività attesa. Ad esempio, qualora si considerasse un prodotto basato su di una nuova tecnologia e realizzato con un processo non ancora maturo, si potrebbe ritenere accettabile anche una notevole quantità di scarti per ogni wafer. Diversamente se si esaminasse un prodotto consolidato ad avanzato stato di maturazione, ci si aspetterebbe un basso livello di scarto a livello di testing EWS.

In una produzione si possono dunque verificare due diverse tipologie di scarti: quelli randomici e quelli sistematici. I primi sono endemici, ovvero non sono legati ad una specifica motivazione, bensì a quei fattori che determinano la resa attesa per un determinato tipo di produzione. Diversamente, gli errori sistematici sono imputabili a cause ben precise legate alle attrezzature utilizzate nella lavorazione
dei lotti. La diagnosi di processo si occupa dell’individuazione di questa seconda tipologia di errori e delle loro cause.

2.1.2 Le possibili cause di scarti sistematici

In un processo complesso come quello imposto dalla tecnologia *ULSI*, una imperfezione in un qualsiasi passo della catena produttiva può essere causa di un difetto sistematico.

In prima battuta, si potrebbero considerare le operazioni elementari come possibili cause. Tuttavia, annoverare le sole operazioni elementari quali uniche cause dei problemi in esame non sarebbe accurato. Infatti una singola operazione può essere effettuata da più attrezzature di cui solo una potrebbe essere fuori controllo. La duplicazione delle attrezzature è un fattore pressoché costante negli impianti produttivi ed è legata alla complessità del processo costruttivo. Quest’ultimo richiede tolleranze molto basse che comportano frequenti arresti delle attrezzature coinvolte nelle operazioni più delicate al fine di ricalibrare alcune regolazioni. Non potendo sospendere l’intera produzione diviene chiara la necessità di duplicare le attrezzature a disposizione. Tuttavia anche considerare le sole attrezzature come principali cause degli scarti sistematici potrebbe essere riduttivo. Infatti, un singolo equipment di processo realizza più operazioni durante la produzione di un lotto e spesso accade che solo durante una di queste crei delle anomalie.

In conclusione considerare come fattore scatenante di un calo di resa la sola attrezzatura o la sola operazione elementare può essere fuorviente ai fini di una diagnosi accurata. Diviene quindi impossibile scindere la coppia operazione, attrezzatura. Nel seguito non si farà mai riferimento alla singola attrezzatura ma essa sarà sempre associata all’operazione elementare che realizza. Per motivi di sintesi ci si riferirà alla coppia semplicemente col termine *step di processo* o *lavorazione*.

2.1.3 Le principali azioni di una diagnosi di processo

Alla luce delle considerazioni precedenti, una diagnosi di processo è composta, almeno in prima battuta, da due diverse fasi: una di individuazione (detection) ed una di ricerca (isolation).
Nella prima, mediante l’uso di appropriati dati end-of-line, debbono essere individuati cali di resa sistematici. In un processo produttivo complesso come quello in esame ogni singola lavorazione può produrre un difetto sistematico. Considerando la quantità di lavorazioni in gioco, le tolleranze richieste, la delicatezza dei procedimenti chimici e fisici impiegati, non stupisce che anche in pochi lotti possano esser presenti diverse tipologie di cali di resa sistematici ognuna legata ad una propria causa fisica. Nel seguito ci si referì a queste tipologie con il temine di fallimenti.

Nella seconda fase di una procedura di diagnosi si debbono trovare le lavorazioni responsabili dei vari fallimenti fra le diverse lavorazioni eseguite sui lotti in esame. La causa fisica responsabile di ogni fallimento può essere dovuta al malfunzionamento di una o più lavorazioni. In questa fase della procedura di diagnosi risulta di fondamentale importanza la conoscenza precisa delle varie macchine che hanno lavorato i vari lotti di produzione. Questa informazione, solitamente presente in tutti i sistemi informativi delle aziende di semiconduttori, viene chiamata storia lotti. In Figura 2.1 viene riportato uno stralcio di una storia lotti archiviata presso lo stabilimento di Agrate della ST Microelectronics.

<table>
<thead>
<tr>
<th>Lotto</th>
<th>Operazione</th>
<th>Attrezzatura</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCxxxx</td>
<td>1000</td>
<td>AT1</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1001</td>
<td>AT2</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1002</td>
<td>AT3</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1003</td>
<td>AT3/A</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1004</td>
<td>AT3/B</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1005</td>
<td>AT6</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1006</td>
<td>AT7</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1007</td>
<td>AT8</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1008</td>
<td>AT9</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1009</td>
<td>AT11</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1010</td>
<td>AT11/A</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1011</td>
<td>AT11/B</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1012</td>
<td>AT13</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1013</td>
<td>AT14</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1014</td>
<td>AT15</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1015</td>
<td>AT17</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1016</td>
<td>AT18</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1017</td>
<td>AT19</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1018</td>
<td>AT20</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1019</td>
<td>AT21</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1020</td>
<td>AT22</td>
</tr>
<tr>
<td>ABCxxxx</td>
<td>1021</td>
<td>AT23</td>
</tr>
</tbody>
</table>

Figura 2.1: Esempio di Storia Lotto - (i nomi sono stati cambiati per motivi di confidenzialità)
2.2 I dati end-of-line

Nel primo capitolo abbiamo visto come i controlli end-of-line siano divisi in due categorie principali: i test parametrici e quelli EWS. La prima categoria indica delle misure effettuate su TAG appositamente progettati per operazioni di monitoraggio, mentre la seconda si riferisce alle verifiche di funzionamento eseguite su tutti i dies di un wafer.

2.2.1 I test parametrici

Sebbene i test parametrici diano importanti informazioni sul progetto del dispositivo e sulla qualità di realizzazione delle strutture base del dispositivo, alcune semplici considerazioni di carattere generale portano a preferire l’impiego dei test EWS, quando disponibili, per la costruzione di procedure di diagnosi. La prima considerazione riguarda la scarsità di TAG misurati: generalmente su di un singolo wafer vengono effettuate meno di una decina di misure. Considerando che una fetta ospita fra i 400 ed 1200 dies, un solo TAG dovrebbe essere indicativo del funzionamento di un numero compreso fra gli 80 ed 240 dies. Questo dato appare più rilevante se si considera che in dispositivi ad alta produttività lo scarto ammesso potrebbe essere di meno del 3% dei dispositivi realizzati, ovvero fra 12 ed 36 dies. La seconda considerazione ha carattere topografico. I dispositivi non sono solo fisicamente diversi dai TAG che sono posizionati all’interno delle scribe line ma anche spazialmente lontani. Può quindi accadere che un wafer con uno scarto pari ad oltre la metà dei dispositivi prodotti, abbia alcuni test parametrici in controllo statistico.

2.2.2 I test elettrici

Le apparecchiature presenti nel reparto EWS si connettono in maniera sequenziale a tutti i dispositivi del wafer, rilevando una serie di grandezze elettriche. Per ogni singolo chip valutato, si raccolgono i valori misurati che prendono il nome di test ingegneristici. Questi test sono molto numerosi e per motivi di sintesi come risultato del test EWS si assegna ad ogni die un valore, denominato hard bin o bin, indicativo dell’eventuale tipologia di anomalia verificata. In alcune tipologie di prodotto come le memorie flash per i dispositivi risultanti funzionanti viene fatta un’ulteriore
CAPITOLO 2. LA DIAGNOSI END-OF-LINE

differenziazione tra conformi, a cui viene assegnato il bin 0, e non conformi, cui è associato bin 1. Per i dispositivi guasti vengono utilizzati valori di bin superiore all’unità attribuiti in ragione della fase di testing nella quale è avvenuto il fallimento.

La metodologia di attribuzione del valore di bin usata è di tipo first fail: quando un test previsto dalla sequenza predefinita fallisce, producendo un valore d’uscita risultante esterno ai limiti di specifica del prodotto, la fase di test viene interrotta evitando ulteriori prove, e si assegna il dispositivo ad una classe di scarto. Questo, da un lato velocizza la procedura, dall’altro provoca la perdita di informazioni utili per un’analisi statistica. Ad esempio, un dispositivo potrebbe non risultare idoneo anche per un test successivo a quello in cui si è verificato il primo fallimento.

Rese e Rendimenti

Con il termine resa di un wafer si indica il rapporto, solitamente espresso in percentuale, fra il numero di dispositivi funzionanti ed il gross. Dove il gross è un numero calcolato in base a diversi fattori legati sia alla tecnologia di produzione sia a fattori di processo, e rappresenta il limite teorico di dispositivi producibili sulla fetta di silicio. Essendo un valore limite esso risulta solitamente maggiore del numero di dispositivi fisicamente realizzati.

Per una procedura di diagnosi l’impiego della resa può risultare poco accurato. Infatti un processo potrebbe produrre solo dispositivi funzionanti ma avere una resa inferiore all’unità. Per evitare questa discrepanza si preferisce utilizzare in diagnosi di processo il rendimento di un wafer. Esso viene naturalmente definito come il rapporto fra i dies funzionanti e quelli interamente presenti sulla fetta.

Rese e rendimenti rappresentano la più semplice informazione end-of-line estraiibile dai test EWS.

Mappe EWS Complete

In precedenza è stato asserito che in una produzione possono presentarsi diverse tipologie di problemi sistematici. Il mero dato di resa, sebbene permetta di discernere fra un wafer affetto da problemi sistematici o randomici, in molti casi potrebbe fornire una informazione non qualificante al fine di descrivere un fallimento [10]. Ad esempio, due wafer potrebbero avere lo stesso rendimento ma essere affetti da
2.2. I DATI END-OF-LINE

Figura 2.2: Mappa EWS completa

diverse tipologie di problemi: nel primo gli scarti potrebbero essere stati generati da una errata operazione litografica, mentre nel secondo potrebbero essere stati generati da una difettosità legata ad operazioni di CMP. In questi casi, avere informazioni circa la topologia, fornita dalla posizione dei dies scartati, e la qualità dei fallimenti riscontrati, indicata dal valore del bin, potrebbe essere determinante ai fini di un’accurata diagnosi di processo.

Questo tipo di informazioni è fornita dalla mappa EWS del wafer dove in corrispondenza di ogni die vine riportato il rispettivo bin, come mostrato in Figura 2.2.

Mappe EWS Binarie

Mediante l’utilizzo di una mappa EWS completa è possibile ottenere informazioni relative alla tipologia di fallimento riscontrata nel singolo die guasto. Purtroppo, la procedura “first fail” introduce un problema di mascheramento del bin. Per ovviare a questo problema, spesso si preferisce rinunciare all’informazione relativa al valore del bin a favore di quella certa di dispositivo funzionante e guasto. I wafer viene rappresentato come una mappa binaria, dove i dispositivi funzionanti, siano essi
CAPITOLO 2. LA DIAGNOSI END-OF-LINE

Figura 2.3: Mappa EWS binaria - rappresentazione esadecimale

conformi o non conformi, sono indicati con degli 1 e mentre il valore nullo viene attribuito ai device guasti, come mostrato in Figura 2.3.

Al fine di ottenere una rappresentazione più efficiente e di rapida lettura, la mappa binaria viene convertita in una immagine dove i dispositivi guasti vengono visualizzati con toni scuri, mentre quelli funzionanti con colori chiari. Il risultato è visibile in Figura 2.4.

2.3 Gli strumenti di diagnosi esistenti

I dati end-of-line e la storia lotti contengono una notevole quantità di informazioni. Considerando, ad esempio, un solo lotto, la sua "storia" si compone di diverse centinaia di lavorazioni; inoltre ognuno delle decine di wafers che lo compongono possiede centinaia se non migliaia di dies che vengono sottoposti a centinaia di misure. Risulta quindi ovvia la difficoltà di realizzare uno strumento semplice ed efficace capace di analizzare una così estesa mole di dati.

Negli ultimi anni il data-mining ha rappresentato sicuramente il metodo più importante e più utilizzato per estrarre informazioni da un’ingente quantità di dati.
2.3. GLI STRUMENTI DI DIAGNOSI ESISTENTI

Le metodologie di data-mining sono impiegate in diversi campi d’applicazione tra cui il settore biologico, finanziario, medico ecc. Questo approccio, denominato anche KDD (Knowledge Discovery in Database), permette di organizzare le informazioni registrate in un database, al fine di trovare pattern, modelli e relazioni tra le variabili. Le tecniche di data-mining utilizzano una varietà di approcci: esistono sia tradizionali tecniche statistiche sia altre che coinvolgono sistemi di diversa natura come reti neurali, regole d’associazione, alberi di decisione, algoritmi di clustering ecc. [11], [12].

Nell’ambito della produzione di semiconduttori gli studi di data mining più promettenti utilizzano una metodologia operativa definita analisi di commonalità o Commonality Analysis (CA). Questi studi hanno portato alla realizzazione di strumenti di diagnosi quali, ad esempio, il software Klarity ACE (Advanced Correlation Engine) [13], implementato da KLA ed attualmente in vendita sul mercato.

2.3.1 Commonality Analysis

Lo scopo ultimo di una analisi di questo genere è rappresentato dalla ricerca, laddove possa esistere, di una corrispondenza tra la presenza di alcune caratteristiche del prodotto finito e le macchine impiegate nel processo costruttivo. La metodologia di CA segue uno schema costituito da cinque passi fondamentali [14]:

Figura 2.4: Mappa EWS binaria - rappresentazione grafica
1. *Data Classification*: In questa prima fase i lotti di produzione vengono suddivisi in categorie in ragione del tipo di fallimento che essi presentano. L’obiettivo di questa fase è infatti di riconoscere ed isolare le unità che presentano il fallimento in esame raggruppandole in un insieme.


La scelta dei lotti su cui effettuare l’analisi rappresenta il passo forse più importante. Infatti, la determinazione della numerosità del campione gioca un ruolo importante nello svolgersi dello studio. Da un punto di vista statistico avere a che fare con un numero di lotti elevato potrebbe rappresentare un vantaggio: su un campione più grande, infatti, è maggiore l’informazione a disposizione. Tuttavia, questo è vero solo quando i lotti in esame presentano la stessa tipologia di problema; più conveniente altrimenti, risulta l’utilizzo di un numero minore di lotti.

3. *Data Extraction*: una volta scelti i lotti su cui effettuare lo studio occorre raggruppare la maggior informazione possibile su di essi (numero del lotto, step di processo effettuati, data e ora, misure relative ai test elettrici, ecc.)


5. *Follow up Actions*: Sebbene questo passo sia indicato come integrante della procedura di CA, esula dal concetto di diagnosi di processo, in quanto prevede l’esecuzione, con la giusta tempestività, delle operazioni necessarie al fine di rimuovere il problema in esame.
2.4. LA METODOLOGIA PROPOSTA

Pertanto, lo scopo ultimo di un procedura di CA è il confronto della storia dei lotti a buona e cattiva resa e, in relazione ad esso, il calcolo di un "voto" per ciascuna lavorazione nel processo di fabbricazione. Il voto può anche essere interpretato come la probabilità che la lavorazione in esame abbia generato il problema [15].

2.3.2 Le lacune della Commonality Analysis

Ad una analisi approfondita, il modus operandi della CA presenta alcune lacune.

Il primo limite riscontrato è rappresentato dal fatto che i lotti analizzati vengono considerati come totalmente buoni o totalmente cattivi: non è previsto nella procedura alcun peso relativo dei lotti. Nella CA un lotto con solo un wafer a bassa resa, se incorporato nei bad lots, ha lo stesso peso di un lotto formato da soli wafer a bassa resa. Inoltre, la maggior parte delle procedure di CA non prevedono la presenza di indici o strumenti oggettivi che aiutino l’esperto umano nella classificazione dei wafer o lotti nei due gruppi.

Un’altro importante limite di quest’analisi è dato dal fatto che solo un sottoinsieme, il più delle volte ristretto, dei lotti di produzione partecipa alla fase di data analysis. Di fatto si perdono alcune informazioni che possono risultare preziose per la diagnosi di processo.

L’analisi effettuata in questo lavoro si pone come obiettivo la determinazione di una metodologia per l’analisi di processo che consente il superamento di tali limiti. Ulteriore obbiettivo della tesi è la creazione di uno strumento software che implementi queste metodologie.

2.4 La metodologia proposta

In base a quanto visto nel paragrafo precedente, l’unità informativa della CA risulta essere il lotto di produzione. Infatti, una volta costituiti gli insiemi di good and bad lots, viene considerato il percorso di ogni singolo lotto. Questa considerazione trova il suo fondamento nel fatto che ogni lotto subisce le stesse lavorazioni cosicché risulta naturale utilizzare il lotto come unità informativa. Tuttavia, capita spesso che due wafers appartenenti allo stesso lotto presentino differenti tipologie di problemi. Nella CA questa tipologia di lotti viene scartata.
Questo capitolo tratta la diagnosi end-of-line, un processo che si svolge alla fine dell'asse di produzione, per identificare e valutare gli errori che si verificano durante la produzione di dispositivi elettronici. La diagnosi end-of-line è un'attività essenziale per identificare e classificare in modo accurato gli errori, assicurando la qualità dei prodotti finiti.

La perdita di informazione potrebbe essere risolta considerando una procedura di diagnosi di processo che consideri come unità informativa il singolo wafer, invece del lotto. Tale metodologia consenzirebbe anche la risoluzione delle lacune descritte nel precedente paragrafo. Infatti se tutti i wafers in esame partecipassero alla diagnosi di processo, non solo non ci sarebbe più perdita di informazioni, ma verrebbe a mancare la necessità di un criterio oggettivo che qualifichi un lotto come buono o cattivo.

La procedura proposta in questa sede utilizza come unità informativa il wafer e, ricordando anche quanto espresso al Paragrafo 2.1.3, si articola in tre passi:

1. Identificazione dei guasti. Tutti i wafer in esame vengono analizzati allo scopo di identificare i vari difetti sistematici presenti nei lotti in esame.

2. Selezionare del problema. Un esperto umano seleziona il fallimento che intende sottoporre alla diagnosi di processo.

3. Correlazione di processo. Questa fase fornisce l'indicazione delle lavorazioni che risultano maggiormente legate alla produzione dei wafer che presentano il difetto in esame.

Nella prima fase della procedura i dati end-of-line provenienti da tutti i wafers in esame vengono analizzati. Per quanto esposto in precedenza, le mappe $EWS_1$, siano esse binarie o complete, rappresentano i dati più idonei. Considerando che in una analisi diagnostica di media entità, le fette da analizzare sono dell'ordine delle migliaia; risulta lampante l'esigenza di una procedura automatica che identifichi i difetti sistematici. Nel terzo e nel quarto capitolo della presente si illustrerà la scelta del miglior algoritmo per identificare i cali di resa sistematici basandosi sulla classificazione di mappe $EWS$ binarie.

Nella seconda fase un esperto umano deve scegliere fra le tipologie di guasto ricavate al passo precedente, quelle che meglio descrivono il problema in analisi. In questa scelta viene richiesto a colui che effettua la diagnosi di decidere quali dei guasti identificati possano avere la stessa origine fisica. Nel Paragrafo 6.3.2 sono forniti alcuni criteri utili a tale scopo.

L’obiettivo della terza fase della procedura proposta coincide con il quarto della $CA$: stilare una classifica delle lavorazioni più plausibilmente responsabili del pro-
2.4. LA METODOLOGIA PROPOSTA

Il problema selezionato. Nel quinto capitolo verranno presentati vari algoritmi basati su diversi approcci.
CAPITOLO 2. LA DIAGNOSI END-OF-LINE
Capitolo 3

La classificazione delle Mappe Binarie

divide et impera
Filippo il Macedone

Il primo passo della procedura di diagnosi proposta in questa sede prevede il riconoscimento dei diversi tipi di fallimento presenti nei lotti in esame e l’assegnazione di un solo problema ad ognuno dei wafers analizzati. Questo processo consente l’isolamento delle cause dei cali di resa sistematici. L’identificazione dei vari fallimenti agevola di molto il compito della procedura di diagnosi, che si occuperà di individuare le cause di un solo tipo di fallimento, invece che dell’intera produzione a bassa resa.

Se si considerano i volumi di dati in gioco diviene evidente l’esigenza di realizzare una catalogazione delle mappe EWS in maniera automatica. Quest’obiettivo può essere raggiunto mediante diversi metodi ognuno dei quali presenta proprie peculiarità.

In questa sezione verranno presentate le principali tecniche per l’identificazione dei fallimenti basate sull’analisi di mappe EWS binarie.
3.1 L’identificazione dei fallimenti

Prima di considerare i metodi di identificazione proposti, è opportuno considerare in che modo le mappe EWS binarie sono legate ai fallimenti. In Figura 3.1 sono riportate alcune delle mappe EWS binarie relative ad un prodotto flash, sviluppato presso gli stabilimenti di Agrate della ST microelectronics. Si può notare come alcune di queste mappe abbiano dispositivi non funzionanti nelle stesse posizioni. Ad esempio le mappe dei wafer c, e, f e g presentano dispositivi guasti lungo il bordo della fetta. Parrebbe lecito supporre che i quattro wafer, presentando distribuzioni topologicamente simili, siano affetti dallo stesso problema fisico.

Un modo per rendersi conto velocemente se un gruppo di wafer presenti un’affinità topologica è rappresentato dalla mappa media. Questa mappa viene costruita indicando per ogni die la frequenza relativa dei dispositivi non funzionanti osservati nella stessa posizione delle fette considerate. Se nella mappa media si riscontrano zone a frequenza relativa vicina all’unità i wafer in esame presenteranno
lo stesso pattern d’errore in quelle zone. In figura 3.2 viene riportata la mappa media, visualizzata in toni di grigio, dei wafer c,e,f,g di Figura 3.2; da essa risulta evidente la presenza di un pattern.

Dalla forma dei patterns, detti anche firme d’errore, può essere possibile risalire alla tipologia di lavorazione che ha originato il fallimento [16], [17]. Per esempio pattern a righe o colonne, possono essere generati da lavorazioni di tipo litografico, oppure fallimenti a simmetria circolare, come il bollo o spot riconoscibile nei wafer a, d ed n di Figura 3.1, spesso sono imputabili ad operazioni di CMP.

Pertanto i differenti fallimenti presenti nei lotti di produzione possono essere identificati partizionando le mappe EWS binarie in classi o clusters ognuno accomunato da una propria firma d’errore. Inoltre il pattern caratteristico del singolo cluster fornisce una descrizione del tipo di fallimento presente nei wafer della classe.

3.1.1 La formulazione del problema di Clustering

Eseguire il clustering di un set di dati assegnato significa individuare gruppi di oggetti simili tra loro e dissimili rispetto agli oggetti degli altri cluster. La discriminazione degli oggetti avviene in base ad una qualche misura prestabilita di similarità che spesso è peculiare dell’algoritmo utilizzato. Generalmente la classificazione viene indicata mediante una funzione che associa l’ordinalità dell’oggetto alla classe in cui è stato catalogato. Nel nostro caso viene richiesto di partizionare le $N_w$ mappe
CAPITOLO 3. LA CLASSIFICAZIONE DELLE MAPPE BINARIE

Figura 3.3: Rielaborazione della mappa EWS binaria in forma vettoriale

binarie in $N_c$ classi.

Le tecniche di clustering note in letteratura rappresentano i vari oggetti come punti in uno spazio a più dimensioni. La $w$-sima mappa binaria, per poter essere processata, deve essere rappresentata mediante un vettore $x^w$. La trasformazione adottata definisce le componenti di $x^w$ come le righe della mappa riorganizzate sequenzialmente tramite un’operazione di incolonnamento, come mostrato in Figura 3.3. Ovviamente la dimensionalità di $x^w$ sarà pari al numero di dies $N_d$, che si suppone essere costante in tutti i wafer in analisi.

Nella notazione introdotta il problema di classificare $N_w$ vettori binari $x^w \in \{0,1\}^{N_d}$ in classi $N_c$ può essere definito come la determinazione di un vettore

$$ g = \begin{bmatrix} g(1) & g(2) & \ldots & g(N_w) \end{bmatrix} $$

dove l’elemento $g(w)$ indica il cluster in cui viene posto il $w$-simo vettore binario. La realizzazione del vettore $g$ dipenderà dal tipo di approccio utilizzato.

3.1.2 Possibili approcci di clustering

Vi sono diversi approcci al problema della classificazione automatica:
3.1. L’IDENTIFICAZIONE DEI FALLIMENTI

**tecniche di riconoscimento di forma** Gli algoritmi utilizzati sono in grado di riconoscere specifiche firme d’errore quali ad esempio anelli, bolli, scratches ed altri. Queste forme sono prestabilite ed incorporate nel classificatore.

**tecniche di apprendimento supervisionato** Gli algoritmi di questa categoria, prima di essere utilizzati, necessitano di una fase detta di addestramento. In questa fase preliminare, l’algoritmo configura i propri parametri affinché riesca a classificare correttamente un insieme di dati, definito *training set*. Per svolgere questo compito, il training set deve essere preventivamente classificato utilizzando un qualunque metodo che funge da “supervisore” (un esperto umano per esempio). Terminata la fase di addestramento il classificatore può essere utilizzato su nuovi insiemi di dati per riconoscere i pattern che ha appreso.

**tecniche di apprendimento non supervisionato** Queste tecniche non necessitano di conoscenze pregresse. Si cerca una classificazione che diminuisca il più possibile la distanza fra i dati appartenenti allo stesso cluster. I pattern di riferimento vengono ricavati direttamente dai dati.

**tecniche basate su di un modello probabilistico** Queste tecniche richiedono la formulazione di un modello che descriva il tipo di dati in analisi. La classificazione fornita da questi algoritmi ottimizza, secondo un criterio oggettivo (la verosimiglianza, per esempio), l’aderenza dei dati al modello.


summenzionati svantaggi non son presenti nei restanti due approcci di clustering, che pertanto risultano essere più idonei al tipo di applicazione considerata.

Differenti algoritmi di clustering possono portare a diverse classificazioni. Con l’intento di scegliere il metodo migliore per la classificazione di mappe EWS binarie, in questa sezione verranno descritti due diversi approcci per le tecniche non supervisionate (K-Means e reti neurali) ed un algoritmo di massimizzazione della verosimiglianza, per le tecniche basate su modello probabilistico.

3.2 K-Means

Il metodo denominato K-Means è sicuramente il classificatore più veloce e semplice disponibile in letteratura.

Questa tecnica associa ad ogni classe un centroide che solitamente coincide con il baricentro del cluster. Fissato un set \( A \) di centroidi di partenza, la classificazione avviene mediante l’iterazione di una procedura a due passi. Nel primo passo i dati in ingresso vengono classificati assegnando ogni singolo vettore al centroide ad esso più vicino secondo una qualche metrica. Ultimata la presentazione degli ingressi, si procede col secondo passo, che prevede il calcolo del baricentro di ogni cluster, ed il conseguente aggiornamento del centroide. I due passi vengono iterati, calcolando nuovamente le distanze tra i punti del set di dati ed i nuovi centroidi e riassegnando i dati ai vari cluster. Il processo termina quando, al ripetersi dei passi, non si osserva alcun movimento di oggetti da un gruppo ad un altro [21].

Nell’applicazione in esame, i dati da classificare sono gli \( N_w \) wafer, descritti dai vettori \( x^w \), da suddividere nelle \( N_c \) classi, i baricentri coincidono con le mappe medie e sono descritti dalla:

\[
p^c = \frac{1}{|I_c|} \sum_{w \in I_c} x^w
\]

dove \( |I_c| \) è la cardinalità dell’insieme \( I_c \) dei wafer assegnati alla classe \( c \). Usando come criterio di selezione la distanza euclidea tra ciascun dato e le mappe medie l’assegnamento di un generico wafer al gruppo avente il baricentro più vicino è dato dalla seguente funzione:

\[
g(w) = \arg \min_c \| x^w - p^c \|
\]

dove \( \| x^w - p^c \| \) è la distanza euclidea tra il vettore dei dati d’ingresso \( x^w \) ed il centro
3.3 RETI NEURALI

$p^c$ del cluster $c$. La procedura così definita, ad ogni passo riduce la somma dei quadrati delle distanze:

$$S_W = \sum_{w=1}^{N_w} \| x^w - p^{g(w)} \|^2$$

e può essere sintetizzata dal seguente algoritmo:

1. Si inizializza un set $A$ di $N_c$ centroidi $p^c$, con $c = 1, 2, ..., N_c$.
2. Per ogni vettore di ingresso $x^w$, con $w = 1, 2, ..., N_w$, si esegue una prima classificazione definendo il vettore $g_0$:

$$g_0(w) = \arg \min_c \| x^w - p^c \|$$

3. Si ricalcolano i centroidi, come media delle coordinate di tutti gli oggetti appartenenti a ciascun cluster:

$$p^c = \frac{1}{|I_c|} \sum_{w \in I_c} x^w$$

4. Assegno ogni wafer al gruppo del centroide più vicino tramite la seguente

$$g(w) = \arg \min_c \| x^w - p^c \|$$

5. Se $g = g_0$ allora Fine
6. Pongo $g_0 = g$
7. Torno al passo 3

3.3 Reti Neurali

I modelli di reti neurali sono stati introdotti con molti nomi come parallel distributed Processing systems [22], connectionist systems [22] e neuromorphic systems [23]. Tutti questi modelli tentano di raggiungere buone prestazioni attraverso fitte interconnessioni di elementi computazionalmente semplici.

Gli elementi computazionali, chiamati nodi o neuroni, utilizzati nelle reti neurali sono tipicamente non lineari. Il nodo più semplice, vedi Figura 3.4, è costruito...
CAPITOLO 3. LA CLASSIFICAZIONE DELLE MAPPE BINARIE

Figura 3.4: Neurone Artificiale

in maniera da sommare in maniera pesata i diversi ingressi e trasferire il risultato, attraverso una funzione non lineare, agli altri nodi della rete. Ogni neurone è caratterizzato da una soglia interna e dal tipo di non linearità.

I diversi modelli di reti neurali differiscono, oltre che per tipo di neurone utilizzato, anche nella topologia dei nodi e nella tecnica di apprendimento. Queste tecniche permettono di identificare un insieme di valori di inizio per i pesi delle connessioni ed indicano di quanto e come questi debbano essere riadattati durante l’utilizzo della rete per aumentarne le prestazioni. L’abilità di adattarsi è essenziale nell’esecuzione di compiti molto dinamici ed in continua evoluzione, come ad esempio la classificazione di immagini che presenta nuovi dettagli in ogni dato considerato.

In un classificatore neurale ogni singolo cluster viene associato ad un proprio neurone che memorizza l’informazione relativa al cluster nei pesi delle connessioni del neurone stesso. Nella applicazione considerata quest’informazione è data dalla firma d’errore. Pertanto i pesi delle connessioni di un neurone contengono i valori del pattern di riferimento del cluster associato al nodo.

In Figura 3.5 viene riportato lo schema di funzionamento di un classificatore neurale che determina quale delle M classi sia la più rappresentativa per input N dimensionale in ingresso. In esso possono essere notati tre blocchi distinti. Il primo accetta in ingresso il dato e calcola una misura, chiamata matching score, che indica la vicinanza dell’ingresso alle varie classi. Il secondo blocco seleziona la classe più idonea, mentre il terzo si occupa dell’addestramento della rete. Spesso la rete per poter apprendere, ovvero per determinare con precisione i pesi delle connessioni presenti nel primo blocco, deve esaminare più volte l’intero set di dati.

Nel seguito verranno presentate tre diverse tipologie di reti neurali ed i classificatori di mappe binarie su di esse basati.
3.3. RETI NEURALI

3.3.1 Adaptive Resonance Theory (ART)

Nello schema di Figura 3.5 risulta chiaro come, all’interno di un classificatore neurale, si instauri una retroazione. La chiusura dell’anello, sebbene consenta l’apprendimento [24] della rete può determinarne un comportamento instabile. Infatti l’unità vincitrice potrebbe continuare a cambiare anche quando lo stesso insieme di dati viene presentato in ingresso. Per ovviare a questo problema si potrebbe ridurre il guadagno della reazione, con il rischio che la rete non sia più in grado di riconoscere nuovi cluster e di modificare quelli esistenti. Si deve quindi ottenere uno schema di apprendimento che consenta alla rete di mantenere un comportamento totalmente adattativo (o plastico) in concomitanza di informazioni completamente nuove e contemporaneamente mantenendosi tuttavia stabile (o robusto) nei confronti di quegli stimoli acquisiti o irrilevanti. Questa problematica viene spesso definita dilemma di plasticità e stabilità.

Una possibile soluzione al problema è rappresentata dalla Teoria della Risonanza Adattativa ART di Carpenter e Grossberg [25]. Ogni volta che un dato viene sottoposto alla rete, questa ne valuta la quantità di informazione in rapporto a quelle già immagazzinate. Se viene riscontrato che lo stimolo è sufficientemente vicino alla conoscenza presente in un neurone, denominato vincente, il dato viene assegnato al cluster in esame ed il neurone vincente viene aggiornato in maniera robusta. Qualora non fosse presente nessun nodo abbastanza simile al dato in esame, la rete assume un comportamento plastico, creando un nuovo neurone in cui la nuova informazione viene inserita totalmente. In questo algoritmo i dati vengono sottoposti in maniera sequenziale e ciclica alla rete ed il procedimento termina quando non vi sono più
apprezzabili mutamenti nei pesi delle connessioni neurali.

Carpenter e Grosberg proposero un classificatore in grado di clusterizzare immagini binarie denominato ART1. Questo strumento, largamente impiegato nel campo del riconoscimento di caratteri, è stato recentemente proposto per la catalogazione delle mappe EWS binarie [26]. L’algoritmo ART1 utilizza solo grandezze binarie, sia come dato d’ingresso che come peso delle connessioni neurali. La misura di similitudine impiegata si basa sulle due seguenti grandezze:

\[
S_1(x, p) = \frac{p' \cdot x}{\beta + ||p||}
\]

\[
S_2(x, p) = \frac{p' \cdot x}{||x||}
\]

dove \( \beta \) è un numero piccolo a piacere, \( x \) un generico ingresso, \( p \) un pattern e la notazione \( ||z|| \) indica la somma del vettore binario \( z \). La grandezza \( S_1(x, p) \) indica la percentuale di valori alti del pattern \( p \) che sono presenti nel dato \( x \); mentre \( S_2(x, p) \) indica la percentuale di valori alti nel dato \( x \) presenti nel pattern \( p \). Fra le classi che presentano un valore di \( S_2 \) superiore ad una certa soglia \( \rho \), chiamata vigilanza, viene scelto come cluster più simile quello che presenta il maggior valore di \( S_1 \). Il test così descritto viene chiamato test di vigilanza ed è esprimibile mediante la:

\[
c^* = \arg\min_c S_2(x, p^c) \text{ soggetto a } S_1(x^w, p^c) \geq \rho \quad (3.1)
\]

Se il problema di minimo vincolato 3.1 non ha soluzioni il dato \( x \) viene considerato troppo dissimile dai pattern presenti e viene creato un nuovo pattern inizializzato ad \( x \). Diversamente, se esiste un cluster \( c^* \) in grado di soddisfare il test di vigilanza, il suo pattern viene modificato in maniera da renderlo più simile al vettore \( x \). L’aggiornamento avviene rimuovendo da \( p^{c^*} \) tutti i valori alti non presenti in \( x \). Quest’operazione viene di fatto realizzata applicando l’operatore logico and ai due vettori.

L’apprendimento termina quando dopo aver visionato tutto il set di dati non vi è stata alcuna modifica nei pesi dei neuroni.

La procedura descritta viene realizzata dal seguente algoritmo:

1. Si inizializza il parametro di vigilanza \( \rho \), il numero dei neuroni \( N_c \) ad 1 e si pone \( p^1 \) pari ad un vettore di un.
2. Pongo $A_0 = p^e$.

3. Per ogni mappa $x^w$.

   (a) Eseguo il test di vigilanza (3.1)
   (b) Se esiste un neurone $c^*$ che soddisfa la (3.1)
      i. Modifico il neurone più simile ponendo $p^{c^*} = p^{c^*} \land x^w$
      ii. Pongo $g(w) = c^*$
   Altrimenti
      i. Pongo $N_c = N_c + 1$ e
      ii. Creo il nuovo neurone $p^{N_c} = x^w$
      iii. Pongo $g(w) = c^*$

4. Raccolgo tutti i vettori $p^e$ in una matrice $A$ di dimensione $N_d \times N_c$.

5. Se $A_0 = A$ Fine

6. Pongo $A_0 = A$

7. Torno al Punto 3

### 3.3.2 Mappe di Kohonen

Nell’algoritmo ART1 si è visto come tutta la conoscenza fornita da un dato in ingresso sia appannaggo di un solo neurone. In talune applicazioni può essere produttente fornire l’informazione proveniente da un dato non solo al neurone vincente, ma anche ad un gruppo selezionato di altri neuroni. Ad esempio, qualora si sospettasse la presenza di gruppi di cluster fra loro vicini, l’attirare più neuroni in queste zone ad alta densità potrebbe fornire diversi vantaggi sia in termini di tempo di calcolo che di qualità del risultato raggiunto. Questo secondo modo di operare gli aggiornamenti dei pesi neuronali è proprio delle reti definite soft competitive, fra cui si annoverano le mappe di Kohonen o Self Organizing Maps (SOM) [27].

L’apprendimento in una rete SOM avviene elaborando un dato alla volta. Nell’analisi di ogni dato si possono osservare tre fasi distinte:
CAPITOLO 3. LA CLASSIFICAZIONE DELLE MAPPE BINARIE

Figura 3.6: Griglia bidimensionale di una mappa di Kohonen

**Competizione.** Per ogni input, i neuroni competono tra loro per essere attivati, col risultato che un solo neurone risulta vincitore;

**Cooperazione.** Ogni neurone vincitore determina, mediante una funzione di vicinato, quali dei neuroni verranno modificati.

**Adattamento.** Sia il neurone vincitore che i suoi vicini subiscono un incremento dei propri pesi sinaptici.

Fissato un dato $x$ la fase di competizione si risolve individuando il vincitore nel neurone che presenta il pattern più simile al dato in ingresso. Il criterio utilizzato e’ solitamente quello della distanza euclidea come descritto dalla

$$ c^* = \arg \min_c \|x - p^c\| $$

Nella fase di cooperazione risiede la particolarità dell’algoritmo. In questa rete i neuroni sono dotati di una dimensionalità spaziale: essi vengono infatti disposti lungo un reticolo bidimensionale $N_R \times N_C$ come illustrato in Figura 3.6 [28].

Questa disposizione, fissa ed inamovibile, consente l’uso di semplici criteri geometrici come funzione di vicinato $\delta(c, c^*)$. Una scelta largamente utilizzata è la norma $L_1$ o distanza di Manhattan [29], ovvero il numero minimo di archi che è necessario percorrere sul reticolo per passare dal neurone $c$ al neurone $c^*$. 
3.3. RETI NEURALI

Nella fase di adattamento i neuroni cooperanti vengono aggiornati per essere più simili al dato in esame. La forza dell’aggiornamento è regolata da due fattori: uno costante per tutti i neuroni chiamato velocità di apprendimento ($\varepsilon$), l’altro legato alla distanza del neurone sulla griglia. L’aggiornamento sarà più sensibile nei neuroni più vicini al nodo vincitore e la percentuale di conoscenza elargita viene regolata da una funzione $h$. La stabilità della rete viene assicurata mediante una riduzione graduale del guadagno della rete di update come mostrato dalla funzione di aggiornamento del generico ciclo $t$:

$$p^c = p^c + \varepsilon(t) \cdot h(\delta(c, c^*), t) \cdot (x^w - p^c)$$

Le diverse realizzazione di un’architettura SOM sono quindi definite dalle funzioni $\varepsilon(\cdot)$ e $h(\cdot, \cdot)$.

In [29] viene proposta la seguente velocità di apprendimento:

$$\varepsilon(t) = \varepsilon_i \left(\frac{\varepsilon_f}{\varepsilon_i}\right)^{\frac{t}{t_{\text{max}}}}$$

(3.2)

dove $\varepsilon_f$ ed $\varepsilon_i$ sono delle costanti e $t_{\text{max}}$ il massimo numero di iterazioni. L’impiego della (3.2) diminuisce di molto la sensibilità della rete alle condizioni iniziali.

In [30] viene proposta la seguente funzione $h(t)$:

$$h(t, d) = e^{-\sigma^2_{(t)}}, \quad \sigma(t) = \sigma_i \left(\frac{\sigma_f}{\sigma_i}\right)^{\frac{t}{t_{\text{max}}}}$$

dove $\sigma_f$ e $\sigma_i$ sono appropriate costanti.

Siamo ora in grado di definire l’algoritmo per eseguire la classificazione delle mappe EWS:

1. Si inizializza la griglia $A$ di $N_c = N_R \times N_C$ elementi con i pesi $p^c$, assegnati ai nodi della griglia, e si pone $t = 0$.

2. Per ogni vettore di ingresso $x^w$

   (a) Si determina il neurone vincente $c^*$ dove

   $$c^* = \arg \min_c \| x^w - p^c \|$$
CAPITOLO 3. LA CLASSIFICAZIONE DELLE MAPPE BINARIE

(b) Per ogni neurone $c$

i. Si calcola la distanza $d$ dal neurone $c^*$ secondo la norma $L_1$

ii. Si pone $g(w) = c^*$

iii. Si aggiornano i pesi delle connessioni secondo la seguente formula

$$p^c = p^c + \varepsilon(t) \cdot e^{-\frac{d^2}{\sigma(t)}} (x^w - p^c)$$

(c) Se $t = t_{max}$, Stop

(d) Si pone $t = t + 1$, $\varepsilon(t) = \varepsilon_i \left(\frac{\varepsilon_f}{\varepsilon_i}\right)^{\frac{1}{t_{max}}}$, $\sigma(t) = \sigma_i \left(\frac{\sigma_f}{\sigma_i}\right)^{\frac{1}{t_{max}}}.$

3. Torno al punto 2.

3.3.3 Neural Gas

L’algoritmo Neural Gas (Martinetz e Schulten, 1991), anch’esso basato sul soft competitive learning, presenta forti analogie con il metodo ideato da Kohonen. Come avviene nelle reti SOM, anche questa rete neurale prevede la presenza di un predeterminato numero di neuroni la cui capacità di apprendimento diminuisce nel tempo. Inoltre le due tipologie di reti presentano le stesse fasi di competizione e adattamento. Infatti, sia nel metodo di Kohonen che nell’algoritmo Neural Gas, il neurone vincitore $c^*$ risulta essere quello più simile all’ingresso in esame e l’aggiornamento dei neuroni cooperanti è legato sia alla velocità di apprendimento $\varepsilon$ sia al grado di cooperazione.

Nonostante queste analogie, sono presenti due grandi differenze rispetto all’algoritmo SOM:

- Non esiste una topologia prefissata della rete. I neuroni sono considerati come entità fortemente indipendenti ma libere di aggregarsi in risposta agli stimoli esterni, proprio come avviene nelle molecole di un gas.

3.4. Mistura di Bernoulliane

Per tradurre formalmente questo metodo di aggiornamento dei pesi, Martinetz [31] introduce una funzione \( k_c(x, A) \), dipendente dall’ingresso in esame \( x \), dall’insieme dei pattern \( A \) e dal singolo pattern di riferimento \( c \). Questa funzione restituisce la posizione del valore \( c \) all’interno nella sequenza di indici \( (i_0, i_1, ..., i_{N_c-1}) \) ordinata in maniera decrescente per similitudine dei patterns con il dato \( x \). La funzione \( k_c(x, A) \) fornisce quindi una sorta di distanza fra il neurone vincente, che per definizione occuperà la posizione \( i_0 \), ed il generico neurone \( c \).

Introdotta questa distanza anche per questa rete neurale le diverse implementazioni si differenziano in base alla scelta delle funzioni \( \varepsilon(t) \) e \( h(t, k_c(x, A)) \).

Il classificatore utilizzato per il clustering delle mappe EWS binarie effettua le stesse scelte operate nel precedente paragrafo [29] e viene descritto dal seguente algoritmo.

1. Si inizializza un set \( A \) di \( N_c \) neuroni aventi pesi \( p^c \) e si pone \( t = 0 \).

2. Per ogni vettore di ingresso \( x^w \),
   
   (a) Si determina il neurone vincente \( c^* \)
   
   (b) Si pone \( g(w) = c^* \)
   
   (c) Per ogni neurone \( c \) si aggiornano i pesi delle connessioni secondo la seguente formula
   \[
   p^c = p^c + \varepsilon(t) \cdot e^{-\frac{k_c(x, A)}{\sigma(t)}} (x^w - p^c).
   \]
   
   (d) Se \( t = t_{max} \), Stop
   
   (e) Si pone \( t = t + 1 \), \( \varepsilon(t) = \varepsilon_i \left( \frac{\varepsilon_f}{\varepsilon_i} \right)^{t_{max}} \), \( \sigma(t) = \sigma_i \left( \frac{\sigma_f}{\sigma_i} \right)^{t_{max}} \)

3. Torno al punto 2

3.4 Mistura di Bernoulliane

Un classificatore basato su di un modello stima i parametri descrittivi del modello in modo da ottenere una classificazione che massimizzi l’aderenza del modello ai dati di partenza. Un algoritmo di questo tipo sarà definito da un modello, una misura di aderenza ed un metodo di massimizzazione.
3.4.1 Il modello probabilistico

In una mappa binaria l’esito del test elettrico effettuato sul singolo die può assumere solo due valori: funzionante o guasto. Questo evento per un determinato dispositivo è stato supposto essere indipendente dall’esito degli altri test e di tipo bernoulliano. Ogni wafer può essere caratterizzato da una firma d’errore. Pertanto la probabilità di guasto di un dispositivo viene legata alla posizione spaziale del die sul wafer mediante un funzione di densità spaziale $f$:

$$P(\text{fallimento}) = f(x, y)$$

dove $x$ e $y$ sono le coordinate planari del centro del dispositivo. Volendo mantenere un aspetto generale nel modello si considera il wafer come una superficie circolare piana di raggio unitario al cui centro giace l’origine degli assi cartesiani, $x$ e $y$ sono le coordinate planari del centro del dispositivo. Così facendo si svincola la definizione della funzione di probabilità dalle dimensioni fisiche del wafer. Se un gruppo di fette presenta un pattern, esso può essere associato ad un’opportuna funzione $f$. In caso di difettosità uniforme la probabilità di fallimento sarà costante ad un valore legato al rendimento atteso.

In questo modello, la singola mappa $EWS$ binaria $x = [x(1) \ldots x(N_d)]'$ viene visto come l’esito di $N_d$ prove di Bernoulli. Le probabilità sono indicative del tipo di firma presente nel wafer e vengono raccolte nel vettore $p = [p(1) \ldots p(N_d)]'$ dove

$$p(d) := P(x(d) = 1)$$

con $1 \leq d \leq N_d$. Ad ogni pattern $c$ corrisponde, quindi, un vettore di probabilità di guasto $p^c \in [0, 1]^{N_d}$.

In una produzione possono essere presenti diversi patterns ognuno riscontrabile in un diverso volume di wafer. La probabilità che una fetta estratta a caso presenti una determinata firma d’errore $c$ delle $N$ presenti, viene indicata con $\pi(c)$. Queste probabilità sono considerate indipendenti e vengono raggruppate nel vettore:

$$\pi = [\pi(1) \pi(2) \ldots \pi(N)]$$

dove $\pi(c) \in [0; 1]$ e $\sum_{c=1}^{N} \pi(c) = 1$.

Nonostante l’apparente semplicità del modello, si è trovato che con una scelta
appropriata delle funzione di probabilità $f(x, y)$ si ottengono patterns simulati molto simili a quelli osservati nella realtà [32].

3.4.2 La misura di aderenza

Determinato un modello, quindi i vettori $p^c$ e $\pi$, è possibile utilizzare una misura di aderenza per effettuare la classificazione. La misura di aderenza adottata in questa sede è la verosimiglianza.

Dato un insieme di dati in ingresso $X$ ed un modello descritto da un insieme di parametri $\theta$, la verosimiglianza, indicata con $L(X, \theta)$, è definita come la probabilità che i dati $X$ siano generati dal modello associato a $\theta$:

$$L(X, \theta) = P(X|\theta).$$ (3.3)

Avere un’elevata verosimiglianza è un indice della bontà della classificazione. Spesso per motivi pratici si preferisce utilizzare, come misura di aderenza la log-verosimiglianza $l(X, \theta) := \ln(L(X, \theta))$

3.4.3 L’algoritmo di massimizzazione

L’algoritmo Expectation-Maximization \textit{EM} si pone come obiettivo la massimizzazione della verosimiglianza [33]. Lo sviluppo del metodo nasce dalla difficoltà di massimizzare la 3.3 in modo diretto.

In [34] viene proposto un approccio alternativo alla massimizzazione diretta basato sull’introduzione di un set di variabili ausiliarie, o \textit{competenze} $Z$. In molti testi l’insieme dei dati $X$ viene denominato insieme dei dati incompleto, mentre l’insieme di dati $T := \{X, Z\}$ viene detto insieme dei dati completo. La verosimiglianza, avendo a disposizione i dati completi, diviene

$$P(T|\theta) = P(X, Z|\theta) = P(X|Z, \theta)P(Z|\theta)$$ (3.4)

L’algoritmo \textit{EM} sfrutta l’introduzione dei dati ausiliari per impostare un metodo iterativo in grado di massimizzare la (3.4). Questo metodo si basa su due passi fondamentali detti Expectation e Maximization.
**CAPITOLO 3. LA CLASSIFICAZIONE DELLE MAPPE BINARIE**

**Expectation** si calcola il valore atteso della 3.4 rispetto all’insieme \( Z \) noti \( X \) e \( \theta \) (al passo attuale \( t \))

\[
l_T(\theta, \theta(t)) = E[\ln(P(T|\theta))|X, \theta(t)]
\]

**Maximization** si calcola il nuovo valore di \( \theta(t + 1) \) che massimizza \( l_T(\theta, \theta(t)) \):

\[
\theta(t + 1) = \arg \max_\theta l_T(\theta, \theta(t))
\]

Questi due passi vengono eseguiti iterativamente, fino a non avere più spostamenti nella classificazione degli ingressi da una classe all’altra, oppure fino a quando si ottengono variazioni trascurabili del valore della log-verosimiglianza tra un’iterazione e la successiva [35].

Punto chiave del metodo è che la funzione che si va a massimizzare, grazie all’introduzione dei dati \( Z \), sia più semplice di quella calcolata con i soli dati incompleti. Dempster [34] ha dimostrato che, sotto opportune condizioni legate alla natura delle variabili \( Z \), ad ogni iterazione dell’algoritmo la verosimiglianza è maggiore di quella associata ai parametri calcolati nel ciclo precedente.

Nell’applicare la metodologia al problema in esame si sono definite come dati ausiliari delle variabili bernoulliane \( r_w(c) \). Queste grandezze sono poste ad uno se il wafer \( w \) appartiene alla classe \( c \) mentre assumono il valore nullo nel caso opposto. L’insieme dei dati ausiliari \( Z \) sarà quindi composto da \( N_w \times N_c \) elementi. L’algoritmo EM stima le competenze

\[
\bar{r}_w(c) := E[r_w(c)|X, \theta]
\]

ed assegna il wafer \( w \) alla classe che dimostra massima competenza:

\[
g(w) = \arg \max_c \bar{r}_w(c)
\]

La scelta di queste variabili ausiliari ha consentito di ricavare, come mostrato nell’Appendice 7.1, il seguente algoritmo

1. Si considera un set \( A \) di \( N_c \) patterns \( p^c \).

2. Per ogni mappa \( w \),

   (a) si calcola la verosimiglianza con tutti i patter
(b) indicando con $c^*$ il pattern a verosimiglianza maggiore, si definiscono le competenze:

$$r^w(c^*) = 1, \quad r^w(c) = 0, \forall c \neq c^*$$

3. Si inizializza il vettore $\pi$:

$$\pi(c) = \frac{\sum_{w=1}^{N_w} r^w(c)}{N_w}$$

4. Si aggiornano le variabili ausiliare mediante il passo di Expectation

$$\overline{r^w(c)} = \frac{\pi(c) \prod_{d=1}^{N_d} (x^w(d) \cdot p^c(d) + (1 - x^w(d))(1 - p^c(d)))}{\sum_{c=1}^{N_c} \pi(c) \prod_{d=1}^{N_d} (x^w(d) \cdot p^c(d) + (1 - x^w(d))(1 - p^c(d)))}$$

5. Si aggiornano i parametri del modello mediante il passo di Maximization

$$p^c(d) = \frac{\sum_{w=1}^{N_w} r^w(c) x^w(d)}{\sum_{w=1}^{N_w} r^w(c)}$$

$$\pi(c) = \frac{\sum_{w=1}^{N_w} r^w(c)}{N_w}$$

6. Si torna al Passo 4 finché non si rilevano variazioni significative della verosimiglianza

7. Si estrapola la classificazione ponendo per ogni wafer $w$

$$g(w) = \arg \max_c r^w(c)$$
CAPITOLO 3. LA CLASSIFICAZIONE DELLE MAPPE BINARIE
Capitolo 4

Confronto dei classificatori per mappe binarie

I metodi proposti in precedenza differiscono per vari fattori quali ad esempio il numero dei parametri liberi, la struttura imposta ai cluster identificati e le caratteristiche di stabilità. Ciononostante tutti possono essere utilizzati per isolare e descrivere le differenti firme d’errore presenti nelle mappe EWS binarie.

In questo capitolo ci proponiamo di individuare quale fra i classificatori presentati nel precedente capitolo, sia il più idoneo ad essere implementato nella procedura proposta in questa sede; l’efficacia della classificazione ai fini di una corretta diagnosi di processo rappresenterà il fattore dominante nella scelta del miglior metodo.
Tabella 4.1: Tabella di sparsità relativa ad una classificazione in cinque classi di 150 wafer marcati con quattro differenti firme d’errore.

4.1 La valutazione di una classificazione

Lo scopo della classificazione analizzato in questa sede è di raccogliere nello stesso cluster tutte e sole le mappe che presentino la stessa firma d’errore. Un cluster che soddisfi questo requisito viene detto true cluster o classe vera, parimenti una classificazione composta da sole classi vere sarà definita true classification o classificazione vera.

Supposto di conoscere quale sia la classificazione vera, nell’eseguire una partizione possono verificarsi due diverse categorie d’errore: mappe con la stessa firma d’errore vengono catalogate in due diverse classi identificate, errore di non univocità, oppure lo stesso cluster identificato potrebbe contenere wafers con diversi patterns, errore di non omogeneità. Le due tipologie hanno un impatto decisamente diverso nel proseguo della procedura di diagnosi proposta.

Si consideri, ad esempio, la tabella di sparsità [36], riportata in Tabella 4.1, e relativa ad una classificazione di 150 wafer marcati con quattro differenti firme d’errore.

Il secondo cluster corrisponde ad un cluster vero in quanto contiene esclusivamente la totalità della mappe EWS caratterizzate dalla prima firma d’errore.

Il primo ed il quinto cluster evidenziano un fenomeno di non univocità in quanto formati solo da mappe che presentano la stessa firma d’errore e quindi lo stesso difetto fisico; tali cluster vengono per tanto denominati omogenei. L’analisi di processo del primo cluster, rispetto a quella ottenibile utilizzando la classificazione vera, risente solo della mancanza di alcune mappe. Inoltre il pattern di riferimento della classe identificata, con buona probabilità, risulterà simile a quello definito dalla classificazione vera in quanto composto da wafer che presentano la stessa firma d’errore.
4.1. LA VALUTAZIONE DI UNA CLASSIFICAZIONE

Gli errori di non uivocità di fatto si traducono in uno split delle classi vere, come mostrato dai cluster 1 e 5 di Tabella 4.1. Per correggere un errore di questo tipo sarà sufficiente rendersi conto, in fase di scelta del guasto da analizzare, che le due classi, presentando pattern visualmente simili, descrivono in realtà lo stesso difetto fisico.

Le classi 3 e 4 di Tabella 4.1 presentano errori di non omogeneità in quanto formate da mappe EWS con diverse firme d’errore e quindi relative a wafer affetti da differenti difetti fisici. Classi di questo tipo vengono dette non omogenee. Effettuare la diagnosi relativa a firme d’errore definite mediante classi non omogenee, può essere arduo. Viene infatti richiesto di trovare le lavorazioni responsabili di diversi problemi fisici e non di uno solo; perdendo così uno dei principali vantaggi della procedura proposta. Inoltre la mappa media di un cluster non omogeneo risulta falsata in quanto rappresenta una composizione delle diverse firme d’errore realmente presenti nel cluster. Ad esempio il pattern di referimento delle classi 3 e 4 risentirà della presenza rispettivamente di 5 e 10 wafer affetti da un diverso difetto fisico.

Mentre la prima tipologia d’errore può essere corretta mediante un operazione di accorpamento di classi, la seconda presenta problematiche insolubili nella procedura proposta. Per tanto un buon classificatore dovrà produrre clusters il più possibile omogenei.

4.1.1 Misura di omogenità di un cluster

Per definizione, in una classe non omogenea sono presenti wafer appartenenti a diverse classi vere. Cononostante, per ogni cluster identificato $j$, è possibile definire almeno una classe dominante $c_d(j)$ come la classe reale dalla quale proviene il maggior numero dei wafers della classe $j$. Indicando con $G_{i,j}$ il generico elemento della tabella di sparsità la classe dominate è data da:

$$c_d(j) = \arg \max_i G_{i,j}$$

Nota la classe dominante, una buona misura della disomogeneità del cluster è data dal rapporto del numero di wafers che risultano misclassificati, ovvero quei wafer non assegnati alla classe dominante, rispetto al totale dei wafer presenti nel cluster.
Per la $j$-esima classe identificata si ha che:

$$M(j) = \frac{\sum_i G_{i,j} - G_{cd(j),k}}{\sum_i G_{i,j}} \quad (4.1)$$

Si noti che qualora ci fossero più classi dominanti la scelta di una di esse non influirebbe nel computo della 4.1.

Poiché per ogni cluster esiste almeno un wafer proveniente dalla classe dominante, $M(j)$ può assumere valori compresi tra 0 e $K$, con $K < 1$. Un valore nullo di $M(j)$ indica che la totalità delle mappe appartiene alla classe dominante, quindi che il cluster è omogeneo. Il massimo valore di $M(j)$ si ha per una distribuzione uniforme delle mappe fra le $N_c$ classi reali. In questo caso ogni classe vera è anche classe dominante per il cluster considerato e possiede $\frac{\sum_i G_{i,j}}{N_c}$ all’interno della classe. In quest’ultimo caso, il valore massimo dell’indice sarà:

$$K = \frac{(N_c - 1)\sum_i G_{i,j}}{\sum_i G_{i,j}} = \frac{(N_c - 1)}{N_c}$$

Volendo ottenere un indice che assuma valori tra 0 e 1, sarà sufficiente moltiplicare $M(j)$ per il fattore $K^{-1}$. Si ottiene, così l’indice normalizzato:

$$M_N(j) = \frac{N_c}{(N_c - 1)}M(j) = \frac{N_c(\sum_i G_{i,j} - G_{cd(j),k})}{(N_c - 1)\sum_i G_{i,j}}$$

che valuta l’omogeneità della $j$-esima classe identificata.

### 4.1.2 Misura di omogeneità di una classificazione

In una classificazione sono presenti $N_c$ classi identificate. Per avere una valutazione numerica del grado di omogeneità dei diversi cluster identificati si definisce l’indice $F$ come la media di $M_N(j)$ pesata per il numero totale dei wafer della classe di appartenenza $\sum_i G_{i,j}$:

$$F = \frac{\sum_j (\sum_i G_{i,j} M_N(j))}{\sum_j \sum_i G_{i,j}} = \frac{N_c \sum_j (\sum_i G_{i,j} - G_{cd(j),k})}{N_w(N_c - 1)}$$

La misura proposta varia fra il valore nullo, assegnato ad una classificazione a cluster omogenei e l’unità che corrisponde ad una distribuzione uniforme delle firme d’errore all’interno di ogni classe identificata.
4.2. IL CONFRONTO PROPOSTO

Il confronto proposto in questa sede utilizzerà l’indice $F$ come criterio di valutazione nella scelta del metodo migliore.

Nell’Appendice 7.3 viene riportato un confronto fra i principali indici proposti in letteratura e l’indice $F$. I risultati di quest’analisi saranno utili qualora si volesse comunque valutare una classificazione non avendo a disposizione la conoscenza della classificazione vera.

4.2 Il confronto proposto

L’indice $F$, per stabilire il grado di omogeneità della partizione considerata, abbisogna della conoscenza di una classificazione vera.

In un processo produttivo complesso come quello in esame, discernere se un wafer sia affetto da un solo e ben determinato problema risulta spesso proibitivo, anche quando si sia riusciti ad ovviare al calo di resa. Pertanto nel confronto proposto non risulta possibile utilizzare dati reali. Si è deciso quindi di basare la scelta del miglior metodo sull’analisi di dati simulati da impiegare come banco di prova o benchmark.

4.2.1 Il Benchmark

I dati simulati sono stati ottenuti utilizzando il modello proposto nel Paragrafo 3.4.1. Sono state individuate alcune funzioni di probabilità utili a descrivere alcune firme d’errore comunemente riscontrate nella fabbricazione di semiconduttori.

**Uniforme** $p_U(c) = c$. Il pattern è caratterizzato da una distribuzione omogenea della probabilità di fallimento.

**Bolla:** $p_B(x_c, y_c, \sigma) = \exp\left(\frac{(x-x_c)^2+(y-y_c)^2}{2\sigma^2}\right)$. La bolla è centrata in $(x_c, y_c)$ e la sua larghezza viene controllata dal parametro $\sigma$.

**Anello:** $p_A(x_c, y_c, \sigma) = 1 - p_B(x_c, y_c, \sigma)$. L’anello è centrato in $(x_c, y_c)$ e la sua larghezza è controllata dal parametro $\sigma$; mediante una scelta opportuna delle coordinate del centro è possibile ottenere delle mezze lune traslando l’anello.

**Ripetizioni a righe:** $p_R(T, \phi) = \frac{1+\sin(2\pi y/T+\phi)}{2}$. Il pattern è caratterizzato da più
CAPITOLO 4. CONFRONTO DEI CLASSIFICATORI PER MAPPE BINARIE

Figura 4.1: Funzioni di probabilità utilizzate nei dati simulati

Ripetizioni di righe di dispositivi guasti. I parametri che ne controllano il periodo e l’allineamento sono rispettivamente $T$ e $\phi$.

**Ripetizioni a colonne:** $p_C(T, \phi) = \frac{1+\sin(2\pi x/T+\phi)}{2}$. Per questa distribuzione valgono le stesse considerazioni fatte per la precedente.

Firme più complesse possono essere ottenute combinando le precedenti funzioni di base con i seguenti operatori.

**Combinazioni di pattern di tipo OR:** $p^1(x, y) \lor p^2(x, y) = p^1(x, y) + p^2(x, y) - p^1(x, y)p^2(x, y)$. La tipologia del difetto corrisponde all’OR logico delle tipologie di tipo le cui probabilità corrispondenti sono $p^1(x, y)$ e $p^2(x, y)$.

**Combinazioni di pattern di tipo AND:** $p^1(x, y) \land p^2(x, y) = p^1(x, y)p^2(x, y)$. La tipologia del difetto corrisponde all’AND logico delle tipologie di tipo 1 e 2 le cui probabilità corrispondenti sono $p_1(x, y)$ e $p_2(x, y)$.

Di concerto con ST Microelectronics sono state stabilite delle funzioni di probabilità atte a ricreare firme d’errore riscontrabili in lotti di produzione. Le otto tipologie di difetto che si è deciso di trattare vengono riportate in Figura 4.1 e sono caratterizzate dalle seguenti distribuzioni di probabilità e numerosità di wafer:

1. $p^1 = p_U(.7)$: bassa resa (4% dei wafer).
4.2. **IL CONFRONTO PROPOSTO**

La metodologia di test

Il confronto fra i vari metodi viene fatto scegliendo i migliori risultati forniti dai singoli metodi al variare dei loro parametri. I parametri da tarare sono riassunti in Tabella 4.3. Ad eccezione del metodo ART1, tutti gli altri algoritmi possiedono tra i propri parametri il vettore di inizializzazione $A$ ed il numero di classi $N_c$. (Si ricorda che per il metodo SOM $N_c = N_R \times N_C$). Si è deciso, quindi, di considerare questi due aspetti in maniera generale e separata dagli altri parametri. Questa scelta è motivata dal fatto che entrambi i parametri sono spesso considerati in maniera indipendente dagli altri [21], [37].

Sebbene in un problema di clustering determinare il corretto numero di classi sia un compito considerato complesso [38], in questa particolare applicazione assume
una importanza relativa. Infatti, per i nostri scopi siamo interessati, più che a determinare il reale numero di clusters, ad ottenere classi omogenee (eventualmente in numero maggiore rispetto al numero di classi reali). A fronte di queste considerazioni si è deciso di fissare il numero di classi da identificare. Nel nostro caso si è posto \( N_c = 12 \), ovvero una volta e mezzo il numero di classi reali utilizzate nel benchmark (8).

L’inizializzazione dei pattern rappresenta il secondo parametro comune a molti dei metodi in esame. Si noti che le prestazioni di alcuni algoritmi sono fortemente legate alla tecnica impiegata per inizializzare l’algoritmo. Pertanto si è deciso di effettuare la taratura dei parametri in due fasi: la prima dedicata alla taratura dei parametri propri del metodo, la seconda atta a determinare il miglior metodo di inizializzazione.

### 4.3 Taratura automatica dei parametri

Gli algoritmi che presentano dei parametri propri sono tre: l’algoritmo ART1, le mappe di Kohonen ed il Neural Gas.

#### 4.3.1 L’algoritmo ART1

Il metodo ART1 determina in maniera automatica il numero di classi, una volta impostato il parametro di vigilanza \( \rho \) ed il parametro \( \beta \). In [17] viene proposta una tecnica di taratura dei parametri per il metodo utilizzata nel riconoscimento di
4.3. TARATURA AUTOMATICA DEI PARAMETRI

Figura 4.2: Classi identificate dal metodo ART1 al variare della verosimiglianza $\rho$

pattern in mappe EWS. La tecnica proposta assegna $\beta = 0.0005$ e testa diversi valori di vigilanza finché i risultati della classificazione non sono soddisfacenti. Come prima prova si è pensato di utilizzare la taratura proposta.

In Figura 4.2 vengono riportati gli esiti della taratura considerata applicati ai set 2 e 11. Utilizzando diverse prove, qui non riportate, si è visto che, anche con piccoli valori di vigilanza quale 0.02, il numero di classi identificate non è inferiore a 60. Inoltre le classi non sono mai omogenee.

Confrontando i risultati riportati in questa sede con quelli descritti in [17] sembra che il metodo abbia delle difficoltà a gestire un grande numero di wafer. Questo può essere imputabile al fatto che la rete identifica i pattern di riferimento mediante una logica "AND". Infatti per appartenere allo stesso cluster le mappe debbono presentare almeno una percentuale di dispositivi falliti nella stessa posizione. Requisito riscontrabile nell’analisi di poche mappe EWS, ma improponibile in quella di un numero dati considerevoli come quelli richiesti per svolgere una diagnosi di processo.

Pertanto si è deciso di eliminare il metodo dal confronto.

4.3.2 L’algoritmo SOM

L’algoritmo SOM possiede cinque parametri propri: il learning rate iniziale e quello finale ($\varepsilon_i, \varepsilon_f$), la larghezza iniziale e finale della funzione di vicinato ($\sigma_i, \sigma_f$) ed il
numero di iterazioni ($t_{\text{max}}$). Non esiste un criterio per la scelta dei parametri che abbia validità generale. Nel nostro caso si è deciso di sfruttare una procedura di taratura empirica per la taratura dei parametri delle reti SOM che ha ottenuto buoni risultati nella classificazione di dati reali [39].

La procedura proposta divide i parametri in base alle funzioni che essi svolgono nella rete e tara separatamente i vari gruppi di parametri. I parametri $\varepsilon_i$, $\varepsilon_f$, $\sigma_i$, $\sigma_f$ influenzano la velocità di convergenza della rete, mentre il parametro $t_{\text{max}}$ si riferisce al tempo di apprendimento. Il primo gruppo di parametri viene fissato a valori costanti utilizzando la taratura proposta da [29]. Una volta fissata la velocità di convergenza con dei valori noti, si può automaticamente trovare il tempo di apprendimento relazionando tale velocità di apprendimento con la dimensione del set di dati seguendo la relazione $t_{\text{max}} = LN_w$. La costante $L$ viene scelta osservando le prestazioni del classificatore.

In questa sede abbiamo deciso di adottare la procedura precedentemente descritta su parte del benchmark, cambiando però il criterio di valutazione. Fissato un set di dati, ed un numero di cicli $L$ vengono effettuate 20 differenti classificazioni ognuna originata da un diversa inizializzazione e le si valuta mediante il relativo indice $F$. Come indicatore di bontà della classificazione ottenuta con i parametri adottati si utilizza la mediana degli indici $F$ così calcolati.

In Figura 4.3 vengono riportati i risultati relativi ad alcuni benchmark testati.
4.3. TARATURA AUTOMATICA DEI PARAMETRI

Figura 4.4: Taratura dei parametri per l'algoritmo Neural Gas

Si può notare che dopo 80 cicli, vi è una stabilizzazione delle prestazioni sui set considerati.

4.3.3 Neural Gas

L'algoritmo Neural Gas necessita, come descritto per il metodo di Kohonen, di una particolare attenzione nei confronti della taratura dei parametri che lo caratterizzano. Si è deciso di seguire lo stesso approccio usato per l'algoritmo SOM. Vengono fissati i valori dei quattro parametri riguardanti la velocità di convergenza della rete in modo da trovare esclusivamente il numero di cicli necessari per rendere stabili e soddisfacenti le prestazioni.

Per fissare i parametri relativi alla velocità di convergenza si è impiegata la taratura proposta in [31] che fissa i seguenti valori: $\varepsilon_i = 0.5$, $\varepsilon_f = 0.005$, $\sigma_i = 10$, $\sigma_f = 0.001$. Così come accaduto per l'algoritmo SOM, il tempo di simulazione viene ricercato come multiplo della dimensione dello spazio dei dati $t_{max} = LN_w$.

I risultati ottenuti, illustrati in Figura 4.4, mostrano che un numero di cicli pari a $L = 50$ è da ritenersi soddisfacente. Dopo cinquanta cicli, infatti, per le diverse tipologie di benchmark prese in esame, vi è una stabilizzazione dei valori dell’indice $F$. 
4.4 La Scelta del miglior punto di inizializzazione

Eliminata la tecnica ART1 dal confronto e determinata la taratura dei parametri propri dei restanti metodi, il punto di inizializzazione rappresenta l’ultimo grado di libertà di scelta non ancora fissato. La valutazione dei metodi di classificazione per mappe binarie si riduce quindi al confronto fra i risultati ottenuti dalla miglior tecnica di inizializzazione.

4.4.1 Tecniche di inizializzazione

Il compito di una tecnica di inizializzazione quello è di distribuire $N_c$ centroidi $p^c$ in uno spazio $N_d$-dimensionale. Poiché ogni elemento di $p^c$ può assumere solo valori all’interno dell’intervallo $[0, 1]$, i possibili punti di inizializzazione sono tutti contenuti all’interno di un ipercubo a spigolo unitario. Le tecniche di inizializzazione considerate sono le seguenti [37].

**Random** Si distribuiscono i centroidi in maniera casuale, estraendo $N_c$ punti dall’ipercubo. Molti metodi mal sopportano l’inizializzazione in punti estremi dell’ipercubo. Onde evitare questo tipo di inizializzazione, si è limitato lo spazio dei valori validi ad un ipercubo di spigolo lievemente ridotto. I punti di inizializzazione sono quindi dati dalla seguente:

$$p_{\text{rand}}^c = \text{rand}\{x \in [\epsilon, 1 - \epsilon]^{N_d}\}$$

dove $\epsilon$ è una costante positiva introdotta per escludere i valori estremi di probabilità. Nel presente lavoro si è imposto $\epsilon = 0.005$. Lo svantaggio più rilevante di questo tipo di inizializzazione è che si ottengono pattern che difficilmente rispecchieranno figure ben definite, e che hanno poco in comune con il training set di dati. Per ovviare al problema si può utilizzare un secondo tipo di inizializzazione, denominata random prototypes.

**Random prototypes** E’ un’inizializzazione intermedia tra la tecnica precedente e l’estrazione casuale di uno dei pattern simulati:

$$p_{\text{proto}}^c = \alpha p_{\text{rand}} + (1 - \alpha) x_{\text{rand}}^{\{1,2,\ldots,N_w\}}$$

dove $\alpha$ ($0 < \alpha \leq 1$) misura quanto prevale la componente casuale rispetto al dato estratto a caso. In questo tipo di inizializzazione si è posto $\alpha = 0.005$. 
4.4. LA SCELTA DEL MIGLIOR PUNTO DI INIZIALIZZAZIONE

Max-min L’idea di base di questo metodo di inizializzazione è di ottenere centroidi ben distanziati tra di loro e vicini al maggior numero possibile di elementi. Obiettivo che viene realizzato mediante la seguente procedura. Supposto di aver fissato $c$ centroidi per ogni dato viene calcolata la distanza minima dai centroidi noti. Il punto che possiede la maggior distanza minima, ragionevolmente, indicherà una zona mal descritta dall’attuale insieme di centroidi. La procedura prevede che questo punto diventi il $c+1$-esimo centroide. Questa inizializzazione viene realizzata dal seguente algoritmo:

$$
p_{\text{max-min}}^{1} = x_{\text{rand}}^{\{1,2,...,N\}}
$$

$$
p_{\text{max-min}}^{c} = \arg \max_{w} \min_{i=1,...,c-1} d(x^{w}, p_{\text{max-min}}^{i})
$$

dove la funzione $d(x, y)$ esprime la distanza tra i punti $x$ e $y$.

Hypercube centre In questa tecnica tutti centroidi sono fissati nelle immediate vicinanze del centro dell’ipercubo, ovvero nel punto di coordinate $\bar{x} = (0.5, ..., 0.5)$. I diversi centroidi sono dati dalla:

$$
p_{\text{centre}}^{c} = \alpha \bar{x} + (1 - \alpha)p_{\text{rand}}^{c}
$$

Sommare un valore casuale al centro dell’ipercubo consente di ottenere centroidi diversi tra loro. Va sottolineato, infatti, che determinati algoritmi, quali il K-Means, non possono tollerare pattern identici: sommando un piccolo errore casuale ($\alpha = 0.005$) si evita tale rischio.

Data mean I centroidi sono fissati intorno alla media dei dati, ovvero nella regione dove i dati sono maggiormente concentrati. Indicato con $\bar{x}$ il baricentro dei dati, i centroidi vengono calcolati, così come nel caso precedente, sommando a tale valore un errore

$$
p_{\text{mean}}^{c} = \alpha \bar{x} + (1 - \alpha)p_{\text{rand}}^{c}
$$

Anche in questo caso si è scelto $\alpha = 0.005$.

4.4.2 L’esperimento proposto

Le tecniche di inizializzazione sono tutte dotate di una componente casuale. Al fine di rimuovere l’aleatoricità dal confronto si sono effettuate, per ogni tecnica
Inizializzazione | SOM | Neural Gas | K-Means | EM  
---|---|---|---|---  
Random       | 0.028159 | 0.039583 | 0.042623 | 0.20953  
Data Mean    | 0.029286 | 0.043317 | 0.087786 | 0.21315  
Hypercube center | 0.029310 | 0.043079 | 0.040369 | 0.19346  
Max-Min      | 0.029429 | 0.043103 | 0.081821 | 0.60589  
Random Prototype | 0.028706 | 0.037829 | 0.079147 | 0.61586  

Tabella 4.4: Media delle mediane dell’indice $F$ tra i vari set di dati

testata, 60 classificazioni ognuna basata su di una differente estrazione. Tutte le classificazioni vengono valutate mediante l’indice $F$ e la mediana della distribuzione sarà considerata quale indice della tecnica di inizializzazione per il determinato set di dati.

L’esperimento proposto viene ripetuto per tutti i 12 set di dati simulati costituenti il benchmark proposto. Volendo ottenere un criterio unico per valutare le prestazioni del metodo d’inizializzazione, si è deciso di adottare la media delle mediane quale cifra di merito.

### 4.5 Risultati Sperimentali

L’esperimento descritto è stato eseguito su tutti quattro metodi ancora in confronto, I risultati dei test sui singoli benchmark sono riportati nelle Figure 4.5, 4.6, 4.7 e 4.8; mentre nella nella Tabella 4.4 sono riportati i valori della media delle mediane, di tutti i dodici benchmark, per ogni algoritmo e per ogni metodo di inizializzazione.

Consideriamo ora i risultati per ogni metodo e successivamente la scelta del metodo migliore.

#### 4.5.1 Mappe di Kohonen

Il confronto tra le varie tecniche di inizializzazione, per l’algoritmo $SOM$, ha portato ai risultati delle simulazioni riassunte in Figura 4.5. Dall’analisi dei risultati indicati in Figura 4.5 risulta evidente che sia la dispersione dei valori dell’indice $F$ sia la sua mediana, non risentono sostanzialmente del metodo di inizializzazione adottato.
Figura 4.5: Confronto fra le tecniche di inizializzazione per l’algoritmo SOM
Questo conferma la bontà della scelta della funzione di velocità di apprendimento fatta nella determinazione dell'algoritmo.

I valori riportati nella Tabella 4.4 relative alle mappe di Kohonen, sono molto bassi per tutti le inizializzazioni; il numero dei wafer misclassificati, infatti, risulta mediamente inferiore al 3.5%. Comunque il metodo di inizializzazione migliore risulta essere quello random.

4.5.2 Neural Gas

Il confronto tra le inizializzazioni è riassunto dai cinque grafici in Figura 4.6. Risulta evidente, anche in questo caso, un andamento decrescente dell’indice F con l’aumentare dei wafer e dei dispositivi. In particolare un notevole miglioramento delle prestazioni si riscontra nei benchmark 10, 11 e 12, con 950 dispositivi e wafer che crescono da 500 ad 800 e 1200 rispettivamente. Come indicato nella Tabella 4.4 l’inizializzazione migliore risulta essere quella denominata random prototypes, anche se come già sottolineato per le mappe di Kohonen, il classificatore basato sulle reti neural gas fornisce buone prestazioni con tutti i metodi di inizializzazione.

4.5.3 Algoritmo K-Means

In Figura 4.7 vengono riassunti i risultati ottenuti per questo classificatore. Risulta subito evidente come la dispersione dei valori sia elevata per tutti i metodi di inizializzazione, in particolare per l’inizializzazione max-min. I valori migliori dell’indice si ottengono con l’aumentare del numero di wafers e di dispositivi; infatti per i benchmark centrali, dal 5 all’8, corrispondenti a 377 dispositivi, si hanno le prestazioni peggiori. Arrivando a 950 dispositivi, la percentuale di wafer misclassificati si riduce drasticamente, in particolare per l’inizializzazione random e quella ”hypercube centre”, dove possiamo osservare come la mediana risulti vicinissima al caso migliore, il minimo dei valori.

Per l’algoritmo K-Means si è deciso di considerare le prestazioni di due inizializzazioni, quella random e hypercube centre, che risultano molto simili nelle prestazioni, come si può osservare in Tabella 4.4.
Figura 4.6: Confronto fra le tecniche di inizializzazione per l’algoritmo Neural Gas
Figura 4.7: Confronto fra le tecniche di inizializzazione per l'algoritmo K-Means
4.5.4 Algoritmo EM

Le prestazioni dell’algoritmo EM, riassunte in Figura 4.8, non sono risultate, in generale, molto soddisfacenti. Particolarmente deludenti si sono rivelate quelle ottenute con i metodi di inizializzazione max-min e random prototypes. In questi ultimi due casi, infatti, la percentuale dei wafer misclassificati è molto alta, anche nel caso migliore, dove, mediamente, i valori minimi dell’indice $F$ si assestano su valori superiori allo 0.6. Le prestazioni migliorano nettamente con le inizializzazioni random, hypercube centre e data mean. In questi tre casi, si ritrova l’andamento decrescente dell’indice $F$ con l’aumentare del numero di wafer, da 300 a 500, 800 e 1200.

Nel complesso le medie delle mediane, riportate in Tabella 4.4, sono alte. Il metodo di inizializzazione più efficace risulta essere l’hypercube centre, con un numero di wafer misclassificati di poco inferiore del 20%.

4.5.5 Confronto fra i metodi

Dopo aver stabilito, per ogni algoritmo, quale sia la tecnica di inizializzazione che garantisce le migliori prestazioni, possono essere confrontati tra di loro i quattro algoritmi considerati. Ogni algoritmo è stato inizializzato per mezzo della tecnica più efficace. Per l’algoritmo K-Means le simulazioni sono invece stati utilizzati due metodi di inizializzazione (random e hypercube centre). I risultati sono riassunti in Figura 4.9.

Si può notare che il classificatore di Kohonen risulta essere il più efficace in tutti i set considerati tranne che nel quinto set di dati. In quel caso il metodo fornisce il secondo miglior risultato. Pertanto l’algoritmo SOM con la taratura dei parametri presentata in questa sede, viene considerato come il miglior classificatore per l’identificazione delle mappe EWS binarie.
Figura 4.8: Confronto fra le tecniche di inizializzazione per l’algoritmo EM
Figura 4.9: Confronto dei metodi di classificazione
CAPITOLO 4. CONFRONTO DEI CLASSIFICATORI PER MAPPE BINARIE
Capitolo 5

La diagnosi di processo

*All models are wrong, some are useful*

*G. E. P. Box*

In questo capitolo verrà trattato il terzo passo della procedura di diagnosi illustrata in questa sede, presentando le diverse tecniche proposte per effettuare la diagnosi di processo.

L’obiettivo di questa analisi è l’individuazione della o delle lavorazioni responsabili del calo di resa sistemico riscontrato nei wafer selezionati nei precedenti passi della procedura proposta. Il numero delle lavorazioni che sono state effettuate per la produzione dei lotti in esame risulta essere abbondantemente superiore al migliaio. La complessità e le dimensioni del dato in esame rendono la diagnosi di processo un problema di difficile trattazione.

Le soluzioni proposte sono riconducibili a tre diversi tipi di approcci: Data Mining statistico, model-based e probabilistico.

### 5.1 Diagnosi mediante Data Mining statistico

L’approccio proposto si basa sull’assunzione che il calo di resa sistemico in analisi sia dovuto ad un singolo passo di processo. Per passo di processo o lavorazione ci
si riferisce alla coppia $P_s := (O_p, E_q)$, dove $O_p$ è la singola operazione riportata in una storia lotti (vedi Figura 2.1) mentre $E_q$ indica l’attrezzatura che ha eseguito l’operazione $O_p$.

Generalmente la stessa operazione può essere eseguita da più attrezzature. Ad esempio, considerando due differenti storie lotti, l’operazione $O_1$ può essere eseguita dall’attrezzatura $E_1$ nella prima, e dall’attrezzatura $E_2$ per la seconda. Inoltre la stessa attrezzatura può effettuare differenti operazioni durante la lavorazione di uno o più lotti. Risulta quindi chiaro come, sebbene il numero di apparecchiature non sia elevato quanto quello delle operazioni coinvolte, le lavorazioni effettuate nella produzione di una dozzina di lotti superi abbondantemente il migliaio.

L’obiettivo che si pone questo tipo di approccio è quello di individuare il passo di processo $P^* = (O^*, E^*)$ responsabile del difetto in esame e descritto da una o più firme d’errore.

Una volta selezionate le classi che descrivono il problema in esame, le $N_w$ mappe in analisi possono essere divise in due macro classi: la prima, definita $W_{bad}$, è costituita dai wafer appartenenti ai clusters selezionati, e la seconda contenete il resto delle mappe chiamata $W_{good}$. Analogamente i wafer appartenenti alle due classi verranno definiti bad wafer se appartengono a $W_{bad}$ oppure good wafer in caso contrario.

In questo approccio le $N_m$ lavorazioni vengono considerate singolarmente e, per ognuna di queste, si valuterà il grado di correlazione con la produzione di bad wafer. La criticità di una lavorazione $P_s = (O, E)$ viene valutata considerando quanti wafer, appartenenti ad entrambi gli insiemi, sono state lavorati dall’attrezzatura $E$ rispetto quelli che hanno subito l’operazione $O$. L’informazione richiesta viene riportata in una tabella di contingenza come quella raffigurata nella Tabella 5.1.

### Tabella 5.1: Tabella di contingenza per la lavorazione $(O, E)$.

<table>
<thead>
<tr>
<th>Operazione $O$</th>
<th>Attrezzatura $E$</th>
<th>Altre attrezzature</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>bad wafer</td>
<td>18</td>
<td>42</td>
<td>60</td>
</tr>
<tr>
<td>good wafer</td>
<td>32</td>
<td>108</td>
<td>140</td>
</tr>
<tr>
<td><strong>Totali</strong></td>
<td><strong>50</strong></td>
<td><strong>150</strong></td>
<td><strong>200</strong></td>
</tr>
</tbody>
</table>


5.1. L’analisi di una tabella di contingenza

Ottenuta la tabella di contingenza per una determinata lavorazione \((O, E)\), ci si pone il problema di verificare se l’impiego dell’attrezzatura \(E\) nell’operazione \(O\) risulti determinante per la produzione di bad wafer. La Tabella 5.1 mostra che, dei 200 wafer che hanno effettuato l’operazione \(O\), 60 appartengono a l’insieme \(W_{bad}\) ed i restanti 140 all’insieme \(W_{good}\). Considerando la produzione di bad wafer, si può affermare che l’operazione \(O\) abbia un’efficienza del \(\frac{60}{200} = 30\%\). Durante questa operazione l’attrezzatura \(E\) ha processato 50 wafer, se l’efficienza dell’attrezzatura è pari a quella dell’operazione, ci si attende che \(50 \times \frac{60}{200} = 15\) dei wafer lavorati appartengano a \(W_{good}\). La lavorazione considerata ha però prodotto 18 bad wafer. Questa discrepanza può essere di natura casuale e dipendere dalla numerosità del campione (50 per la lavorazione contro i 200 dell’operazione) oppure può essere imputabile ad una differente efficienza dell’attrezzatura \(E\) rispetto a quella delle altre utilizzate per eseguire la stessa operazione.

Qualora le due efficienza fossero le stesse, si dimostra [35] che la quantità

\[
C_{O,E} = \frac{(W_b^O(E)W_g^O(E) - W_g^O(E)W_b^O(E))^2 N_w^O}{W_b^O W_g^O (W_b^O(E) + W_g^O(E))(W_b^O(E) + W_g^O(E))} \tag{5.1}
\]

dove

\(N_w^O\) wafer che hanno subito l’operazione \(O\)

\(W_b^O\) bad wafer che hanno subito l’operazione \(O\)

\(W_g^O\) good wafer che hanno subito l’operazione \(O\)

\(W_b^O(E)\) bad wafer che hanno subito la lavorazione \((O, E)\)

\(W_g^O(E)\) good wafer che hanno subito la lavorazione \((O, E)\)

\(W_g^O(E)\) good wafer processati da attrezzature altre da \(E\) nell’operazione \(O\)

\(W_b^O(E)\) bad wafer processati da attrezzature altre da \(E\) nell’operazione \(O\)

è distribuita come un \(\chi^2\) standard ad un grado di libertà.
Tabella 5.2: Tabella di contingenza per la lavorazione \((O_1, E_1)\). Si nota come l’attrezzatura \(E_1\) produca un ridotto numero di bad wafers.

Quindi, un buon test per verificare l’efficienza di una lavorazione \((O, E)\) è dato dalla seguente

\[
C_{O,E} > \chi^2_{\alpha} \tag{5.2}
\]

dove \(\chi^2_{\alpha}\) rappresenta un appropriato valore della distribuzione standard \(\chi^2\) ad un grado di libertà. Solitamente si utilizza come soglia un valore di significatività pari al 5% che corrisponde ad un valore della variabile \(\chi^2_{95\%}\) pari a 3.841.

5.1.2 L’identificazione delle possibili cause

Un elevato valore di \(C_{O,E}\) indica che la lavorazione considerata presenta un’efficienza superiore od inferiore a quella dell’operazione \(O\). Non viene fornita nessuna indicazione riguardante la resa, in termini di produzione di bad wafer, del passo di processo in esame.

In Tabella 5.2 viene mostrata la tabella di contingenza relativa alla lavorazione \((O_1, E_1)\). Sebbene \(C_{O_1,E_1}\) risulti essere abbastanza elevato, e comunque superiore alla soglia definita in precedenza, si nota come l’attrezzatura \(E_1\) difficilmente potrebbe essere considerata responsabile del difetto in esame. Infatti la sua produzione di bad wafer risulta veramente esigua.

Per riconoscere fra, tutte le lavorazioni che soddisfano la 5.2, quelle effettivamente legate alla produzione di bad wafer è sufficiente controllare che il rendimento, inteso come produttività di good wafer, sia minore di quello delle altre attrezzature che realizzano la stessa operazione. Considerando la generica lavorazione se \((O, E)\) se la diseguaglianza

\[
\frac{W_b^O(E)}{W_g^O(E)} > \frac{W_b^O(E)}{W_g^O(E)} \tag{5.3}
\]
viene soddisfatta la lavorazione può considerarsi una possibile causa del caso di resa in esame.

Ricapitolando, tutte le lavorazioni che soddisfano sia la (5.3) che la (5.2) possono essere considerate come possibili cause del problema in esame. Inoltre se le diverse lavorazioni che soddisfano questi criteri sono ordinate per valori decrescenti di $C_{O,E}$, si ottiene una classifica che riporta nelle prime posizioni i passi di processo più fortemente correlati con la produzione di bad wafer. La classifica così ottenuta rappresenta il risultato dell’analisi di processo.

5.1.3 Le limitazioni del metodo

L’analisi svolta dall’approccio proposto, è da considerarsi di tipo locale. Essa infatti considera le operazioni in maniera indipendente una dall’altra senza sfruttare la visione d’insieme fornita dai vari percorsi di produzione contenuta nella storia lotti.

Un’altra limitazione di questo metodo è l’incapacità di gestire macchine singole. Per macchina singola s’intende una macchina che effettua una determinata operazione sull’intera produzione in esame. In questo caso, non essendo presenti altre attrezzature che svolgano la stessa operazione, risulta impossibile costruire una tabella di contingenza valida, e quindi utilizzare il metodo proposto.

Nel corso dell’impiego di questo metodo ci si è accorti che le due limitazioni, per quanto reali, non hanno intaccato la validità della metodologia proposta.

5.2 Approccio Model-Based

Per superare le limitazioni teoriche riscontrate nella precedente analisi sono stati studiati e proposti due metodi entrambi basati su di un modello in grado di considerare i differenti percorsi di produzione seguiti dai singoli lotti in analisi.

5.2.1 Il Modello

Come avveniva nell’approccio precedente anche in questo caso, il modello proposto considera come wafer correttamente lavorati tutti quelli appartenenti a $W_{good}$ lasciando a $W_{bad}$ il compito di descrivere il difetto in analisi. In quest’ottica, per ogni
selezione di pattern identificati, possiamo definire una resa per ogni lotto:

\[ R_j = \frac{W_{\text{good}}(j)}{W(j)}, \quad j = 1...N_l \]  

(5.4)
dove \( N_l \) è il numero totale dei lotti considerati, \( W_{\text{good}}(j) \) è il numero dei good wafer appartenenti al lotto \( j \) mentre \( W(j) \) è il numero totale dei wafer appartenenti al lotto \( j \).

Nel modello proposto viene associato ad ogni step di processo un valore \( \eta \) compreso nell’intervallo \([0, 1]\) che indica la probabilità per quello step di processo di generare good wafer. Definendo l’evento \( E_i \) come il corretto funzionamento dell’i-esimo step di processo risulterà

\[ \eta_i = P(E_i), \quad i = 1, ..., N_m \]

In questo modello il corretto funzionamento dell’i-esima lavorazione costituisce un successo in una prova di Bernoulli con probabilità di successo \( \eta_i \).

Sotto l’ipotesi che il corretto funzionamento di uno step di processo non dipenda dal comportamento delle altre fasi coinvolte è possibile affermare che la probabilità che un prodotto venga lavorato correttamente sarà data dalla produttoria delle probabilità di corretto funzionamento di tutti gli step coinvolti.

Per agevolare la notazione, introduciamo una variabile d’attivazione \( T_{i,j} \) di tipo binario: il suo valore sarà posto ad ‘1’ se il \( j \)-esimo lotto è stato sottoposto allo step di processo \( i \), mentre sarà ‘0’ se questo non è avvenuto.

Alla luce di queste definizioni è possibile scrivere per il \( j \)-esimo lotto la probabilità di ottenere una lavorazione corretta come

\[ R_j = \prod_{i=1}^{N_m} \eta_i^{T_{i,j}} \]  

(5.5)

Le variabili di attivazione \( T_{i,j} \) vengono raccolte in una matrice \( T \) detta matrice di processo.

Generalizzando la (5.5) su tutti i lotti si ottiene il sistema

\[
\begin{cases}
R_1 = \prod_{i=1}^{N_m} \eta_i^{T_{i,1}} \\
\vdots \\
R_{N_l} = \prod_{i=1}^{N_m} \eta_i^{T_{i,N_l}}
\end{cases}
\]  

(5.6)
Nella definizione del modello (5.6) le uniche informazioni richieste sono la classificazione per determinare il vettore dei rendimenti e la storia lotto da cui estrarre la matrice di processo.

Definendo con $M(N_r, N_c)$ lo spazio delle matrici ad $N_r$ righe e $N_c$ colonne, non tutte le matrici ad esso appartenenti possono però essere considerate matrici di processo. Queste ultime, in forza del loro significato, debbono rispettare alcuni requisiti riportati nella seguente definizione:

Definizione 1: La matrice $T$ di dimensioni $N_m \times N_l$ appartiene alla famiglia $B(N_m, N_l)$ se ha le seguenti proprietà:

- $T_{i,j} \in \{0, 1\} \forall i, j$ (la matrice $T$ è binaria)
- $\sum_{j=1}^{N_l} T_{i,j} \neq 0 \forall i$ (la matrice $T$ non ha righe di soli zeri)
- $\sum_{i=1}^{N_m} T_{i,j} \neq 0 \forall j$ (la matrice $T$ non ha colonne di soli zeri)

Il secondo e il terzo punto della definizione precedente si riferiscono rispettivamente al fatto che non rientreranno nella matrice di processo nè step che non hanno processato almeno un lotto nè, d’altro canto, lotti che non sono stati sottoposti ad almeno uno step di processo.

Il modello descritto dalle (5.6) risulta di difficile utilizzo per gli scopi prefissi, pertanto se ne propone una formulazione più pratica. Consideriamo una generica equazione del modello:

$$R_j = \prod_{i=1}^{N_m} \eta_i^T_{i,j}$$

 Questa può essere riscritta nella forma

$$-\log(R_j) = -\log(\prod_{i=1}^{N_m} \eta_i^{T_{i,j}})$$ (5.7)

 da cui si ricava

$$-\log(R_j) = -\sum_{i=1}^{N_m} \log(\eta_i^{T_{i,j}}) = \sum_{i=1}^{N_m} T_{i,j}(-\log(\eta_i))$$
Il sistema (5.6) diventa quindi
\[
\begin{align*}
- \log(R_1) &= \sum_{i=1}^{N_m} T_{i,1}(- \log(\eta_i)) \\
&\vdots \\
- \log(R_{N_l}) &= \sum_{i=1}^{N_m} T_{i,N_l}(- \log(\eta_i))
\end{align*}
\]  
(5.8)

Posso ora riscrivere in forma compatta il sistema (5.8) come
\[
T'y = b
\]  
(5.9)

dove
\[
y \in [y_1, ..., y_{N_m}]^T \text{ con } y_i = - \log(\eta_i)
\]  
(5.10)
e
\[
b \in [b_1, ..., b_{N_l}]^T \text{ con } b_i = - \log(R_i)
\]  
(5.11)

Si può osservare come sia stato possibile ottenere un sistema lineare introducendo le variabili ausiliarie \( y_i \).

### 5.2.2 La formulazione del problema

Il modello descritto dalle (5.9) ,(5.10),(5.11) nel precedente paragrafo permette di ricavare a partire dal set dei parametri \( \eta \) tramite la matrice di processo \( T \), il vettore dei rendimenti dei lotti \( R \). In una diagnosi di processo viene richiesto esattamente l’inverso: ricavare una stima delle probabilità \( \eta_i \) di corretta esecuzione di ciascuno step di processo, a partire dal vettore \( R \) delle rese dei lotti. Una possibile soluzione è data da:
\[
\tilde{y} = \arg \min_{y_i > 0} \|b - T'y\|^2
\]  
(5.12)
da cui si ricava
\[
\tilde{\eta} = \exp(-\tilde{y})
\]  
(5.13)

Il sistema descritto dalla (5.9) presenta un numero di incognite spesso molto superiore a quello delle equazioni e di conseguenza non è possibile determinare univocamente \( \tilde{\eta} \). Le cause che portano ad avere un sistema di questo tipo risiedono nel fatto che è possibile scrivere un’equazione per ogni lotto di cui conosciamo la resa, mentre il numero delle incognite corrisponde al numero di lavorazioni impiegate per
portare a termine il processo produttivo di tutti i lotti. Di norma il numero di step di processo è molto maggiore del numero di lotti relativi al periodo in analisi.

Per poter fornire una valida stima $\tilde{\eta}$ tra tutte le possibili soluzioni del sistema, ossia tra tutti i possibili vettori di rendimenti compatibili con i dati, occorre determinare quelle più plausibili alla luce delle nostre conoscenze sul processo.

Durante l’esecuzione di un processo, sebbene complesso come quello in analisi, è lecito supporre che il numero di macchine che possano presentare un malfunzionamento sia contenuto. E’ molto raro, infatti, che le lavorazioni responsabili di un singolo calo di resa sistematico possano essere più di mezza decina. Questa semplice osservazione fa sì che tra tutte le possibili soluzioni vengano privilegiate quelle che presentano il maggior numero di elementi del vettore $\eta$ prossimi ad uno.

Un modo di imporre questa conoscenza a priori è quello di ricercare, fra soluzioni del problema (5.12), quelle che presentino solo $N$ componenti maggiori di zero. A tale scopo si definisce l’insieme $Y^N$ delle soluzioni ammissibili.

Definizione 2 Dato $N > 0$ e $y \in \mathbb{R}^n$ con $n \geq N$, diremo che $y \in Y^N$ se il numero di elementi $y_i \neq 0$ del vettore $y$ è minore o uguale a $N$. Si noti che $Y^{N-1} \subset Y^N$.

Definito l’insieme $Y^N$ è possibile dare una nuova formulazione del problema (5.12):

$$\tilde{y} = \arg\min_{y \in Y^N} \| b - T' y \|^2$$  
(5.14)

$$\tilde{\eta} = \exp(-\tilde{y})$$  
(5.15)

5.2.3 Criteri di esistenza ed unicità

Nella formulazione (5.14) - (5.15) il sistema presenta per costruzione solamente $N$ gradi di libertà poiché i restanti $N_m - N$ valori del vettore $y$ sono nulli. In questo modo è stato possibile ridurre sensibilmente la dimensionalità. Inoltre, la risoluzione di un problema di minimo garantisce l’esistenza di almeno una soluzione sebbene non ne assicuri l’unicità. Prima di iniziare la trattazione vera e propria è opportuno introdurre alcune definizioni preliminari.

Definizione 3 Data una matrice $T \in M(N_m, N_l)$ e un insieme di indici
I = \{i_1, ..., i_n\}, 1 \leq i_j \leq N_m \ con \ T(I) = [T_{i,j}] \ i \in I \ e \ j = 1, ..., N_t \ indicheremo \ la \ matrice \ ottenuta \ azzerando \ gli \ elementi \ delle \ righe \ \notin I \ della \ matrice \ T. 

**Definizione 4** La matrice T appartiene alla classe \( O^N \) se

- \( T \in B \)
- \( \forall I \) avente cardinalità \( N + 1 \) tutte le matrici \( T(I) \) di \( T \) rispettano la condizione \( \text{rango } T(I) = N + 1 \);

Si noti che per \( N = 1 \), essendo la matrice \( T \) binaria, la definizione (4) è equivalente a richiedere che la matrice \( T \) non abbia righe uguali \( (T(i) \neq T(j) \ \forall i \neq j) \).

Nell’esporre i risultati dell’analisi fatta verrà considerato in un primo momento il caso in cui la soluzione del problema reale \( \bar{y} \) appartenga all’insieme \( Y^1 \) e successivamente sarà considerato un caso più generale di soluzione in \( Y^M \).

**Teorema 1** Si ipotizzi \( T \in O^1 \) e che \( b = T'\bar{y}, \bar{y} \in Y^1 \). Allora, la soluzione del Problema (5.14) con \( N = 1 \) è unica e coincide con \( \bar{y} \).

*Dimostrazione:* si supponga per assurdo che il Problema (5.14) ammetta una seconda soluzione \( \tilde{y} \neq \bar{y}, \bar{y} \in Y^1 \). Dato che \( b - T'\bar{y} = 0 \) deve risultare anche \( b - T'\tilde{y} = 0 \), da cui \( T'(\bar{y} - \tilde{y}) = 0 \). Si indichino ora con \( \bar{i} \) e \( \tilde{i} \) gli indici tali che \( \bar{y}_{\bar{i}} \neq 0, \tilde{y}_{\tilde{i}} \neq 0 \), e si definisca \( I = \{\bar{i}, \tilde{i}\} \). Risulta che \( T(I)'[\bar{y}_{\bar{i}} - \tilde{y}_{\tilde{i}}] = 0 \) contraddicendo l’ipotesi \( T \in O^1 \).

*Q.E.D.*

Nel Teorema 1 viene asserito che la soluzione vera appartiene a \( Y^1 \). Non sempre quest’ipotesi è soddisfatta. Viene pertanto proposto un altro teorema atto a descrivere questa situazione.

**Teorema 2** Si ipotizzi che \( T \in O^{N+M-1} \) e che \( b = T'\bar{y}, \bar{y} \in Y^M \). Allora la soluzione del problema (5.14) è unica e coincide con \( \bar{y} \).
5.2. APPROCCIO MODEL-BASED

Dimostrazione: si supponga per assurdo che il Problema (5.14) ammetta soluzione \( \tilde{y} \neq \bar{y}, \bar{y} \in Y^N \). Dato che \( b - T' \bar{y} = 0 \) deve risultare anche \( b - T' \tilde{y} = 0 \), da cui \( T' \bar{y} = T' \tilde{y} \). Definendo \( \bar{I} \) e \( \tilde{I} \) in modo analogo al Teorema precedente, si ottiene

\[
T(\bar{I})' \bar{y} = T(\tilde{I})' \tilde{y} \tag{5.16}
\]

Si possono verificare due casi, a seconda che \( \bar{I} \subset \tilde{I} \) oppure no. Nel primo caso una soluzione del sistema (5.16) nell’incognita \( \tilde{y} \) è data \( \tilde{y} = \bar{y} \) e tale soluzione è unica poiché \( T(\bar{I}) \) ha rango pieno. Nel caso in cui \( \bar{I} \) non è incluso in \( \tilde{I} \) il sistema ammette soluzione per il Teorema di Rouchè - Capelli se il rango di \( T(\bar{I}) \) è uguale al rango di \( T(\{\bar{I}, \tilde{I}\}) \), contro le ipotesi in base alle quali \( T(\{\bar{I}, \tilde{I}\}) \) ha rango pieno.

Q.E.D.

I criteri qui esposti, sebbene abbiano validità generale, impongono vincoli molto restrittivi: infatti, le condizioni di rango sulle sottomatrici della matrice \( T \) possono rappresentare un requisito molto forte anche per valori di \( N \) e \( M \) non necessariamente elevati. Per tanto, si è deciso di limitare la ricerca in insiemi a piccole dimensioni. Inoltre, sono stati sviluppati alcuni criteri che consentono di determinare l’appartenenza di una matrice alla classe \( O^N \) con un numero ridotto di controlli. Questi criteri vengono riportati nell’Appendice 7.2.

5.2.4 Il primo algoritmo risolutivo

La risoluzione del problema di minimo vincolato (5.14) - (5.15) presenta notevoli difficoltà dovute sia alla presenza di minimi locali, qualora non siano verificate le condizioni espresse nel Teorema 2, sia all’impossibilità di risolvere il problema in forma chiusa [40]. Quest’ultima è imputabile alla non regolarità dell’insieme \( Y^N \).

L’insieme \( Y^N \) può essere scomposto in una serie di sotto insiemi più regolari, ognuno caratterizzato dall’avere solo \( N \) dimensioni non vincolate a zero. Il numero dei suddetti insiemi è pari \( N_I = \binom{N_m}{N} \) che indica il numero delle combinazioni di \( N_m \) a gruppi \( N \). Se l’ordinalità delle dimensioni non vincolate viene inserita in un insieme di indici \( I_k \) i summenzionati insiemi sono definiti dalla seguente:

Definizione 5 Dato l’insieme di indici \( I^k \)

\[
Y^N(I^k) := \{y : y_j = 0, \forall j \notin I^k, y_i \geq 0 \ \forall i\}
\]
allora

\[ Y^N = \bigcup_{I^k \in S^N} \bar{Y}^N(I^k) \]

dove \( S^N \) è l’insieme di tutti gli insiemi di indici aventi la cardinalità pari ad \( N \) (pertanto la cardinalità di \( S^N \) è pari ad \( N_I \)).

Il problema di minimo vincolato (5.14) risulta quindi equivalente al seguente

\[ \tilde{y} = \tilde{y}^k \text{ con } \tilde{k} = \arg\min_k \| b - T'\tilde{y}^k \|^2 \]  

(5.17)

dove

\[ \tilde{y}^k = \arg\min_{y \in \bar{Y}^N(I^k)} \| b - T'y \|^2 \]  

(5.18)

Si tratta ora di risolvere \( N_I \) problemi di minimo (5.18) e stabilire a quale vettore \( I^\tilde{k} \) sia associata la soluzione del problema di minimo (5.14). Un’analisi di questo genere prende il nome di analisi esaustiva in forza del fatto che la ricerca viene eseguita su tutti i possibili vettori \( I^k \) per \( N \) fissato.

Il problema vincolato (5.18) è risolubile in forma chiusa, mediante il metodo dei minimi quadrati per la stima di parametri (stima LS) [41]. Nel caso in esame, il set di parametri da stimare è rappresentato dal vettore \( y \in \bar{Y}^N(I^k) \). Di conseguenza nel calcolo della stima dei parametri non apparirà mai l’intera matrice \( T \), ma solo le righe relative agli step di processo individuati da \( I^k \).

Se la la matrice di processo \( T \in O^N \), la (5.18) ammette un’unica soluzione in quanto la restrizione di \( T' \) su \( Y^N(I^k) \) ha rango pieno. Si ha quindi che \( \tilde{y}^k \) è data da

\[ \tilde{y}^k = (T(I^k)T(I^k)')^{\dagger}T(I^k)b \]  

(5.19)

dove l’operatore \( \dagger \) indica la pseudoinversa. Tuttavia, è facile vedere che eliminando dalla matrice \( T \) le righe il cui indice non appartiene a \( I^k \) è possibile ottenere il medesimo risultato usando matrici di ordine \( N \).

Associata alla soluzione, vi è la cifra di merito

\[ J^k := \| b - T'\tilde{y}^k \|^2 \]

che misura l’aderenza ai rendimenti osservati.

A questo punto è possibile descrivere l’algoritmo risolutivo proposto:
1. Determinazione di \( N \) e di conseguenza degli insiemi \( I^k \) ordinati da 1 ad \( N_I \).

2. Si pone \( J^{\text{ott}} = \infty \) e \( k = 1 \).

3. Si calcola \( \tilde{y}^k \) tramite la (5.19) e la cifra di merito associata \( J^k \).

4. Se \( J^k < J^{\text{ott}} \), si pone \( I^{\text{ott}} = I^k \), \( \tilde{y}^{\text{ott}} = \tilde{y}^k \) e \( J^{\text{ott}} = J^k \).

5. Si incrementa \( k \) e si torna al Passo 3 finché \( k = N_I \).

6. Si determina la soluzione del problema (5.14) \( \tilde{y} = \tilde{y}^{\text{ott}} \).

Per poter applicare l’algoritmo proposto è necessario determinarne la dimensione \( N \) dello spazio in cui cercare la soluzione. All’aumentare del numero dei gradi di libertà \( N \) si presenta una riduzione del residuo generato (al più potrebbe rimanere invariato). Di conseguenza non è possibile determinare il numero dei parametri liberi utilizzando come criterio la minimizzazione di \( J^{\text{ott}} \) al variare di \( N \).

Di seguito è riportata una possibile procedura d’analisi al variare di \( N \):

1. Si fissa \( N = 1 \)

2. Si crea la lista (inizialmente vuota) delle righe di \( T \) che sono escluse dall’analisi in quanto combinazione lineare di altre righe

3. Si garantisce la condizione di unicità su \( T \), ovvero \( T \in O^N \), aggiungendo opportunamente nella precedente lista alcune righe di \( T \)

4. Si esegue il calcolo di \( \tilde{y} \) nell’insieme \( Y^N \) delle soluzioni

5. Si verifica il risultato direttamente sull’impianto (si può valutare, per esempio, se il set macchine associate agli step di processo individuati è tale da essere in accordo col tipo di guasto riscontrato o, più semplicemente, è possibile eseguire un esame specifico sulle macchine stesse);

Si possono verificare due situazioni:

(a) Il riscontro effettuato sull’impianto è positivo e dunque la ricerca è terminata a buon fine
Il riscontro sull’impianto non è in accordo col risultato ottenuto dalla ricerca. Proseguo nella ricerca.

6. Viene incremento il valore di $N$. Sposto quindi la ricerca in insiemi sempre crescenti e torno al passo 3

La procedura proposta presenta difficoltà di calcolo dovute alla crescita combinatoria di $N_I$. Problema che diviene ancora più importante se si pensa che gli step di processo su cui si effettuano le analisi sono generalmente migliaia. Considerare tutte le possibili combinazioni risulta spesso proibitivo, per tanto con questo tipo di algoritmo non è consigliabile ricercare una soluzione oltre l’insieme $Y^3$.

5.2.5 Il secondo algoritmo risolutivo

Per diminuire i tempi di calcolo imposti dall’analisi esaustiva viene proposto un secondo algoritmo basato sulla stepwise regression [42]. In questo approccio le componenti non nulle del vettore $y$ vengono ricercate mediante una selezione ”passo-passo”.

L’algoritmo di stepwise regression viene normalmente utilizzato per scegliere la struttura di un modello ed identificare i relativi parametri avendo a disposizione un numero $r$ di regressori ed un numero di dati $D$. Nell’ottica di risoluzione del problema di minimo (5.14) si pensa ai regressori come alle differenti lavorazioni (quindi $r = N_m$), i parametri stimati corrisponderanno alle $N_m$ componenti di $y_i$ ed i dati al numero di rendimenti dei lotti disponibili ($D = N_l$).

Per utilizzare una stepwise regression è necessario determinare la cifra di merito per la scelta dei regressori: nel caso in analisi questa è rappresentata dal $FPE$ (Final Prediction Error):

$$FPE = SSR \frac{N_l + q}{N_l - q}$$

(5.20)

dove $q$ sono i parametri attualmente utilizzati nel modello, ed SSR (Sum of Squared Residuals) è dato dalla somma dei quadrati dei residui tra le osservazioni e le predizioni ottenute stimando di volta in volta i parametri $y_i$. Solitamente, al crescere di $q$ dapprima $FPE$ diminuisce in quanto diminuisce sensibilmente la somma dei quadrati dei residui. Se il numero dei parametri aumenta troppo la $FPE$ aumenta poiché il rapporto fra $N + q$ ed $N - q$ al tende a crescere mentre il l’SSR si stabilizza.
Definita la cifra di merito, si identificano, tramite la stima LS, r modelli distinti ciascuno dei quali costituito da un singolo regressore. Tra tutti i modelli viene scelto quello costituito dal regressore (e il relativo parametro) più significativo ossia quello che minimizza la cifra di merito precedentemente scelta. Il regressore ottenuto farà parte dei modelli considerati nei passi successivi. Di passo in passo il modello viene aggiornato inserendo un nuovo regressore (quello più significativo per FPE) e ristimando tutti i parametri ad essi associati.

Alla luce della definizione degli insiemi $Y$, inserire un nuovo regressore nel modello, significa spostare la ricerca in insiemi dimensionali via via crescenti. Mentre nell’analisi esaustiva vengono stimati i parametri di tutte le possibili lavorazioni considerate prima singolarmente e poi in gruppo in funzione dell’insieme ($Y^1$, $Y^2$, $Y^3$) in cui viene effettuata la ricerca, la stepwise incorpora nel modello un regressore per volta scegliendo tra tutti quello che di volta in volta decrementa in modo più marcat o la cifra di merito adottata (5.20).

Il procedimento descritto può venire riassunto dal seguente algoritmo.

1. Si inizializzano gli insiemi dei regressori non utilizzati $I^L = \{1, ..., N_m\}$, di quelli utilizzati $I^M = \{\}$ si pone $FPE_o = +\infty$ e $q = 0$

2. per ogni $r \in I^L$

   (a) Si considerano i regressori $I = I^M \cup r$

   (b) Si identifica il modello basato sui regressori $I$ mediante stima LS

   (c) Si valuta l’$FPE(r)$

3. Pongo $r^* = \arg\min_{I^L} FPE(r)$

4. Se $FPE(r^*) > FPE_o$ allora Stop

5. Pongo $FPE_o = FPE(r^*), I^M = I^M \cup r, q = q + 1$ e rimuovo $r$ da $I^L$

6. Torno al 2

La ricerca descritta da questo algoritmo non necessita della prova di tutte le possibili combinazioni di regressori bensì richiede la sola valutazione del comportamento della cifra di merito associata ad una particolare configurazione di regressori,
CAPITOLO 5. LA DIAGNOSI DI PROCESSO

considerando l’ingresso nel modello di un parametro per volta. Il grosso limite di tale approccio risiede nella sua non esaustività ovvero nella sua subottimalità. Pertanto il risultato ottenuto è solo localmente ottimo nel senso che non è possibile migliorare aggiungendo un singolo regressore.

5.3 Approccio Probabilistico

Nel precedente approccio è stata introdotta l’ipotesi secondo cui il numero di lavorazioni responsabili indiziato di un fallimento è limitato. Quest’ipotesi, peraltro riscontrata nella realtà produttiva, veniva tradotta in un vincolo esplicito nella stima dei rendimenti delle variabili. In questa sezione si presenta un algoritmo che, invece di forzare l’ipotesi, ne introduce le conoscenze direttamente nella stima dei parametri \( \eta \) senza limitarne i valori assegnabili.

Supponiamo che \( \eta \) abbia una densità di probabilità definita come \( f_\eta(\eta) \) (prior), che riflette le informazioni disponibili a priori riguardo al vettore dei parametri. Per la definizione di probabilità condizionata è possibile scrivere

\[
f_{R,\eta}(R, \eta) = f_{R|\eta}(R | \eta) f_\eta(\eta)
\]

che mostra come la densità di probabilità congiunta dei vettori dei dati e dei parametri possa essere espressa come il prodotto di un termine di verosimiglianza \( f_{R|\eta}(R | \eta) \) e del prior.

5.3.1 La determinazione del prior

L’ipotesi introdotta in precedenza, prevede che la maggior parte delle lavorazioni abbia un rendimento vicino all’unità, si è pertanto ipotizzato che la densità di probabilità per una generica variabile \( \eta_i \) sia:

\[
\begin{align*}
f_\eta(\eta) &= a e^{\rho(\eta - 1)}, & 0 \leq \eta \leq 1 \\
f_\eta(\eta) &= 0, & \text{altrimenti}
\end{align*}
\]

(5.21)

Il parametro \( \rho \) viene utilizzato per descrivere l’andamento della curva. Il parametro \( a \) si ottiene imponendo che l’integrale della d.d.p. esteso a tutto l’asse sia pari ad uno:

\[
\int_{-\infty}^{+\infty} f_\eta(\eta)d\eta = \int_{0}^{1} f_\eta(\eta)d\eta = \int_{0}^{+1} a e^{\rho(\eta - 1)}d\eta =
\]
5.3. APPROCCIO PROBABILISTICO

![Diagram](image)

Figura 5.1: d.d.p. di $\eta$ in funzione del parametro $\rho$.

\[
a \int_{0}^{1} e^{\rho(\eta-1)} d\eta = a \int_{0}^{1} e^{\rho \eta} d\eta = a \left[ \frac{1}{e^{\rho \eta}} \right]^{+1}_{0} = \frac{a}{e^{\rho \eta}} e^{\rho \eta} \left[ e^{\rho \eta} \right]^{+1}_{0} = \frac{a}{e^{\rho \eta}} (e^{\rho} - 1) = 1
\]

Da quest’ultima relazione si ricava:

\[
a = \frac{\rho e^\rho}{e^\rho - 1}
\]

Sostituendo il valore di $a$ nella 5.21 si ottiene

\[
\begin{cases}
f_\eta(\eta) = \frac{\rho e^\rho}{e^{\rho-1}} e^{\rho(\eta-1)} = \frac{\rho e^\rho}{e^{\rho-1}}, & 0 \leq \eta \leq 1 \\
f_\eta(\eta) = 0 & \text{altrimenti}
\end{cases}
\]

Nel grafico in Figura 5.1 è riportato un andamento delle curve associate alle funzioni di densità di probabilità della variabile $\eta$ per diversi valori del parametro $\rho$. Come si può osservare al crescere del valore del parametro $\rho$, le curve tendono a schiacciarsi sull’asse delle ascisse per poi crescere vicino al valore 1. In questo modo più grande è il valore di $\rho$ minore è la probabilità di avere lavorazioni con rendimenti bassi.

Dato che nella formulazione del problema si farà riferimento ancora a $y = -\log(\eta)$, ricaviamo la d.d.p. a priori di $y$. Per una generica funzione $y = g(\eta)$, data la d.d.p. di $\eta$ posso calcolare la d.d.p. di $y$ tramite:
CAPITOLO 5. LA DIAGNOSI DI PROCESSO

\[ f_Y(y) = \frac{f_\eta(\eta(y))}{|g'(\eta(y))|} \]
dove
\[ y = g(\eta) = -\log(\eta) \quad (5.22) \]
Sostituendo si ottiene
\[ f_Y(y) = \frac{f_\eta(\eta(y))}{|g'(\eta(y))|} = \left| \frac{\rho e^{\rho e^{-y}}}{e^{\rho} - 1} \right| = \frac{e^{-y}\rho e^{\rho e^{-y}}}{e^{\rho} - 1} = \frac{\rho}{e^{\rho} - 1} (e^{-y + \rho e^{-y}}) \]
Considerando quindi la d.d.p. relativa all’i-esima lavorazione in esame:
\[ f_{y_i}(y_i) = \frac{\rho}{e^{\rho} - 1} (e^{-y_i + \rho e^{-y_i}}) \]
Ipotizzando che le \( y_i \) siano tra loro indipendenti
\[ f_y(y) = \prod_{i=1}^{N_m} f_{y_i}(y_i) = \prod_{i=1}^{N_m} \frac{\rho}{e^{\rho} - 1} (e^{-y_i + \rho e^{-y_i}}) = \]
\[ = \left( \frac{\rho}{e^{\rho} - 1} \right)^{N_m} \cdot e^{\sum_{i=1}^{N_m} (-y_i + \rho e^{-y_i})} \]

5.3.2 Riformulazione del problema

L’approccio che si segue con questo nuovo tipo d’analisi non considera in modo parziale la matrice di processo bensì analizza in modo globale tutto il set di dati disponibile. Ricordando che il sistema ottenuto dal modello costruito ha più incognite che equazioni, l’obiettivo che si cerca di perseguire è quello di risolvere il sistema (5.9) andando a scegliere tra le infinite soluzioni possibili quella che sia maggiormente in linea con il prior.
\[ \tilde{y} = \arg \min_{y} \left( \frac{\rho}{e^{\rho} - 1} \right)^{N_m} \cdot e^{\sum_{i=1}^{N_m} (-y_i + \rho e^{-y_i})} \]
soggetto a
\[ T'y = b \]
Una formulazione più semplice, ma del tutto analoga, è data dalla massimizzazione del logaritmo del prior:

\[ \tilde{y} = \arg \min_y J_1(y) \]

soggetto a

\[ T'y = b \]

dove

\[ J_1(y) = \sum_{i=1}^{N_m} (-y_i + \rho e^{-y_i}) \]

Così formulato, il problema è equivalente a cercare il vettore \( y \) più probabile che soddisfa i vincoli sui rendimenti. Ovviamente il valore di \( \rho \) gioca un ruolo importante e la sua taratura deve essere eseguita con attenzione.

L’implementazione numerica di questo approccio è risultata molto complessa a causa soprattutto del grande numero di incognite in relazione al numero di osservazioni relative ai lotti. Inoltre, il vincolo di uguaglianza (\( T'y = b \)) si è rivelato essere troppo stringente ed ha appesantito i tempi di calcolo. Di conseguenza, è stato necessario rilassare questo vincolo. Ciò è stato fatto modificando il funzionale di costo, ora composto da due termini: il primo associato al prior, il secondo relativo alla risoluzione del sistema delle rese:

\[ J_2(y) = \sum_{i=1}^{N_m} (y_i - \rho e^{-y_i}) + \gamma \cdot \sum_{i=1}^{N_l} \varepsilon_i^2 = \sum_{i=1}^{N_m} (y_i - \rho e^{-y_i}) + \gamma \cdot (b - T'y)'(b - T'y) \]

Il problema con il nuovo funzionale diviene quindi

\[ \tilde{y} = \arg \min_{y \geq 0} J_2(y) \]

Come è possibile osservare, oltre a \( \rho \) è stato necessario inserire un nuovo parametro \( \gamma \), al fine di poter pesare i due termini del funzionale.

L’approccio basato sulla massimizzazione di \( J_2 \) è detto semi-bayesiano in quanto \( J_2 \) non è il logaritmo di una densità a posteriori. Infatti, la somma dei quadrati dei residui che compare nel secondo addendo, pur essendo un ragionevole indice di aderenza alle osservazioni sperimentali, non coincide con la log-verosimiglianza del modello probabilistico impiegato. D’altro canto utilizzare la log-verosimiglianza comporterebbe maggiori difficoltà computazionali.
5.4 Confronto su casi reali

Verificare la bontà di una analisi di processo non è sempre facile. In una azienda di semiconduttori le attrezzature sono sempre sottoposte a controlli e la vita stessa dei dispositivi è in continuo mutamento. Per tanto difficilmente si hanno evidenze certe di quale sia il reale responsabile del calo di resa.

In questa sezione proporremo i risultati delle analisi effettuate dai metodi proposti, su due casi reali di cui si conoscono, con ragionevole certezza, i responsabili.

5.4.1 I casi testati

I casi di studio proposti sono relativi a dispositivi flash con differente grado d’integrazione e capacità d’immagazzinamento. Entrambi i set di dati sono stati raccolti dal reparto produttivo del sito di Agrate della multinazionale STMicroelectronics.

Il primo processo

Il primo processo di lavorazione considerato viene descritto mediante una matrice di processo $T_A$ composta da 785 righe (quindi differenti lavorazioni) e 54 colonne ognuna relativa ad un diverso lotto di produzione.

Prima di poter effettuare la diagnosi è stato necessario individuare i pattern e classificare preliminarmente i wafer in analisi. Operazione realizzata mediante il tool ACID (Automatic Classification for Interactive Diagnosis) [43] che utilizza le mappe di Kohonen per stimare i pattern e classificare i wafer. Vengono riportate in Figura 5.2 i pattern di guasto individuati per la prima analisi di processo (le zone più scure delle mappe si riferiscono a valor medi di dispositivi guasti più elevati) e nell’istogramma di Figura 5.3 il numero di wafer per classe.

Le classi che descrivono il calo di resa in esame sono le 1,4,7. Si può notare come i clusters selezionati presentino pattern visualmente molto simili. Effettuata la scelta del calo di resa sistematico da analizzare si è potuto definire il vettore dei rendimenti $R_A = [ R_A(1), \ldots, R_A(54) ]$ dove ogni singolo elemento è dato dalla 5.4.

Da successive analisi e riscontri sull’impianto di produzione si è scoperto che il
5.4. CONFRONTO SU CASI REALI

Figura 5.2: Malfunzionamenti identificati per il primo processo produttivo

Figura 5.3: Suddivisione dei wafer nelle 9 classi identificate
Il secondo processo

Per quanto riguarda la seconda analisi, la matrice di processo $T_B$ è composta in modo molto differente: i dati sono relativi ad un’analisi di 470 lotti, mentre le lavorazioni coinvolte sono oltre 1800.

In Figura 5.4 viene riportato il risultato della classificazione; il difetto considerato in questa analisi è descritto dalla classe #22.

Analogamente al caso precedente i rendimenti dei lotti in analisi vengono raccolti in un vettore $R_B$.

I tecnici dell’azienda hanno stabilito che la causa di quel calo di resa fosse dovuto all’interazione di due lavorazioni.

5.4.2 I risultati dell’analisi esaustiva

Prima di poter applicare l’analisi esaustiva si è proceduto all’applicazione dei criteri di unicità descritti nel Teorema 2. L’impiego diretto del controllo del rango avrebbe...
5.4. CONFRONTO SU CASI REALI

<table>
<thead>
<tr>
<th></th>
<th>(O^1)</th>
<th>(O^2)</th>
<th>(O^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processo 1</td>
<td>328 m sec</td>
<td>657 m sec</td>
<td>22.21 sec</td>
</tr>
<tr>
<td>Processo 2</td>
<td>766 m sec</td>
<td>4.18 sec</td>
<td>190.89 sec</td>
</tr>
</tbody>
</table>

Tabella 5.3: Tempi di calcolo per all’applicazione dei criteri di unicità

<table>
<thead>
<tr>
<th></th>
<th>(O^0)</th>
<th>(O^1)</th>
<th>(O^2)</th>
<th>(O^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processo 1</td>
<td>785 × 54</td>
<td>758 × 54</td>
<td>730 × 54</td>
<td>721 × 54</td>
</tr>
<tr>
<td>Processo 2</td>
<td>1883 × 470</td>
<td>1875 × 470</td>
<td>1875 × 470</td>
<td>1875 × 470</td>
</tr>
</tbody>
</table>

Tabella 5.4: Riduzione delle tabelle dovute all’applicazione dei teoremi di unicità

richiesto tempi di attesa proibitivi. Per tanto sono stati applicati i criteri ridotti descritti nell’Appendice 7.2 che, come mostrato dai dati riportati in Tabella 5.3, richiedono tempi di calcolo del tutto ragionevoli.

In Tabella 5.4 sono riportate le dimensioni delle matrici di processo ottenute dopo i passi di riduzione (con \(O^0\) si indica la matrice di processo iniziale). Questi dati evidenziano un aspetto interessante: mentre l’applicazione dei criteri di unicità relativa al primo processo di lavorazione ha portato ad una sensibile riduzione delle dimensioni della matrice di processo, nel secondo caso questo non è avvenuto. La motivazione di questa discrepanza potrebbe risiedere nella differente numerosità di lotti. Il secondo caso in esame presentando un elevato numero di lotti renda improbabile l’esistenza di righe duplicate all’interno della matrice di processo (corrispondente a più lavorazioni eseguite sul medesimo insieme di lotti). Le stesse considerazioni possono essere estese anche al caso di ricerca di righe in combinazione lineare.

Dopo aver effettuato le necessarie riduzioni sulla matrice di processo si espongono i risultati dell’analisi esaustiva applicata ai due casi reali esaminati. Le due trattazioni saranno considerate separatamente.

**Il primo processo**

Dopo aver garantito l’appartenenza della matrice di processo alla classe \(O^1\), è stata implementata la ricerca in \(Y^1\). L’algoritmo ha evidenziato come potenziale colpevole sulla catena di lavorazione l’operazione ‘1001’ eseguita dalla macchina ‘ATLLL1’.

Il parametro \(y\) associato a questa lavorazione è stato stimato ad un valore pari a circa
0.9125 che corrisponde ad una probabilità di corretto funzionamento pari a 
\[ \tilde{\eta} = \exp(-\tilde{y}) = \exp(-0.9125) \cong 0.4015 \]

Ciò che conta ai fini dell’analisi svolta non è tanto il valore del parametro ottenuto (poco significativo soprattutto per N piccolo) quanto a quale step di processo esso sia associato.

Al fine di ottenere una misura dell’errore medio relativo alle rese è necessario calcolare il residuo rispetto alle rese stesse. La somma dei quadrati dei residui si può scrivere nella forma:
\[ \varepsilon' \varepsilon = (R_A - \exp(-T_A \tilde{y}))' (R_A - \exp(-T_A \tilde{y})) \cong 3.83 \]

che corrisponde ad un Errore Quadratico Medio pari a 
\[ \sqrt{\frac{\varepsilon' \varepsilon}{N_l}} = \sqrt{\frac{3.83}{54}} \approx 0.26 \]

Effettuando la ricerca in \( Y^1 \) (una sola lavorazione “colpevole”) risulta molto difficile descrivere con precisione le rese dei lotti che escono da un impianto produttivo così complesso avendo a disposizione un solo parametro libero. Nonostante ciò l’indicazione della lavorazione responsabile concorda perfettamente con quella rilevata sulla linea di produzione dai tecnici dell’azienda. I tempi di calcolo relativi all’analisi esaustiva di primo livello si sono rilevati molto brevi. Per effettuare una ricerca in \( Y^1 \) sono stati impiegati circa quattro decimi di secondo.

**Il secondo processo**

Seguendo l’algoritmo della ricerca esaustiva è stata effettuata l’analisi insiemi dimensionalmente crescenti a partire ovviamente da \( Y^1 \). Nella Tabella 5.5 vengono riportati i risultati dell’analisi in \( Y^1, Y^2 \) ed \( Y^3 \) (per problemi di riservatezza aziendale non è stato possibile riportare i nomi e i numeri reali delle lavorazioni).

La ricerca sia in \( Y^1 \) sia in \( Y^2 \) non ha portato a risultati in linea con la reale situazione dell’impianto produttivo. La ricerca in \( Y^3 \) ha evidenziato, tra le tre lavorazioni identificate, una delle reali responsabili dell’abbattimento della resa dell’impianto (l’operazione ‘1111’ eseguita dalla macchina ‘ATHDC4’), mentre la
5.4. CONFRONTO SU CASI REALI

<table>
<thead>
<tr>
<th>lavorazione 1</th>
<th>Y¹</th>
<th>Y²</th>
<th>Y³</th>
</tr>
</thead>
<tbody>
<tr>
<td>1801 ATDH4</td>
<td>7141 ATSMW1</td>
<td>1111 ATHDC4</td>
<td></td>
</tr>
<tr>
<td>lavorazione 2</td>
<td>-</td>
<td>1801 ATDH4</td>
<td>7141 ATSMW1</td>
</tr>
<tr>
<td>lavorazione 3</td>
<td>-</td>
<td>-</td>
<td>1801 ATDH4</td>
</tr>
<tr>
<td>(\varepsilon)' (\varepsilon)₀</td>
<td>0.7398</td>
<td>0.6779</td>
<td>0.6328</td>
</tr>
<tr>
<td>tempo di calcolo</td>
<td>22.2 sec</td>
<td>115 sec</td>
<td>16 ore circa</td>
</tr>
</tbody>
</table>

Tabella 5.5: Risultati dell’analisi sul secondo processo riferita alla resa per la classe 22 utilizzando un processore a 4 GHz.

seconda lavorazione (l’operazione '2222' eseguita dalla macchina 'ATF33') non è stata individuata.

Contrariamente ai risultati relativi al primo processo studiato, l’analisi esaustiva applicata a questo processo si è rilevata decisamente più complessa. Diverse sono le cause di tale complessità: il numero di fallimenti riscontrato durante la classificazione, l’elevato quantitativo di lavorazioni coinvolte, il numero di macchine ritenute realmente responsabili. Sebbene il numero di lotti in esame fosse alto, che equivale a disporre di una ingente quantità di informazione sul processo, questo semplice fatto potrebbe essere non sufficiente a garantire una semplice risoluzione del problema.

Alla luce dei risultati ottenuti e di queste ultime considerazioni è possibile affermare che l’algoritmo ha comunque portato ad un buon risultato sebbene i tempi di calcolo dell’analisi esaustiva in \(Y³\), come ci si aspettava, si sono rilevati molto onerosi: le triplette di parametri di cui è stata calcolata la stima \(LS\) sono circa \(\frac{1800^3}{6} = 972 \cdot 10^6\). Pertanto, sebbene una analisi in uno spazio dimensionalmente maggiore potrebbe portare all’individuazione di entrambe le lavorazioni incriminate, essa risulterebbe proibitiva per tempi di calcolo. Infatti l’esecuzione di una analisi in \(Y⁴\) richiederebbe la risoluzione di approssimativamente \(\frac{1800^4}{24} = 437.4 \cdot 10^9\) problemi di stima \(LS\).

5.4.3 I risultati della stepwise regression

Il metodo della stepwise regression fornisce soluzioni sub ottimali rispetto alla ricerca esaustiva, quindi se consiglia l’impiego solo quando, per motivi computazionali, non
Figura 5.5: Andamento dell’FPE in funzione del numero di regressori del modello identificati mediante stepwise regression

Sia possibile utilizzare il primo approccio. Sul primo processo produttivo non è stato necessario implementare la stepwise regression in quanto la ricerca in $Y^1$ è riuscita a determinare correttamente la lavorazione responsabile del calo ri resa in esame.

Per il secondo processo studiato, a causa della struttura più complessa del problema, è stato possibile aggiungere un numero elevato di regressori prima che la curva $FPE$ presentasse un’andamento crescente come mostrato in Figura 5.5

Il valore in corrispondenza del quale si verifica il minimo della curva indica il numero di lavorazioni che possono essere considerate a resa minore di uno per la cifra di merito utilizzata. Fra queste lavorazioni compaiono entrambe le macchine responsabili del guasto in esame.

Confrontando il risultato ottenuto dalla stepwise regression con quello ricavato dall’analisi esaustiva è possibile osservare come i primi tre regressori inseriti nel modello sono relati alle stesse tre lavorazioni evidenziate dall’analisi esaustiva nell’insieme $Y^3$ delle soluzioni. Alla luce di queste considerazioni, sebbene il risultato di un’analisi esaustiva risulta sicuramente più completo, la procedura stepwise, pur essendo una metodologia d’approccio subottima rispetto all’analisi esaustiva, ha portato al medesimo risultato ma in tempi decisamente ridotti. Infatti la ricerca in $Y^3$ tramite l’approccio stepwise richiede poco meno di un minuto d’elaborazione.

La ricerca non necessita della prova di tutte le possibili combinazioni di regressori (coppie, triplette e così via) bensì della sola valutazione del comportamento della
cifra di merito associata ad una particolare configurazione di regressori considerando l’ingresso nel modello di un parametro per volta. Come già detto Il grosso limite di tale approccio risiede nella sua subottimalità.

5.4.4 I risultati dell’approccio semi-bayesiano

Consideriamo in primo luogo le analisi effettuate sul primo processo produttivo. Fissato il valore di $\gamma$ ed aumentando opportunamente il valore di $\rho$ è stato possibile far sì che la soluzione appartenesse all’insieme $Y^1$ delle soluzioni fissando $\rho = 10$ e $\gamma = 1$. Lo step di processo individuato è rappresentato dall’operazione ’1001’ eseguita dalla macchina ’ATLLL1’. Il risultato è perfettamente in linea con quello ottenuto dall’implementazione dell’analisi esaustiva. Il tempo in cui è stato ottenuto questo risultato è di 350 sec circa. In questo caso l’approccio semi-bayesiano ha evidenziato il reale colpevole dell’abbattimento della resa della linea di fabbricazione nel primo caso esaminato con un tempo decisamente superiore a quello impiegato dagli altri algoritmi.

Imponendo $\rho = 10$ e $\gamma = 3$ è stato possibile ottenere soluzioni nell’insieme $Y^3$. L’implementazione di questa tipologia di approccio ha richiesto un tempo molto più contenuto rispetto all’analisi esaustiva: la ricerca è stata effettuata in circa 2 ore e ha condotto al medesimo risultato dell’analisi esaustiva. Le tre lavorazioni evidenziate dall’algoritmo identificano la medesima tripletta ottenuta dall’analisi esaustiva; di conseguenza ancora una volta viene evidenziata una delle due lavorazioni che hanno generato effettivamente il guasto durante il processo di fabbricazione ma in tempi decisamente ridotti.

Così come nell’implementazione dell’analisi esaustiva, le caratteristiche associate al secondo processo industriale studiato hanno generato difficoltà sia nell’individuazione dei reali responsabili sia nei tempi di calcolo sia sebbene questi si siano rilevati decisamente più contenuti rispetto all’analisi esaustiva.

In conclusione, l’approccio semi-bayesiano, mediante un’opportuna taratura dei suoi parametri, appare in grado di replicare i risultati dell’analisi esaustiva e della stepwise regression. Tuttavia è anche in grado di fornire soluzioni con più gradi di libertà per quanto rimanga aperto il problema della taratura dei parametri nel caso più generale.
5.4.5 Approccio Data Mining

Nei due problemi in esame le classifiche ottenute mediante l’approccio basato sulle tabelle di contingenza riportavano le lavorazioni "incriminate" nei primi posti. Per la precisione lo step di processo responsabile del primo caso in esame occupava il vertice della lista, mentre le due lavorazioni legate al difetto analizzato nel secondo caso si attestavano in seconda ed ottava posizione.

I tempi di calcolo delle analisi sono stati in ognuno dei due casi inferiori al minuto. Questa apparente discrepanza è legata al tipo di implementazione utilizzata per il metodo proposto. Non presentando particolari complessità computazionali si è deciso di realizzare un applicativo basato su query *SQL* [44]. La realizzazione consente di trattare problemi di diversa complessità in tempi molto simili e legato solamente al tipo di data base utilizzato. I tempi di calcolo si riferiscono alla realizzazione mediante un database access installato su di un processore intel centrino ad 1.4 GHz.

Per questo tipo di approccio sono disponibili altri risultati. Infatti, il metodo è stato testato su diverse decine di casi reali verificatisi nello stabilimento di Agrate della ST Microelectronics. I risultati di queste analisi non sono riportati sia per motivi di privacy industriale che per brevità. Comunque è possibile asserire che, per quei casi in cui è stata possibile una verifica diretta, la o le lavorazioni incriminati erano sempre presenti nei primi posti della lista prodotta dal metodo.

5.4.6 Il risultato del confronto

Gli algoritmi proposti sono stati ideati in tempi diversi. Il primo ad essere stato sviluppato è il metodo basato sulle tabelle di contingenza. Per tanto questo metodo ha goduto di una lunga fase di testing, dovuta anche allo sviluppo di un software di diagnosi. In questa fase il metodo ha fornito brillanti risultati, in molti casi superiori alle aspettative, viste le limitazioni del tipo di tecnica.

Gli altri metodi proposti in questa sede, sebbene ognuno abbia degli aspetti ancora da approfondire, hanno comunque fornito risultati interessanti, comparabili con quelli forniti dall’analisi delle tabelle di contingenza.
Capitolo 6

Il software ACID

La prova più forte contro una teoria è la sua applicabilità

Karl Kraus

La metodologia descritta in questa sede è stata sviluppata in collaborazione con la multinazionale STMicroelectronics. L’azienda ha trovato le soluzioni proposte interessanti al punto da finanziare la sviluppo di un prototipo o test vehicle con cui sondarne l’effettiva efficacia.

Il buon esito delle prove effettuate su diversi casi reali e l’interesse mostrato dalle persone che hanno utilizzato il test vehicle, ha spinto il reparto di R&D (Ricerca e Sviluppo) della sede di Agrate a sviluppare, in stretta collaborazione con l’autore, un software di diagnosi da integrare con i sistemi aziendali. Nasce così ACID (Automatic Classification for Interactive Diagnosis).

Lo sviluppo del software ACID ha presentato diverse difficoltà, sia dal punto di vista progettuale che realizzativo. In questa sede descriveremo brevemente l’architettura e le principali funzionalità del tool.
6.1 Le tecniche scelte

La metodologia proposta si fonda su due passi automatici: quello di classificazione (AC) e quello di diagnosi interattiva (ID). La procedura di diagnosi viene detta interattiva poiché l’utente, non soddisfatto dei risultato dell’analisi, potrebbe desiderare di modificarne alcuni parametri, giungendo, per affinamenti successivi, al risultato richiesto. Per ognuna delle due fasi principali previste dalla metodologia AC/ID, si è dovuto scegliere l’algoritmo da implementare nel software.

Per la fase di identificazione del guasto si è mostrato nel Capitolo 4 come la miglior scelta sia fornita dall’algoritmo SOM con mappe binarie.

Per la diagnosi di processo si è utilizzato l’approccio basato sul Data Mining Statistico. Questa scelta si basa su due motivazioni. La prima è legata ad un fattore di prudenza: infatti i metodi model based allo stato attuale non sono stati sufficientemente testati. La seconda è rappresentata dal fatto che le tabelle di contingenza forniscono un’informazione facilmente interpretabile dall’utenza, rispetto ad un vettore di rendimenti la cui estrapolazione può risultare a volte oscura.

6.2 La struttura del Software

Nel progetto dell’architettura software si sono posti i seguenti obiettivi:

- Integrare il tool con il sistema informativo di ST Microelectronics
- Progettare un’architettura non vincolata all’impianto
- Fornire uno strumento pratico e veloce per gli utenti finali
- Scegliere una realizzazione vincolata alla metodologia ma che consenta una sostituzione agevole degli algoritmi

Ognuno dei summenzionati punti è stato considerato nel progetto dell’architettura software.

Il sistema informativo di un’azienda produttrice di semiconduttori contiene un’enorme mole di informazioni. I dati sono così complessi ed articolati che comunemente ogni sede della stessa azienda possiede diverse basi di dati, ognuna dedicata
6.2. LA STRUTTURA DEL SOFTWARE

ad un differente aspetto della produzione. Inoltre i dati meno recenti spesso sono archiviati in files. Il tool richiede, per effettuare un’analisi, dati provenienti da due reparti diversi e distribuiti su un discreto intervallo temporale. Per tanto in molti casi il recupero di questi dati non è agevole, sia perché parte delle informazioni sono già state archiviate, sia perché necessita di elaborate query per essere estratto. Per recuperare in maniera efficiente i dati si è deciso di dotare il tool di un database proprio in cui raccogliere le informazioni relative alla storia lotti ed alle mappe EWS binarie. Questa base di dati privata viene aggiornata giornalmente mediante caricatori appositamente scritti in maniera totalmente trasparente all’utente.

Per rendere il software meno sensibile sia alle macchine che lo ospitano sia all’impianto di produzione, si deciso di dividere il tool in due parti distinte organizzate in una architettura client-server. Il server realizza la raccolta dati mentre il client si occupa della loro elaborazione. Qualora si volesse realizzare una nuova istanza di ACID all’interno dello stesso impianto sarà sufficiente eseguire un nuova installazione del solo lato client; se invece si volesse analizzare la produzione di un differente sito produttivo, basterà installare solo il lato server. Inoltre il database privato costituisce un vero e proprio middleware [45] fra le basi di dati ed il tool; pertanto una installazione in un nuovo sito di produzione comporterebbe solo la riscrittura dei caricatori e non dell’intero lato server.

Nonostante la presenza di un database privato, spesso l’intervallo temporale richiesto è tale da dover richiedere diverse ore solo per l’estrazione dei dati. Per non vincolare l’utente a lunghe attese al monitor si è deciso di dividere, anche sotto il profilo procedurale, l’analisi in due fasi: la prima dedicata al recupero delle informazioni, la seconda alla loro elaborazione. Durante la prima fase, l’utente seleziona il tipo di prodotto, il periodo su cui vuol effettuare l’analisi e fornisce diverse opzioni sul numero di guasti attesi. Dopo di che torna al proprio lavoro mentre la richiesta viene inoltrata al server che la schedula e reinvierà le storie lotti e le classificazioni richieste al client. Questa procedura operativa consente una migliore gestione dei tempi morti e la possibilità di lavorare il locale durante la fase di diagnosi. Inoltre, qualora l’utente lo richiedesse, è possibile archiviare presso il client sia l’analisi effettuata sia i dati inviati dal server. Lo stoccaggio in locale di quest’informazione consente di effettuare analisi successive sullo stesso set di dati senza tempi attesa.
Per garantire una rapida sostituzione degli algoritmi si è deciso di progettare sia il lato client che il lato server con una tecnica modulare orientata agli oggetti [46]. La sostituzione di un algoritmo comporterà la semplice riscrittura del relativo oggetto o modulo a seconda del tipo di sostituzione prevista.

L’architettura così delineata viene schematizzata in Figura 6.1.

### 6.3 Le funzionalità aggiuntive

Oltre alla procedura di diagnosi AC/ID descritta in questa sede, il software realizza alcune funzionalità aggiuntive che semplificano la diagnosi e facilitano il lavoro dell’utente. Molte di queste aggiunte si devono a richieste degli utenti effettuate
6.3. **LE FUNZIONALITÀ AGGIUNTIVE**

6.3.1 **Gli strumenti per la selezione del calo di resa**

Nella maggior parte delle analisi effettuate durante il periodo di test, la descrizione fornita dai pattern identificati dalle mappe di Kohonen si è rivelata sufficiente a descrivere correttamente il difetto in esame. Tuttavia, in qualche sporadico caso la selezione si è rivelata ardua. In *ACID* vengono forniti due criteri di scelta per agevolare questa selezione basati su eventuali conoscenze a priori possedute dall’utente.

Qualora chi effettua l’analisi conoscesse qualche lotto fortemente indiziato del problema in esame, la presenza o assenza di wafer appartenenti ai summenzionati lotti potrebbe essere illuminante nella selezione della classe. In *ACID* quest’informazione viene riportata in un grafico come quello mostrato in Figura 6.2. Lo stesso grafico potrebbe essere utilizzato per verificare l’accuratezza della selezione quando si conoscesse l’entità in termini di wafer del problema in esame.

Spesso si ha un’indicazione temporale riguardo il manifestarsi del problema in esame. Una classificazione corretta dovrebbe presentare wafer testati in date posteriori a questa data. In Figura 6.3 viene mostrato il grafico fornito da *ACID* atto a questo scopo.

Se mediante gli strumenti proposti non fosse possibile isolare il difetto consider-
ato, potrebbe essere consigliabile riclassificare le mappe EWS impiegando un mag- 
gior numero di classi.

6.3.2 I filtri di processo

Il metodo proposto a volte produce una classifica composta da molte lavorazioni. 
Nonostante queste siano ordinate per valori decrescenti di correlazione con la pro-
duzione di bad wafer, a volte potrebbe essere utile scremare alcuni passi di processo. 
Per tanto il tool mette a disposizione dei filtri che evitano il calcolo delle tabelle di 
contingenza per quelle lavorazioni che:

sono fuori da un determinato range di operazione Le operazioni sono se-
quenziali e spesso chi effettua l’analisi potrebbe conoscere in che fasi della 
produzione si presume sia il malfunzionamento

vengono effettuate da alcune attrezzature Alcune attrezzature potrebbero ess- 
sere del tutto aliene alla produzione di difetti in generale (macchine di misura 
non invasive), o del difetto considerato per via della simmetria fisica della loro 
lavorazione.

processano pochi bad wafer Se si ritenesse che il difetto sia imputabile ad una 
sola lavorazione o legato a più passi di processo, ognuno dei quali ha processato 
una larga parte dei bad wafer.
Mediante l’applicazione di questi filtri generalmente si riesce ridurre la classifica proposta a meno di una decina di lavorazioni.

6.3.3 La valutazione dell’analisi

Come precedentemente accennato, valutare la correttezza di una diagnosi di processo non è sempre facile. La prova definitiva si ha rimuovendo le lavorazioni maggiormente incriminate dalla catena di produzione ed osservando il risultato della produzione. Purtroppo non sempre questo è possibile.

Un buon indice di colpevolezza è dato dal raffronto delle produzione di bad wafer delle singole attrezzature che compiono la stessa lavorazione. Se la distribuzione dei bad wafer si mantiene bassa in tutte le attrezzature che svolgono l’operazione tranne che in quella descritta dalla lavorazione in esame, abbiamo un’ulteriore conferma della bontà della diagnosi. In Figura 6.4 viene riportato un grafico fornito da ACID che illustra la situazione descritta; i nomi delle attrezzature non sono stati riportati per motivi di riservatezza aziendale.
6.4 I risultati ottenuti

Attualmente il software ACID è in uso presso lo stabilimento di Agrate come strumento per la diagnosi di processo.

La validità dell’approccio proposto è stata riconosciuta anche dalle altre due sedi italiane dell’ST. La sede di Castelletto ne ha richiesto un’installazione che è attualmente in corso. La sede di Catania ne sta ufficialmente valutando la possibilità.

Inoltre è stata ufficialmente considerata la possibilità di impiegare la metodologia AC/ID a livello corporate, che permetterebbe la distribuzione del software in ogni sede della multinazionale.
Conclusioni

Il lavoro presentato in questa sede coinvolge due diversi aspetti: uno prettamente teorico, volto allo sviluppo di nuove tecniche per la diagnosi di processo, l’altro legato alla realizzazione industriale delle soluzioni proposte.

Per quanto riguarda l’aspetto teorico, a cui è stata data maggior rilevanza in questa sede, è stata sviluppata una nuova metodologia per la diagnosi di processo nell’ambito della produzione di dispositivi a semiconduttore. La metodologia presentata prevede due fasi automatizzate, la prima, denominata AC (Automatic Classification), è dedicata all’identificazione dei cali di resa sistematici mentre la seconda, denominata ID (Interactive Diagnosis), realizza la diagnosi vera e propria.

Per la fase di AC si sono presentati diversi algoritmi di clustering in grado eseguire l’identificazione dei guasti. Per ognuno dei metodi presentati è stata individuata una taratura dei parametri automatica. I metodi sono stati fra loro confrontati sulla base di un indice appositamente creato, denominato indice F. Dal confronto, effettuato su dati simulati ottenuti mediante un modello proposto in questa sede, è emerso che l’algoritmo SOM fornisce le migliori prestazioni per questo tipo di problema.

Per la fase di ID sono stati studiati quattro differenti algoritmi basati su tre diversi approcci. Gli algoritmi proposti sono stati sviluppati in tempi differenti. Per l’algoritmo basato sulle tabelle di contingenza si sono potute effettuare numerose analisi reali, in cui il metodo ha ottenuto risultati eccellenti. Le analisi preliminari effettuate sugli altri metodi hanno mostrato comunque buoni risultati.

Parte dei risultati presentati in questa sede sono stati pubblicati in:

Conclusione


In ambito industriale si è realizzato un prototipo, con cui diffondere e testare la metodologia AC/ID. L’esito positivo di questa fase ha convinto la multinazionale STMicroelectronics a sviluppare ed a rendere operativo in almeno due siti industriali un software di diagnosi basato sulla metodologia proposta.

Il progetto del software e parte della sua realizzazione ha costituito parte integrante del lavoro di dottorato affrontato. La diffusione all’interno di più sedi della multinazionale dimostra la bontà del risultato ottenuto.
Capitolo 7

Appendici

7.1 Derivazione dell’algoritmo EM

Per derivare l’algoritmo EM bisogna definire la log- verosimiglianza. Utilizzando il modello definito al Paragrafo 3.4.1 la log-verosimiglianza è definita dalla seguente

\[
l(T, \vartheta) = \ln(P(T \mid \vartheta)) = \ln(P(Z, Z^n \mid \vartheta)) = \\
= \ln(P(Z \mid \vartheta)) + \ln(P(Z^n \mid \vartheta)) = \\
= \sum_{u=1}^{N_u} \sum_{c=1}^{N_c} r^{w}(c) \sum_{d=1}^{N_d} [x^{w}(d) \ln(p^c(d)) + (1 - x^{w})(d) \ln(1 - p^c(d))] + \\
+ \sum_{u=1}^{N_u} \sum_{c=1}^{N_c} r^{w}(c) \sum_{c=1}^{N_c} r^{w}(c) \ln(\pi(c)) = \\
= \sum_{u=1}^{N_u} \sum_{c=1}^{N_c} r^{w}(c) \cdot \\
\cdot \left( \ln(\pi(c)) + \sum_{d=1}^{N_d} \{x^{w}(d) \ln(p^c(d)) + (1 - x^{w})(d)ln(1 - p^c(d)) \} \right)
\]

Possiamo ora ricavare i due passi dell’algoritmo richiesto.

Expectation

In questo passo dell’algoritmo il vettore dei parametri \( \vartheta \) e dei dati \( X \) è supposto noto. Il nuovo valore delle competenze \( Z \) viene ottenuto calcolando il valore atteso
della log-verosimiglianza.

\[ E \! \left[ l(T, \vartheta) \mid X, \vartheta \right] = \sum_{u=1}^{N_u} \sum_{c=1}^{N_c} E \! \left[ r^u(c) \mid X, \vartheta \right] \cdot \left( \ln(\pi(c)) + \sum_{d=1}^{N_d} \left\{ x^u(d) \ln(p^c(d)) + (1 - x^u(d))\ln(1 - p^c(d)) \right\} \right) \]

Ricordando che le variabili ausiliari sono variabili bernuolliane indipendenti, possiamo asserire che

\[ E \! \left[ r^u(c) \mid X, \vartheta \right] = P(r^u(c) = 1 \mid X, \vartheta) \]

Applicando il teorema di Bayes si ottiene

\[ P(r^u(c) = 1 \mid X, \vartheta) = \frac{P(X \mid r^u(c) = 1, \vartheta)P(r^u(c) = 1 \mid \vartheta)}{P(X \mid \vartheta)} \]

dove

\[ P(X \mid r^u(c) = 1, \vartheta) = \prod_{d=1}^{N_d} \left\{ x^k(d)p^c(d) + (1 - x^k(d))(1 - p^c(d)) \right\} \]

\[ P(r^u(c) = 1 \mid \vartheta) = \pi(c) \]

\[ P(X \mid \vartheta) = \sum_{c} \pi(c) \prod_{d=1}^{N_d} \left\{ x^k(d)p^c(d) + (1 - x^k(d))(1 - p^c(d)) \right\} \]

Definendo le competenze come

\[ \overline{r^u(c)} := E \! \left[ r^u(c) \mid X, \vartheta \right] \]

tsì ha che:

\[ \overline{r^u(c)} = \frac{\pi(c) \prod_{d=1}^{N_d} \left\{ x^k(d)p^c(d) + (1 - x^k(d))(1 - p^c(d)) \right\}}{\sum_{c} \pi(c) \prod_{d=1}^{N_d} \left\{ x^k(d)p^c(d) + (1 - x^k(d))(1 - p^c(d)) \right\}} \quad (7.1) \]

**Maximization**

In questa fase dell’algoritmo si suppone di conoscere l’intero insieme dei dati totali \( T \). Il nuovo valore dei parametri \( \tilde{\vartheta} \) si ottiene massimizzando il valore atteso della
log-verosimiglianza nel rispetto dei vincoli del modello. Viene richiesto di risolvere
il seguente problema di ottimizzazione vincolato:

$$\tilde{\vartheta} = \arg \max_{\vartheta} E \left[ l(\vartheta, T) \mid T \right]$$

(7.2)

soggetto a

$$\sum_{c=1}^{N_c} \pi(c) = 1$$

(7.3)

dove

$$E \left[ l(\vartheta, T) \mid T \right] = \sum_{w=1}^{N_w} \sum_{c=1}^{N_c} r^{w}(c) \cdot$$

$$\cdot \left( \ln(\pi(c)) + \sum_{d=1}^{N_d} \{ x^{w}(d) \ln(p^c(d)) + (1 - x^{w})(d) \ln(1 - p^c(d)) \} \right)$$

Il problema (7.2)-(7.3) può essere trasformato in un problema di ottimizzazione non
vincolata introducendo la lagrangiana $J(\vartheta, \lambda)$ [40]:

$$\tilde{\vartheta} = \arg \max_{\vartheta, \lambda} J(\vartheta, \lambda)$$

dove

$$J(\vartheta, \lambda) := E \left[ l(\vartheta, T) \mid T \right] + \lambda \left( \sum_{c=1}^{N_c} \pi(c) - 1 \right)$$

La soluzione del problema non vincolato è da ricercarsi nei punti stazionati di $J(\vartheta, \lambda)$.

Le derivate della lagrangiana sono:

$$\frac{\partial J}{\partial p^c(d)} = \sum_{w} r^{w}(c) \left( \frac{x^{k}(d)}{p^c(d)} - \frac{1 - x^{k}(d)}{1 - p^c(d)} \right) = \sum_{w} \left( \frac{r^{w}(c)x^{k}(d) - (p^c(d))}{p^c(d)(1 - p^c(d))} \right)$$

$$\frac{\partial J}{\partial \pi(c)} = \lambda + \sum_{w} \frac{r^{w}(c)}{\pi(c)}$$

$$\frac{\partial J}{\partial \lambda} = \sum_{c=1}^{N_c} \pi(c) - 1$$

Uguagliandole a zero si ottiene il seguente sistema di equazioni:

$$0 = \sum_{w} \left( \frac{r^{w}(c)x^{k}(d) - (p^c(d))}{p^c(d)(1 - p^c(d))} \right) \Rightarrow \sum_{w} \left( \frac{r^{w}(c)x^{k}(d) - (p^c(d))}{p^c(d)(1 - p^c(d))} \right) = 0$$

(7.4)

$$0 = \lambda + \sum_{w} \frac{r^{w}(c)}{\pi(c)} = 0 \Rightarrow \pi(c) = -\frac{\sum_{w} r^{w}(c)}{\lambda}$$

(7.5)

$$1 = \sum_{c=1}^{N_c} \pi(c)$$

(7.6)
Dalla (7.4) si ottiene:

\[ \sum_w r^{\hat{w}}(c)x^k(d) - \sum_w r^w p^\hat{w}(d) = 0 \]

da cui

\[ p^\hat{w}(d) = \frac{\sum_w r^{\hat{w}}(c)x^k(d)}{\sum_w r^w} \]  \hspace{1cm} (7.7)

Sostituendo la (7.5) in (7.6) si ottiene:

\[ 1 = -\sum_{c=1}^{N_c} \frac{\sum_w r^{\hat{w}}(c)}{\lambda} \Rightarrow \lambda = -\sum_{c=1}^{N_c} \sum_w r^{\hat{w}}(c) \]

da cui

\[ \hat{\pi}(c) = \frac{\sum_w r^{\hat{w}}(c)}{\sum_{c=1}^{N_c} \sum_w r^{\hat{w}}(c)} = \frac{\sum_w r^{\hat{w}}(c)}{N_w} \]  \hspace{1cm} (7.8)

L’algoritmo EM si basa sull’iterazione delle (7.1), (7.7) e (7.8).
7.2 Utilizzo pratico dei criteri di esistenza ed unicità

Nel Paragrafo 5.2.3 sono stati introdotti alcuni teoremi che dimostrano come l’appartenenza della matrice di processo ad alcuni insiemi di matrici $O^N$, descritti dalla Definizione 4, garantisca l’esistenza e l’unicità della soluzione del problema di minimo vincolato (5.18). Purtroppo la verifica delle condizioni imposte dai suddetti teoremi richiede il calcolo di un numero elevato di sottomatrici rappresentando di fatto un requisito molto forte anche per valori di $N$ e $M$ non necessariamente elevati.

Per garantire l’unicità della soluzione, almeno per piccoli valori di $N$, in questo lavoro sono stati utilizzati metodi di clustering sulle righe e altri metodi di riduzione della matrice che hanno permesso di evitare problemi di indistinguibilità.

7.2.1 Verifica di appartenenza ad $O^1$

Una matrice $T$ appartiene alla classe $O^1$ se il rango di tutte le sottomatrici composte da due righe $T$ è massimo. Al fine di garantire l’appartenenza della matrice di processo alla classe $O^1$ è quindi sufficiente controllare che non esistano righe uguali. Questo si traduce in $(N_m)(N_m - 1)/2$ controlli di uguaglianza su vettori ad $N_l$ dimensioni, quindi $N_l(N_m)(N_m - 1)/2$ controlli di uguaglianza. Per analisi complesse la tecnica potrebbe richiedere tempi elevati.

Considerando la natura binaria della matrice $T$ è facile dimostrare che l’elemento $Q_{i,j}$ della matrice

$$Q = TT' + (1 - T)(1 - T)'$$

indicherà il numero di elementi coincidenti fra la righe $i$ e $j$ della tabella $T$. Ottenuta la matrice $Q$ il numero di controlli da effettuare si riduce di un fattore $N_l$.

7.2.2 Verifica di appartenenza ad $O^2$ ed $O^3$

Per garantire l’appartenenza a classi superiori, in particolare $O^2$ e $O^3$ ci si è serviti di un metodo che ha permesso di ricercare le righe in combinazione lineare solamente su uno spazio ridotto della matrice $T$ di processo sfruttando il vincolo di positività del modello (si ricorda che i valori $y_i$ sono valori maggiori o al più uguali a zero).
Una matrice rettangolare con più righe che colonne ha rango massimo se nessuna delle proprie righe e in combinazione lineare con un qualunque sottoinsieme delle restanti [47].

La condizione viene violata se esiste una riga $i^*$ ed un insieme di righe $\{i^1, i^2, ..., i^p\}$ tale che valga la seguente:

$$T(i^*) = \alpha_1 T(i^1) + \alpha_2 T(i^2) + \alpha_3 T(i^3) + ... + \alpha_p T(i^p) \quad \text{con} \quad \alpha_i \in \mathbb{R}$$

Dalla dimostrazione del Teorema 2 si evince come i valori $\alpha_i$ siano dati dai valori $y_i$ ottenuti dalla trasformazione (5.10). Per questo motivo il vincolo di dipendenza lineare va considerato solo per i valori $\alpha_i$ positivi in quanto è nostro intento eliminare le righe $T(i^*)$ col numero maggiore di uni:

$$T(i^*) = \alpha_1 T(i^1) + \alpha_2 T(i^2) + \alpha_3 T(i^3) + ... + \alpha_p T(i^p) \quad \text{con} \quad \alpha_i > 0 \quad (7.9)$$

L’elemento ‘0’ in un qualsiasi posto di una qualunque riga $T(i^*)$ della matrice di processo può essere ottenuto o imponendo tutti gli $\alpha_i$ a zero (condizione mai verificata in quanto genererebbe una riga di soli zeri) oppure come somma pesata di righe che hanno lo ’0’ nello stessa posizione all’interno della riga. Pertanto la condizione di rango massima va controllata solo per quelle selezioni di righe che presentano zeri nelle stesse colonne. Questa considerazione ha permesso di semplificare notevolmente l’individuazione delle righe in combinazione lineare.

### 7.2.3 Riduzione della tabella

Qualora, analizzando una tabella di processo $T$, si riscontrassero o due righe uguali o una riga in combinazione lineare con altre si rende neccessario un metodo che consenta di svolgere l’analisi.

Se si riscontrassero delle righe uguali verrebbero rimosse dalla tabella di processo lasciandone solo una per tipo. Ovviamente, per non perdere il dato corrispondente alle righe cancellate, gli indici relativi alle corrispondenti lavorazioni debbono essere memorizzati. Questo equivale a costituire una macrolavorazione che tiene conto di tutti gli step di processo che sono stati eseguiti sullo stesso sottoinsieme di lotti. Se l’analisi indicasse come colpevole una macro lavorazione il metodo fornisce l’elenco delle lavorazioni ad essa associata informando l’utente che non è in grado di discernere fra esse.
Le modalità operative, sebbene più complesse, sono analoghe per quanto riguarda le classi d’indistinguibilità $O^N$. Per la classe $O^2$ vengono individuate e rimosse dalla matrice di processo tutte le righe pari alla somma di altre due. Anche in questo caso, dopo aver effettuato la ricerca delle lavorazioni colpevoli, sarà necessario verificare se la somma di tutte le coppie di righe associate alle lavorazioni individuate non corrisponda ad una riga che è stata precedentemente eliminata. In tal caso il metodo fornirà come possibili alternative o le due soluzioni effettivamente trovate, o la lavorazione rimossa.

Per garantire invece l’appartenenza a classi superiori ad $O^2$ è necessario il calcolo del rango delle sottomatrici; anche in questo caso viene eliminata una delle righe appartenenti alla classe d’indistinguibilità (in particolare viene eliminata la riga associata alla lavorazione che ha lavorato il maggior numero di lotti in quanto il medesimo risultato in termini di resa sui lotti può essere ottenuto dagli step di processo che lavorano ciascuno su un set di lotti ridotto).
7.3 Catalogazione di dati reali

Il problema cruciale nella valutazione di una catalogazione di dati reali, è che, a differenza di quanto accade con i dati simulati, non c'è la possibilità di sfruttare la conoscenza a priori della classificazione corretta. Questo si traduce nell'impossibilità del calcolo dei wafer misclassificati e, dunque, dell'utilizzo dell'indice \( F \), per testare la bontà della classificazione.

Diventa, quindi, necessario utilizzare nella classificazione dei dati reali, indici che non necessitino di questa informazione. In letteratura sono stati proposti indici con tali caratteristiche: l'indice Davies-Bouldin (\( DB \)), l'indice Dunn (\( D \)), l'indice Calinski Harabasz (\( CH \)) e l'indice \( I \) [48].

Va sottolineato che gli indici sopra citati si basano su caratteristiche della classificazione che, per i nostri scopi, potrebbero non risultare particolarmente rilevanti. Nello specifico tali indici giudicano positivamente una classificazione, che massimizza la distanza intra-cluster e minimizza le distanze tra i dati all'interno del cluster. In sostanza, questa tipologia di indici mira ad ottenere clusters il più possibile disgiunti, e contemporaneamente compatti [49].

Va ricordato che alla base della valutazione dell'indice \( F \), vi sia la necessità di trovare classi omogenee, piuttosto che disgiunte; quindi più che la separazione dei cluster, siamo interessati alla loro compattezza. Alla luce di queste considerazioni, si è pensato di aggiungere al confronto la log-verosimiglianza, che diventa un indice molto interessante fornendo un valore che descrive quanto i wafers si avvicinino ai patterns di riferimento della classe a cui vengono associati.

7.3.1 Descrizione degli indici

Vengono ora riassunte le principali caratteristiche degli indici in esame.

**Indice di Davies-Bouldin (\( DB \))**

L'indice \( DB \) [50] è funzione del rapporto tra le massime distanze intra-cluster, cioè tra i vari dati appartenenti al cluster e il centroide, e la somma delle distanze tra i cluster, ovvero tra la distanza tra i centroidi. Dalle considerazioni precedenti, risulta ovvio che la classificazione è tanto migliore quanto è minore il valore dell'indice \( DB \).
Indicando con \( c \) il \( c \)-esimo cluster e con \( I_c \) l’insieme degli indici dei dati appartenenti al cluster \( c \), la distanza intra-cluster, \( S_c \), è data dall’espressione:

\[
S_c = \frac{1}{|I_c|} \sum_{w \in I_c} \| x^w - p^c \|
\]

dove \( p^c \) rappresenta il centroide dell’\( c \)-esimo cluster, \( x^w \) il generico dato appartenente al cluster e \( |I_c| \) la cardinalità dell’insieme \( I_c \). La distanza tra i cluster \( i \) e \( j \) è definita come

\[
d_{i,j} = \| p^i - p^j \|
\]

L’indice \( DB \) è definito dalla seguente espressione:

\[
DB = \frac{1}{|N_c|} \sum_{i=1}^{N_c} R_i
\]

dove \( R_i = \max_{j,j \neq i} \frac{S_i + S_j}{d_{i,j}} \).

**Indice di Dunn \((D)\)**

In questo indice la distanza viene calcolata con il diametro della classe, definito come segue [51]. Sia \( d(x, y) \) la distanza tra i punti \( x \) e \( y \) dello spazio considerato. Allora il diametro di un cluster non vuoto \( c \), denominato \( \Delta \), è definito come:

\[
\Delta(c) = \max_{k,h \in I_c} d(x^k, y^h)
\]

La distanza inter-cluster \( d \) tra i cluster \( i \) e \( j \) è definita come

\[
\delta(i, j) = \min_{k \in I_i, h \in I_h} d(x^k, y^h)
\]

Per una qualunque partizione, Dunn definisce il seguente indice:

\[
D = \min_{1 \leq c \leq N_c} \left( \min_{1 \leq j \leq N_c, j \neq c} \frac{\delta(i, j)}{\max_{1 \leq c \leq N_c} \Delta(c)} \right)
\]

Valori elevati dell’indice \( D \) corrispondono a ”buoni” clusters, e la partizione che massimizza \( D \) può considerarsi ottimale.
Indice Calinski Harabasz (CH)

Questo indice, per $N_w$ dati e $N_c$, clusters viene calcolato utilizzando il rapporto delle tracce di due matrici di dispersione, $B$ e $W$ [52], rispettivamente tra cluster e intra-cluster:

$$CH = \frac{\text{trace}B/(N_c - 1)}{\text{trace}W/(N_w - N_c)}$$

La traccia della matrice di dispersione tra i cluster, $B$, indicando con con $p$ il centroide dell’intero set di dati, può essere scritta come

$$\text{trace}B = \sum_{c=1}^{N_c} |I_c| \cdot \|p^c - p\|^2.$$

La traccia della matrice di dispersione intra-cluster, $W$, può essere scritta come

$$\text{trace}W = \sum_{c=1}^{N_c} \sum_{w \in I_c} \|x^w - p^c\|^2.$$

In definitiva l’indice CH viene calcolato secondo la seguente espressione:

$$CH = \left[\frac{\sum_{c=1}^{N_c} |I_c| \cdot \|p^c - p\|^2}{N_c - 1}\right] / \left[\frac{\sum_{c=1}^{N_c} \sum_{w \in I_c} \|x^w - p^c\|^2}{N_w - N_c}\right]$$

Anche in questo caso la partizione ottimale è quella che massimizza l’indice CH.

Indice $I$

L’indice $I$ è definito dalla seguente espressione:

$$I = \left(\frac{1}{N_c} \times \frac{E_1}{E_{N_c}} \times D_{N_c}\right)^p$$

dove la potenza $p$ è stata posta pari a 2 [48]. L’indice è composto da tre fattori principali, $\frac{1}{N_c}$, $\frac{E_1}{E_{N_c}}$ e $D_{N_c}$ che competono tra loro, e si bilanciano l’uno con l’altro, dove

$$E_K = \sum_{c=1}^{K} \sum_{w \in I_c} \|x^w - p^c\|$$

$$D_{N_c} = \max_{1 \leq i \leq N_c; 1 \leq j \leq N_c} \|p^i - p^j\|$$
7.3.1. CATALOGAZIONE DI DATI REALI

Il fattore $\frac{1}{Nc}$ riduce l’indice al crescere del numero di classi. Il fattore $\frac{E_1}{ENc}$ è il rapporto tra $E_1$, che è costante per un dato set di dati, ed $ENc$, che decresce con l’aumentare del numero di clusters; quindi il fattore spinge per ottenere la formazione di molti cluster compatti. Infine, il fattore $D_{Nc}$ misura la massima distanza tra i cluster, su tutte le possibili coppie di centroidi. La classificazione corretta tende a far crescere il valore dell’indice $I$.

Verosimiglianza media per die $AL$

La verosimiglianza è stata introdotta trattando l’algoritmo $EM$ nel Paragrafo 3.4.2. Per motivi di leggibilità verrà qui considerata la verosimiglianza media per die (dispositivo), definita dalla seguente espressione:

$$AL = \exp \left( \frac{1}{N_dN_w} \sum_{w=1}^{N_w} \sum_{c=1}^{N_c} \sum_{d=1}^{N_d} \left\{ x^w(d) \ln(p^c(d)) + (1 - x^w(d)) \ln(1 - p^c(d)) \right\} \right)$$

dove al solito $N_w$, $N_c$, $N_d$ sono rispettivamente il numero di wafer, classi e dispositivi, mentre $x^w(d)$ indica il risultato binario del test $EWS$ sul dispositivo $d$ del wafer $w$ e $p^c(d)$ indica il pattern di riferimento.

7.3.2 Definizione dell’esperimento

Per effettuare tale confronto si è prima pensato a come creare il set di dati su cui eseguire il test. Si vogliono ottenere alcune classificazioni si cui testare il grado di correlazione fra l’indice $F$ ed i vari indici proposti.

Fissato il set di dati $s$ composto da $N_w$ wafer, il risultato di una classificazione viene indicato dal vettore $g_s(w) = [ g_s(1) \ldots g_s(N_w^s) ]$ dove $g(w)$ indica la classe in cui è stato catalogato il wafer $w$. Partendo dalla classificazione vera $g_s^*$, una classificazione di test $g'_s$ può essere ottenuta estraendo in maniera un numero casuale $M$ di wafer ed assegnando a ciascuno di essi ad una classe diversa da quella vera. Si ottiene, in questo modo, una classificazione ”errata” sulla quale effettuare il calcolo dell’indice $F$ come degli altri cinque indici. Si estrarranno casualmente più classificazioni ”errate” ottenute da tutti i 12 set di dati simulati.

L’esperimento viene quindi definito dal seguente algoritmo:

1. Per ogni set di dati $s = 1, \ldots, 12$, 

a) Per $t = 1, \ldots, 25$,
   
i. si pone $g_s^t = g_s^*$
   
ii. si pone $M = \text{rand}(N_w^*)$
   
iii. per $w = 1, \ldots, M$ si pone $g_s^t(\text{rand}(N_w^*)) = \text{rand}(12)$
   
iv. Si calcolano gli indici reali per $g_s^t$
   
v. Si calcola l’indice $F$ per $g_s^t$ e $g_s^*$
   

   dove con $\text{rand}(K)$ si definisce l’estrazione casuale di un numero intero compreso fra 1 e $K$.

7.3.3 Confronto fra gli indici

I grafici di Figura 7.1 mostrano i risultati del confronto tra i vari indici descritti, verosimiglianza ($AL$), Davies-Bouldin ($DB$), Dunn ($D$), Calinski Harabasz ($CH$) e $I$, e l’indice $F$. Appare evidente come la migliore correlazione risulti tra l’indice $F$ e l’indice $CH$, ma buoni risultati arrivano anche dal confronto con la $AL$ e con l’indice $DB$. In questi tre grafici, infatti, si possono notare i quattro andamenti, per quattro diversi benchmarks, distinti l’uno dall’altro. Lo stesso non si può dire nei confronti tra l’indice $F$ e gli indici $D$ e $I$: risulta basso, in questi due grafici, il grado di correlazione tra gli indici.

In Tabella 7.1 sono riassunti i valori dei coefficienti di correlazione per i vari algoritmi per i dodici set di dati in esame. I valori più alti, in valore assoluto, si riscontrano tra l’indice $F$ e gli indici: $CH$, con un indice di correlazione medio pari a -0.99118, $AL$, con un indice di correlazione medio pari a -0.99092, e $DB$, con un indice di correlazione medio pari a 0.98851. Minore è risultata la correlazione tra l’indice $F$ e l’indice $I$ (correlazione media pari a -0.89391), e $D$ (correlazione media pari a -0.56323).
Figura 7.1: Confronto fra le valutazioni degli indici proposti con quelle dell’indice $F$
<table>
<thead>
<tr>
<th>Set</th>
<th>$AL$</th>
<th>$DB$</th>
<th>$D$</th>
<th>$CH$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.98615</td>
<td>0.98163</td>
<td>-0.69396</td>
<td>-0.98369</td>
<td>-0.90632</td>
</tr>
<tr>
<td>2</td>
<td>-0.99273</td>
<td>0.99413</td>
<td>-0.62637</td>
<td>-0.99180</td>
<td>-0.89611</td>
</tr>
<tr>
<td>3</td>
<td>-0.99458</td>
<td>0.99326</td>
<td>-0.77624</td>
<td>-0.99325</td>
<td>-0.91208</td>
</tr>
<tr>
<td>4</td>
<td>-0.99465</td>
<td>0.99549</td>
<td>-0.68879</td>
<td>-0.99384</td>
<td>-0.89003</td>
</tr>
<tr>
<td>5</td>
<td>-0.98064</td>
<td>0.97936</td>
<td>-0.57651</td>
<td>-0.98821</td>
<td>-0.86694</td>
</tr>
<tr>
<td>6</td>
<td>-0.99295</td>
<td>0.99252</td>
<td>-0.28766</td>
<td>-0.99303</td>
<td>-0.89808</td>
</tr>
<tr>
<td>7</td>
<td>-0.99175</td>
<td>0.98100</td>
<td>-0.25622</td>
<td>-0.99155</td>
<td>-0.89747</td>
</tr>
<tr>
<td>8</td>
<td>-0.99343</td>
<td>0.99364</td>
<td>-0.62026</td>
<td>-0.99221</td>
<td>-0.87866</td>
</tr>
<tr>
<td>9</td>
<td>-0.99171</td>
<td>0.98000</td>
<td>-0.70487</td>
<td>-0.99204</td>
<td>-0.90624</td>
</tr>
<tr>
<td>10</td>
<td>-0.99002</td>
<td>0.98348</td>
<td>-0.56974</td>
<td>-0.98873</td>
<td>-0.89609</td>
</tr>
<tr>
<td>11</td>
<td>-0.99434</td>
<td>0.99289</td>
<td>-0.40962</td>
<td>-0.99271</td>
<td>-0.87182</td>
</tr>
<tr>
<td>12</td>
<td>-0.99414</td>
<td>0.99472</td>
<td>-0.56879</td>
<td>-0.99308</td>
<td>-0.90708</td>
</tr>
</tbody>
</table>

Tabella 7.1: Coefficienti di correlazione tra l’indice $F$ e gli indici proposti al variare dei set di dati simulati da cui vengono estratti dati
Bibliografia


Ringraziamenti

Dopo anni di scuola, università e dottorato di ricerca è finalmente Finita, per quanto possa essere difficile, adesso non ci sono più scuse: non sono più uno studente. Ora mi attende un mondo nuovo, sconosciuto fatto di lavoro sodo, ricco di riunioni, di cose che non si possono fare e, temo, sottopagato…. Tutto sommato forse non è così alieno… Ironia a parte questi tre anni sono stati, almeno nel mio caso, un degno completamento alla mia formazione accademica. Se il mio percorso è stato tanto fortunato lo debo a molte persone.

Innanzitutto debbo ringraziare, il Prof. Riccardo Scattolini, per non aver insistito troppo, per avermi dato l’opportunità di accedere al dottorato e di avermi dato la possibilità di realizzare un sogno.

Parimenti importante è stato il mio tutore, il Prof. Giuseppe De Nicolao, per avermi aiutato ed insegnato molto sia sul punto di vista professionale che umano.

Un ringraziamento sentito va ai miei compagni di viaggio in ST, in particolare il dott. Guido Miraglia e l’ing. Oliver Donelli, senza di voi ACID non sarebbe mai uscito da un portatile ed io non conoscerrei ne i vini piemontesi, ne l’arte del’imburraggio.

Un grazie sentito va a tutti i miei coautori, per aver creduto in me, e per avermi mantenuto vigile su diversi fronti della ricerca.

Ai miei studenti ed i miei tesisti vanno i miei ringraziamenti sia per avermi mostrato errori e pregi del mio modo di insegnare che per le soddisfazioni che hanno condiviso con me.

A tutte le persone che, a diverso titolo, hanno popolato e popolano il laboratorio di Identificazione e controllo dei sistemi dinami ci; la vostra amicizia ed affetto sono state indispensabili.

L’ambito lavorativo non è il solo ad essere stato importante in questo percorso.

Alla mia famiglia va il ringraziamento più grande, per l’appoggio, la non sempre facile comprensione delle mie scelte e delle mie difficoltà e per avermi concesso la libertà di investire nei miei sogni. Grazie!

Alla Compagnia del pesante fardello tutta, debbo le ore di svago e di divertimento indispensabili alla mia sanità mentale nonché l’esperienza necessaria per una sana e fruttuosa comprensione dell’Ucas.

Come non ringraziare colei che mi è rimasta accanto in quasi tutto questo percorso subendo i miei racconti entusiastici e le mie lamentele, mostrandomi i miei pregi ed i miei difetti; Grazie.

Infine a tutte le persone che hanno letto questa pagina, e magari qualche d’una delle precedenti, va il mio ringraziamento e le mie scuse se vi avessi annoiato.