Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica

TESI DI LAUREA

Studio di metodi di apprendimento automatico per la predizione del tipo tumorale

Laureando:
Mattia Donami

Relatore:
Prof. Fabio Vandin

Anno accademico 2015/2016

Fortuna Favet Fortibus

Sommario

Il cancro è una malattia genetica molto complessa, la cui comparsa è da attribuire a determinate mutazioni genetiche indesiderate. Saper riconoscere queste mutazioni genetiche potrebbe essere utile per l'identificazione e la prevenzione dello sviluppo tumorale negli individui. Ancor più utile sarebbe il poter identificare lo sviluppo tumorale a partire da un numero limitato di mutazioni. In questo lavoro si è cercato di identificare e utilizzare tali mutazioni per creare un modello predittivo del tipo tumorale mediante tecniche di machine learning che sia in grado di effettuare diagnosi tumorali accurate per nuovi pazienti da classificare. Per selezionare mutazioni genomiche rilevanti, sono stati sviluppati diversi metodi in grado di analizzare le informazioni mutageniche presenti nel corredo genetico di 3554 pazienti suddivisi secondo 11 tipi tumorali differenti, provenienti dal progetto The Cancer Genome Atlas. Il primo metodo sfrutta le informazioni funzionali della cellula per focalizzare l'attenzione su interazioni geniche potenzialmente influenzate da insorgenze tumorali. Dall'analisi sperimentale si è constatato che il primo metodo non porta a miglioramenti in termini di accuratezza per la predizione del tipo tumorale. Il secondo metodo ricerca mutazioni genomiche frequenti solamente in pazienti affetti da un determinato tipo tumorale. Dall'analisi sperimentale per il secondo metodo si è constatato che i risultati sono stati positivi: si è ottenuto un insieme ristretto di 29 geni che contiene principalmente geni la cui associazione con la malattia è nota. Essi permettono di ottenere un modello predittivo per il tipo tumorale senza perdita significativa di accuratezza rispetto all'utilizzo di tutte le mutazioni.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Da quanto riportato nella più recente relazione stesa dall'American Cancer Society in collaborazione con l'Agency of Research on Cancer sono stati stimati 14,1 milioni di nuovi casi di cancro nel 2015 e 2 milioni di decessi causati dal cancro nel 2012. Si stima inoltre che nel 2030 saranno diagnosticati 21.7 milioni di nuovi casi con un numero di decessi pari a 13 milioni[2]

Il cancro è una patologia genetica causata da mutazioni del DNA che vengono acquisite spontaneamente o dovute ad insulti ambientali. I fattori esterni o ambientali possono essere l'uso di tabacco, la dieta, le infezioni virali o l'esposizione ad agenti chimici. Inoltre esistono mutazioni ereditarie da uno o entrambi i genitori che se non comportano la comparsa inevitabile della malattia, ne aumentano il rischio di insorgenza. Queste mutazioni vengono geneticamente mantenute nelle cellule figlie derivanti dalla divisione cellulare e vengono selezionate positivamente nel caso in cui tali mutazioni comportino un vantaggio nella crescita o nella sopravvivenza cellulare. Le mutazioni posso essere classificate in: sostituzioni di una singolo nucleotide (single base substitution, SBS), amplificazioni o delezioni all'interno del gene e traslocazioni di porzioni tra geni. Nei tumori solidi che colpiscono colon, seno, cervello o pancreas, il 95% delle mutazioni note sono rappresentate da single base sobstitutions, mentre solo il 5% è costituito da piccole inserzioni o delezioni di poche basi nucleotidiche.
Sono state individuate due tipi di mutazione: driver mutations, che possono innescare la trasformazione tumorale, e passenger mutations, più numerose rispetto alle prime, che sono causate dall'instabilità genomica della cellula tumorale. Ad oggi sono stati identificati circa 140 geni di tipo driver in grado di promuovere il processo di tumorogenesi. In generale un tumore contiene da due a otto mutazioni driver[10]. Le rimanenti mutazioni presenti sono invece passenger mutations. Le passenger mutations quindi non causano di per sè l'insorgenza del tumore, ma sono importanti per conferire un vantaggio selettivo nello sviluppo tumorale come l'insorgenza della farmacoresistenza. L'accumulo di queste mutazioni dà origine a delle proprietà che caratterizzano il tumore e sono generalmente correlate all'omeostasi cellulare: le cellule tumorali possono acquisire un'aumentata attività proliferativa, una scarsa o assente risposta a stimoli di morte cellulare (apoptotici o necrotici) ed avere una variazione nel mantenimento del proprio genoma. Questi sono i tre processi cellulari principali regolati dalle dodici cascate di segnale in cui sono state classificate le mutazioni driver note fino ad oggi. Le cascate di segnale (comunemente

chiamate "signalling pathways") sono una serie di "azioni" concatenate tra molecole biologiche (come ad esempio proteine, lipidi, o ioni) che causano l'attivazione di una risposta cellulare.
Il tumore è quindi una patologia altamente complessa in quanto può colpire diversi tessuti, essere causata da mutazioni multiple che portano alla trasformazione cellulare e dar luogo ad una progressione di malattia altamente variabile. Anche la diagnosi della malattia può avvenire con diverse modalità ed attuando differenti metodologie, dal prelievo di una porzione di tessuto, alla citofluorimetria a flusso o all'analisi di proteine rilasciate dal tumore. Queste tecniche permettono di diagnosticare e classificare il tumore. Il tumore è tuttavia causato dall'accumulo di alterazioni genomiche. Per questo il sequenziamento del genoma nelle cellule tumorali è divenuto da anni fondamentale per la diagnosi precoce e per la scelta della terapia farmacologica per il suo trattamento. Negli ultimi decenni la caratterizzazione del genoma è stata effettuata mediante l'utilizzo di microarrays a DNA o RNA (piccole sonde di DNA o RNA attaccate ad una superficie solida, denominata chip) o tramite il first-generation sequencing, noto anche come metodo di Sanger. La limitazione di queste due metodologie è dovuta da un lato al numero limitato di sequenziamenti possibile e dall'altro al conseguente costo eccessivo[9]. Con l'introduzione della next geration sequencing è invece possibile eseguire una mappatura genica di diversi tipi tumorali in un ingente numero di pazienti. Questa metodologia permette di aumentare notevolmente il numero di campionamenti, cosa necessaria di fronte all'eterogeneità della patologia. Il sequenziamento dell'intero genoma tumorale ha raggiunto un costo esiguo di alcune migliaia di dollari, rispetto ai 14 milioni del 2006. Studi recenti hanno dimostrato che alcuni tumori contengono poche mutazioni, mentre altri contengono decine di migliaia di mutazioni, come il cancro alla pelle o al polmone. Con l'applicazione della next generation sequencing si è reso indispensabile lo sviluppo di algoritmi necessari all'analisi e alla classificazione di questa mole di dati.
Grazie alla quantità di dati a disposizione si potrebbe rilevare un gruppo di mutazioni genetiche che favoriscono lo sviluppo di determinati tipi tumorali. Conoscere queste informazioni sarebbe un vantaggio per la diagnosi tumorale, in quanto si potrebbe potenzialmente anticipare lo sviluppo del cancro in un individuo dalla semplice osservazione delle sue eventuali mutazioni genetiche all'interno di questo gruppo. In questo contesto risulta utile l'impiego di tecniche di machine learning, in grado di imparare a riconoscere automaticamente relazioni importanti tra i dati e in seguito utilizzarle per prendere decisioni intelligenti su nuovi dati da classificare.
In questo studio si è deciso di verificare le potenzialità delle tecniche di machine learning applicate sulle informazioni mutagene di una serie di pazienti suddivisi secondo tumori differenti, con lo scopo di creare un modello che sia capace di effettuare delle predizioni tumorali accurate su nuovi pazienti da analizzare. Considerare allo stesso tempo pazienti affetti da tumori differenti ha permesso di poter valutare con più attenzione le mutazioni genetiche esclusive ad un solo tipo tumorale, in quanto la scelta di mutazioni genetiche in comune tra più tumori distinti potrebbe confondere la predizione effettuata del modello.
In base al tipo di dati da analizzare, sono stati utilizzati due principali metodi molto popolari nel campo dell'apprendimento supervised: SVM e Random Forest (Capitolo 2). Per rinforzare tali tecniche di machine learning sono stati sviluppati diversi algoritmi in grado di ottenere un insieme di geni che siano potenzialmente associabili allo sviluppo di un tipo tumorale specifico. L'utilizzo di questo insieme, assieme ai metodi SVM e

Random Forest, dovrebbe migliorare le predizioni effettuate dal modello, in quanto si limiterebbe il numero di geni da osservare rispetto alla totalità dell'interno corredo genetico di un individuo (all'incirca 25000 geni). Sono stati definiti due algoritmi che sfruttano congiuntamente le informazioni mutagene e funzionali della cellula, espresse mediante interazioni tra geni, per scoprire cluster genici (funzioni specifiche di una cellula) potenzialmente influenzabili da diverse tipologie tumorali (Capitolo 3). L'ultimo algoritmo, implementato mediante due paradigmi differenti (Greedy e ILP), analizza le sole informazioni mutagene per scoprire un insieme di geni che presentino frequenti mutazioni solo in presenza di pazienti affetti da un determinato tipo tumorale (Capitolo 4). Tutti gli algoritmi sono stati ideati con lo scopo di associare ad ogni tipo tumorale un insieme di geni potenzialmente esclusivo. In questa maniera è possibile ottenere un modello che sia in grado di eseguire predizioni tumorali soddisfacenti.

Capitolo 2

Classificazione con SVM e Random Forest

In questo capitolo vengono introdotti i principi di funzionamento dei metodi di machine learning utilizzati nel corso della tesi, assieme alla descrizione delle informazioni tumorali prese in esame per questo studio. La sezione 2.1 presenta una descrizione in termini logici e matematici del modello SVM. Nella sezione 2.2 vengono introdotte le fasi principali per la creazione e utilizzo di una Random Forest. La caratterizzazione dei dati avviene nella sezione 2.3. La sezione 2.4 descrive le prime analisi effettuate mediante SVM e Random Forest sulle informazioni tumorali.

2.1 Definizione del modello SVM

Nel campo del machine learning, Support Vector Machine(SVM)[4] è un metodo di classificazione lineare di tipo supervised introdotto da Vladimir Vapnik nel 1963. Scopo del SVM è quello di scegliere un modello mediante l'analisi di un training set, definito secondo un insieme di dati caratterizzati da una serie di attributi e da una classe di appartenza, la quale può assumere solamente due valori distinti. L'allenamento serve per rilevare delle corrispondenze logiche tra determinati valori degli attributi e la classe di appartenenza assegnata ad essi. Una volta completata la fase di training, il modello deve essere in grado di predire correttamente la classe di appartenenza per nuovi dati dove tale valore risulta sconosciuto, in base alle corrispondenze scoperte in precedenza.
Uno dei migliori pregi del metodo SVM è quello di ottenere buoni risultati senza dover incorrere a problemi computazionali o di overfitting. L'overfitting è un problema che si verifica quando un modello di classificazione risulta più complesso di quello che realmente occorre per una corretta classificazione, comportando ad una scarsa predizione per i nuovi dati da classificare. Oltre alla complessità di un modello, la sua presenza varia anche in base alla quantità e qualità dei dati: per training set di dimensione piccoli e/o in presenza di dati affetti da rumore (errori di misurazione), la frequenza dell'overfitting tende ad aumentare. Fortunatamente SVM possiede delle proprietà che permettono di limitare tale problema: esso è robusto al rumore presente nei dati e in aggiunta può utilizzare metodi di regolarizzazione che permettono di limitare l'eccessiva complessità che può raggiungere il modello durante la fase di allenamento. Grazie a queste caratteristiche, SVM rappresenta una valida scelta per la classificazione dei dati.

L'idea che sta alla base del metodo è quella di rappresentare ogni istanza del training set in un punto di uno spazio d-dimensionale mediante la definizione di un vettore, dove d corrisponde al numero di attributi che descrivono i dati. A seguire si cerca di trovare un iperpiano in grado di separare correttamente tutti i punti secondo la loro classe di appartenenza, suddividendo così lo spazio in due aree. Per valutare la classe di nuove istanze, basterà trasformarle in vettori d-dimensionali e poi determinare in quale area dello spazio esse verranno collocate.
Tuttavia alcuni punti potrebbero essere soggetti a rumore, dove in questo caso viene definito come uno sfasamento nello spazio per il punto rispetto alla sua posizione originale. Il problema nasce quando un punto si trova nei pressi dell'iperpiano: per via del rumore esso potrebbe oltrepassare l'iperpiano ed essere classificato in modo errato. Per combattere il rumore, ad entrambe le facce dell'iperpiano viene dedicato uno spazio in egual misura, nel quale i punti non possono risiedere. Esso prende il nome di *margine*, ed è definito come la distanza dall'iperpiano al punto più vicino ad esso. All'aumentare delle dimensioni del margine, il modello diventa sempre più resistente ai rumori, e risulta migliore in termini di generalizzazione per i nuovi dati da classificare. Un margine con queste caratteristiche viene detto forte, e si può ottenere solamente se tutti dati nel training set risultano separabili correttamente nello spazio d-dimensionale. Nella pratica, questo risultato è difficile da ottenere per via della natura dei dati, i quali il più delle volte risultano molto sparsi tra loro. Tuttavia esistono due soluzioni alternative per ottenere dei buoni risultati a livello di classificazione: applicare la tecnica del Kernel Trick oppure ricercare un margine debole.
La tecnica del Kernel Trick, attraverso una formula non lineare, trasforma tutti i punti d-dimensionali del training set in punti di uno spazio di dimensione maggiore, dove aumentano le probabilità di poter separare i dati attraverso un iperpiano (Figura 2.2). L'operazione permette un grosso risparmio in termini computazionali, perché il passaggio dalla dimensione inferiore a quella superiore non viene eseguito direttamente sui dati. Fortunatamente, diversi tipi di kernel sono già stati sviluppati: lineare, polinomiale e Gaussiano RBF (in ordine crescente di complessità) sono quelli che risultano più utilizzati nei vari studi. Nonostante ciò è bene osservare che non sempre il passaggio ad una dimensione superiore permette di trovare un margine forte per dividere i dati correttamente.
In molti casi l'utilizzo di un margine debole risulta essere un buon espediente: esso, a differenza di un margine forte, tollera la presenza di punti all'interno della sua area, comportando anche ad alcuni errori di classificazione per i dati del training set. In cambio, esso permette di ottenere un margine più largo rispetto ad uno forte in base alla configurazione di un parametro di regolarizzazione. Questo consente di regolare la complessità del modello ottenibile, in modo da poter ottenere buoni risultati anche in presenza di alcuni errori di classificazione.
Si può riassumere dalle considerazioni effettuate che un buon iperpiano permette di separare i due insiemi di punti nel modo migliore possibile, cercando al contempo di ottenere il massimo margine raggiungibile (Figura 2.1).

[1]Più formalmente, un training set contiene N istanze, in cui ognuna di esse è definita secondo la tupla $t_i = (\mathbf{x}_i, y_i)$, dove $\mathbf{x}_i$ è un vettore di feature in R^d e y_i corrisponde al valore della classe che l'istanza può assumere $(-1, +1)$. Tutti gli iperpiani in R^d sono

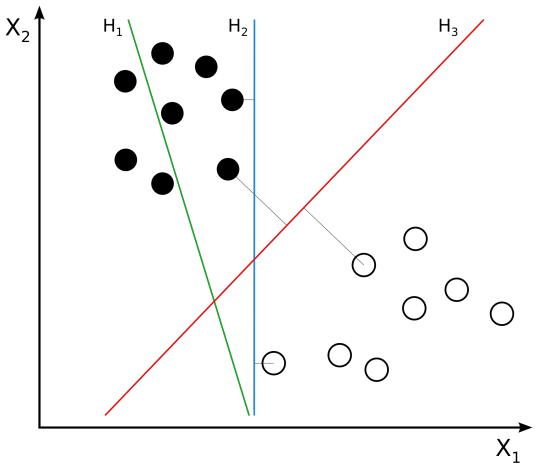

Figura 2.1: Iperpiani. H3 è il miglior iperpiano tra le soluzioni proposte: separa correttamente i dati mantenendo il massimo margine (fonte:Wikipedia.org)

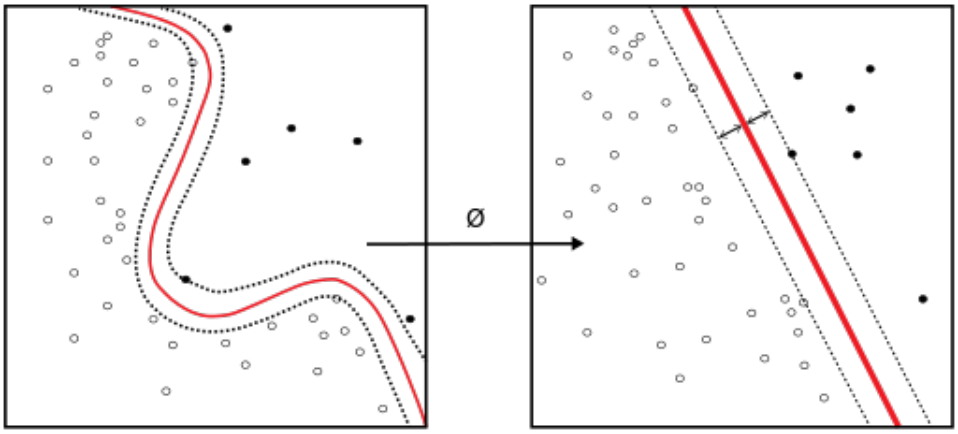

Figura 2.2: Applicazione del Kernel Trick (fonte:Wikipedia.org)

parametrizzati secondo la tupla $h = (\mathbf{w}, b)$, dove $\mathbf{w}$ è un vettore e b una costante, e sono in grado di separare i dati se per ogni $\mathbf{x}_i$ viene verificata la condizione:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 0 \tag{2.1}$$

Tuttavia, dato un iperpiano definito come $(\bar{\mathbf{w}}, \bar{b})$, sono equivalenti ad esso tutti gli iperpiani definiti dalle coppie $(\bar{\mathbf{w}}/\lambda, \bar{b}/\lambda)$ per $\lambda \in R^+$. Per qualsiasi iperpiano è possibile trovare un λ pari al minimo valore che può assumere $y_i(\mathbf{w} \cdot \mathbf{x}_i + b)$, il quale permette di modificare (2.1) in modo che per tutti i punti $\mathbf{x}_i$ la funzione sia maggiore o al più uguale a 1.
Questo tipo di iperpiano viene detto *iperpiano canonico* e soddisfa la seguente:

$$\min_{i=1,\ldots,N} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 \tag{2.2}$$

Le due formule 2.1 e 2.2, per le osservazioni effettuate in precedenza risultano equivalenti. Il limite descritto in 2.2 tuttavia non rappresenta il valore del margine, ma è solo una proprietà funzionale dell'iperpiano canonico che tornerà utile per la ricerca del

massimo margine ottenibile.
Per ottenere quindi una definizione matematica del margine viene prima introdotta la distanza che separa un iperpiano h con un punto generico $\mathbf{x}_i$:

$$dist(h, \mathbf{x}_i) = \frac{y_i(\mathbf{w}\cdot \mathbf{x}_i + b)}{\|\mathbf{w}\|} \tag{2.3}$$

Il margine, come riportato in precedenza, definisce una distanza geometrica nello spazio tra l'iperpiano e il punto più vicino ad esso, ossia la distanza minima che intercorre tra l'iperpiano e tutti i punti del training set:

$$\min_{i=1,.....,N} dist(h, \mathbf{x}_i) = \frac{1}{\|\mathbf{w}\|}\cdot \min_{i=1,.....,N} y_i(\mathbf{w}\cdot \mathbf{x}_i + b) = \frac{1}{\|\mathbf{w}\|} \tag{2.4}$$

Grazie alla 2.2, la formula 2.4 può essere semplificata.

Il margine è quindi pari a $1/\|\mathbf{w}\|$, dalla quale si intuisce che per trovare il massimo margine ottenibile bisogna minimizzare la quantità $\|\mathbf{w}\|$, sotto la condizione 2.2. Questa ricerca viene espressa attraverso un problema di ottimizzazione:

$$\begin{aligned} \underset{b,\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & \min_{i=1,.....,N} y_i(\mathbf{w}\cdot \mathbf{x}_i + b) = 1 \end{aligned} \tag{2.5}$$

Tutte le considerazioni effettuate fino a questo punto si basano sulla ricerca di un iperpiano con un margine "forte". Tuttavia si può modificare 2.5 in modo tale da ottenere un problema di ottimizzazione che sia in grado di considerare anche i margini "deboli":

$$\begin{aligned} \underset{b,\mathbf{w},\xi}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot \mathbf{x}_i + b) \geq 1 - \xi_i \qquad (i = 1, ..., N) \\ & \xi_i \geq 0 \qquad (\forall i) \end{aligned} \tag{2.6}$$

Le modifiche che sono state effettuate sono le seguenti: ora la funzione obiettivo contiene un nuovo termine da minimizzare, $C\Sigma_{i=1}^{N}\xi_i$, che raffigura la violazione complessiva sul margine che si può effettuare. Le variabili ξ_i rappresentano la quantità di violazione del margine assegnata ad ogni punto x_i, ossia quanto lontano un punto può spingersi all'interno del margine. Se il valore invece è pari a 0, il punto suddetto non può risiedere all'intero del margine.
Questa proprietà viene inclusa nei N vincoli del problema, che sostituiscono il vincolo di 2.5. Tali vincoli sono stati "alleggeriti", cambiando il segno dell'equazione precedente, in modo tale da rendere il problema più facile da risolvere in termini computazionali e algebrici.
Il parametro C è una costante, il cui valore viene assegnato a piacere e permette di modulare con la funzione $\Sigma_{i=1}^{N}\xi_i$ da minimizzare. Quando C assume valori alti, si cerca di fare più attenzione alla violazione dei margini da parte dei punti e di conseguenza alle variabili ξ_i verranno assegnati dei valori molto bassi. D'altra parte, per C tendente a valori bassi ci si preoccupa di meno riguardo tale questione e quindi le variabili ξ_i possono assumere

valori più alti. Scegliendo attentamente il valore di C, si può ottenere allo stesso tempo un iperpiano con un margine elevato e con una violazione molto bassa.
In particolare, per $C = \infty$, l'unica soluzione che permette di ridurre la funzione obiettivo è quella di impostare $\xi_i = 0(\forall i)$, ottenendo come soluzione del problema (se esiste) un iperpiano con un margine "forte".

Riassumendo, 2.6 permette di risolvere il problema del margine in modo più semplice e con un controllo sulla complessità del modello, a scapito di ottenere dei risultati non sempre ottimali. Fortunatamente, se i dati sono linearmente separabili la soluzione coincide con quella del problema 2.5.
Un'altra importante motivazione per la modifica effettuata a 2.5 si basa sulla sua risoluzione: la forma di 2.6 è riconducibile ad una famiglia di problemi conosciuti come *quadratic programming* (QP), per i quali esistono diverse tecniche per la loro risoluzione. Uno dei più usati si basa sull'utilizzo dei *moltiplicatori di Lagrange*. Grazie a questo metodo, il problema 2.6, dopo una serie di passaggi algebrici diventa:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{N} y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \qquad (\forall i) \end{aligned} \tag{2.7}$$

dove α rappresenta il vettore dei moltiplicatori di Lagrange, il quale contiene N valori non negativi.

Dalla derivazione delle equazioni in 2.7 si ottiene un risultato fondamentale per la ricerca dell'iperpiano ottimale:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i} \tag{2.8}$$

$$\alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1) = 0 \qquad (\forall i) \tag{2.9}$$

Si può osservare dalla 2.8 come il vettore che delinea l'iperpiano ottimale non è altro che una combinazione lineare tra le istanze t_i del training set e gli α_i. Non tutte le istanze del training set sono considerate valide per l'iperpiano: infatti come descritto dalla 2.9 solo i punti $\mathbf{x}_i$ per i quali vale la condizione $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$ possono assumere un valore $\alpha_i > 0$. Tutti i t_i con $\alpha_i > 0$ vengono definiti con il nome di *support vector*, e sono gli unici punti necessari per definire l'iperpiano.
Lo scopo principale del modello SVM è quindi trovare la combinazione di support vector che permetta di ottenere l'iperpiano ottimale, o una soluzione tendente all'ottimo.

2.2 Definizione del modello Random Forest

Random Forest[6]è un'altro metodo molto popolare nell'ambito del machine learning. Esso si basa sulla costruzione di una collezione di alberi decisionali, allenati su sottoinsiemi differenti dello stesso training set. Questo permette di contenere il problema dell'overfitting presente in alberi decisionali profondi, in quanto essi tendono ad essere modelli troppo complessi per la classificazione dei dati, portando anche a risultati con una varianza elevata. Il modello, una volta allenato, è in grado di predire la classe di appartenenza per nuovi dati in base alla moda calcolata sui risultati di predizione ottenuti dai vari alberi, con lo scopo di ridurre la varianza del risultato finale.
Un singolo albero decisionale (Figura 2.3) possiede le seguenti caratteristiche: ogni nodo interno rappresenta un attributo distinto, mentre gli archi verso i nodi figli rappresentano i possibili valori che tale attributo può assumere. Un nodo foglia invece rappresenta un possibile valore della classe di appartenenza. La classificazione viene rappresentata attraverso il cammino che si compierebbe dal nodo radice fino ad un nodo foglia, attraverso i valori degli attributi del dato da classificare.

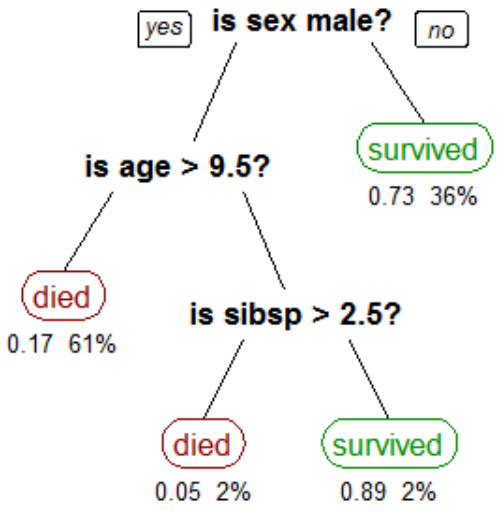

Figura 2.3: Albero di decisione: superstiti del Titanic. (fonte:Wikipedia.org)

Per creare l'insieme di alberi che definiscono la Random Forest, viene applicata la tecnica del bootstrap aggregation: ogni albero viene associato ad un insieme Z contenente una parte del training set selezionato in modo randomico e con eventuali ripetizioni tra i dati. Successivamente viene creato l'albero mediante la costruzione di un singolo nodo che identifica l'attributo in grado di dividere nel miglior modo possibile i dati in Z. L'attributo viene selezionato tra m possibili candidati, prelevati in modo randomico tra l'intero corredo di attributi che descrivono i dati contenuti in Z. Il procedimento viene iterato di volta in volta nei nodi figli, fino alla costruzione completa dell'albero.
Più in dettaglio, la tecnica del bootstrap aggregation permette di creare B alberi i.d.

(identicamente distributi ma non indipendenti tra loro) in cui ogni albero decisionale viene descritto da una varianza pari a σ^2 e presenta una correlazione ρ con i restanti $B-1$ alberi. Si ottiene quindi la seguente formula che descrive la varianza media σ^2_{RF} per il risultato finale:

$$\sigma^2_{RF} = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{2.10}$$

dove si può notare che per mantenere valori di σ^2_{RF} bassi, è necessario utilizzare valori elevati per B e allo stesso tempo tenere la correlazione ρ tra gli alberi bassa. Questo ultimo risultato viene raggiunto grazie alla selezione randomica degli m attributi durante la costruzione dell'albero di decisione.
E' stato dimostrato mediante vari esperimenti che una buona scelta per m è $\sqrt{p}$, dove p è il numero di attributi che definiscono i dati, ottenendo valori di correlazione bassi (di solito 0.05 o minori). Al contrario, il parametro B dipende dalla dimensione e dalla natura dei dati del training set, e di solito può assumere valori compresi tra le centinaia fino alle migliaia di alberi decisionali. Ovviamente all'aumentare di B, aumenta anche il tempo computazionale per costruire la Random Forest. Quindi risulta opportuno assegnare un valore a B che sia il più basso possibile e che permetta di ottenere buoni risultati in termini di classificazione.
Tale valore può essere trovato grazie alla tecnica dell'out-of-bag-error (OOB): per ogni dato $t_i = (\mathbf{x}_i, y_i)$ del training set, viene costruita una Random Forest contenente solo gli alberi allenati con un insieme Z in cui t_i non compare. Il processo viene terminato quando l'errore OOB, pari all'errore di misclassificazione che si compierebbe su nuovi dati, si stabilizza. Successivamente si può trovare il valore di B minore che minimizza l'errore OOB.

2.3 Caratterizzazione dei Dati

In questo studio vengono utilizzate informazioni relative a mutazioni genomiche presenti in pazienti affetti da diverse tipologie tumorali. Tali dati sono stati rappresentati mediante una *mutation matrix*: essa è una matrice binaria, dove righe e colonne rappresentano rispettivamente i pazienti malati e i possibili geni in cui può avvenire una mutazione, tranne la prima colonna che identifica il tipo di tumore al quale il paziente è soggetto. In particolare, una mutazione genomica per un dato paziente viene rappresentata mediante il valore 1, nella colonna del gene corrispondente. Al contrario, viene utilizzato il valore 0 per rappresentare nessun cambiamento nel gene di un paziente.
A seguire si può osservare la forma e una possibile istanza di una mutation matrix:

$$\begin{array}{c} \\ \vdots \\ paziente_{i-1} \\ paziente_i \\ paziente_{i+1} \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccccccc} Tumore & Gene_1 & Gene_2 & \dots & Gene_j & \dots & \dots & Gene_N \end{array} \\ \begin{pmatrix} \vdots & \vdots & \vdots & \dots & \vdots & \dots & \dots & \vdots \\ Tipo_A & 1 & 1 & \dots & 1 & \dots & \dots & 0 \\ Tipo_A & 0 & 0 & \dots & 1 & \dots & \dots & 1 \\ Tipo_B & 1 & 0 & \dots & 1 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \dots & \vdots \end{pmatrix} \end{array}$$

Per ottenere questo tipo di matrice, d'ora in poi chiamata matrice di Score, sono stati analizzati dati provenienti dal progetto The Cancer Genome Atlas [(`http://cbio.mskcc.org/cancergenomics/pancan_tcga/`)]. I campioni genomici contenuti in questi dati sono stati confrontati con il genoma umano di riferimento n.19 (hg19) e a seguire sono stati rilasciati il 31 Marzo 2013.
La tabella 2.1 illustra la suddivisione dei pazienti presenti nei dati in base al tipo di tumore associato.

Tabella 2.1: Dati progetto The Cancer Genome Atlas: Suddivisione pazienti secondo il tipo tumorale

Tipo Tumore	**Zona interessata**	**Numero Pazienti**
blca	Vescica	100
brca	Seno	530
coadread	Colon Rettale	513
gbm	Cerebrale	276
hnsc	Collo e Testa (Tessuto)	306
kirc	Reni	499
laml	Midollo Osseo (Leucemia)	205
luad	Polmoni (Ghiandole)	230
lusc	Polmoni (Tessuto)	183
ov	Ovaie	464
ucec	Utero	248
Totale Pazienti		3554

Analizzando gli attributi che compongono i dati, si sono osservarti diversi campi per descrivere le mutazioni genetiche di un singolo paziente. In particolare esiste un attributo chiamato "Variant Classification" che descrive la tipologia di mutazione presente nel gene. Grazie ad esso, non sono state considerate le mutazioni genetiche che fanno parte dei seguenti gruppi mutageni, perché attualmente si presume che non diano un contributo rilevante per lo sviluppo tumorale:

- *upstream;downstream, UTR5;UTR3, 5'UTR, 3'Promoter, 3'Flank, Silent, IGR, upstream, downstream, Fusion, RNA*

Si è inoltre constatato che il numero di geni distinti affetti da almeno una mutazione è pari a 24799, un numero tendente all'intero corredo genetico di un individuo (il Genoma Umano è composto da 20000-25000 geni), quindi per il momento non è possibile escluderne nessuno.
Sulla base di queste informazioni, la matrice di Score ottenuta contiene 3554 righe (pazienti) suddivise tra 11 tumori differenti, come descritto nella tabella 2.1 e ben 24799 colonne (geni). Sebbene il numero di geni considerati risulta elevato, non si può dire lo stesso del numero effettivo di mutazioni genomiche presenti nella matrice: esse coprono solo il 0.60 % all'interno di Score, mentre il numero di zeri risulta essere molto elevato, 99.4%. Si evince che la matrice di Score risulta essere molto sparsa.

In aggiunta a questa affermazione, nella tabella 2.2 vengono descritti per ogni tipo tumorale il numero medio di mutazioni genomiche e la varianza relativa: si può osservare per alcuni tumori, come ucec, una varianza molto elevata rispetto alla media relativa. Da questo se ne deduce che il numero di mutazioni genomiche per ogni tipo tumorale risulta essere anch'esso sparso.

Tabella 2.2: Media e Varianza (approssimate) del numero di mutazioni genomiche per i vari tumori presenti in *Score*

Label	**Media**	**Deviazione**
blca	206	162
brca	45	38
coadread	279	629
gbm	86	420
hnsc	140	128
kirc	56	33
laml	18	30
luad	230	220
lusc	280	225
ov	50	27
ucec	443	1188

2.4 Applicazione dei metodi e Risultati

Per analizzare la matrice di Score, in modo da trovare eventuali corrispondenze tra mutazioni genomiche e i diversi tipi tumorali considerati, sono stati utilizzati due strumenti molto popolari nell'ambito del machine learning: SVM, come implementato in Liblinear[5] (`http://www.csie.ntu.edu.tw/~cjlin/liblinear/`) e la libreria Scikit-learn (`http://scikit-learn.org/stable/`) per la classificazione attraverso Random Forest.
Liblinear è un classificatore lineare per dati contenenti milioni di instanze e attributi. E' stato sviluppato dal Machine Learning Group dell'Università Nazionale della Taiwan. Esso risolve il modello definito in 2.6 mediante diversi tipi di risolutori, i quali supportano funzioni di regolarizzazione per migliorare la classificazione, limitando allo stesso tempo il problema dell'overfitting. Inoltre, esso è in grado di gestire la classificazione multiclasse, una caratteristica fondamentale nei casi in cui la classe di appartenza contenga più di due valori distinti (come nel caso della matrice di Score).
Scikit-Learn è una libreria open-source per il linguaggio Python. Nasce come progetto presentato al Google Summer of Code del 2007, sviluppato nel corso degli anni a venire. Essa contiene molte funzioni utili per l'analisi dei dati, i quali spaziano tra tecniche di regressione, clustering e classificazione, dove in quest'ultima categoria è presente l'algoritmo Random Forest. Scikit-Learn si appoggia su altre librerie famose come Numpy, Scipy e Matplotlib, che contengono strumenti fondamentali per il calcolo scientifico e per la rappresentazione dei grafici.
Per eseguire correttamente una prova di classificazione è stato necessario dividere la matrice di Score in due sottomatrici disgiunte tra loro: Score Train e Score Validation. La matrice di Train contiene il 75% delle istanze di Score, necessarie per allenare un classificatore mediante il modello SVM o Random Forest, mentre nella matrice di Validation è presente il restante 25% dei valori e vengono utilizzati per testare il livello di generalizzazione del classificatore.
La suddivisione delle istanze di Score nelle due matrici viene eseguita in modo randomico, mantenendo inalterate le proporzioni tra il numero di presenze dei tipi tumorali presenti nella matrice d'origine. Le proporzioni appena descritte si possono osservare nella tabella 2.3. In aggiunta, tale suddivisione avviene ogni volta che viene eseguito una prova di classificazione, in modo da ottenere risultati che non dipendono da una sola configurazione delle due matrici di Train e Validation.

In base alle caratteristiche della matrice di Score, per i test di classificazione con Liblinear è stato utilizzato il risolutore "L2-regularized L1-loss support vector classification (dual)" ,chiamato d'ora in poi R2-L1, con parametro di costo C pari a 1. Esso permette di ottenere buoni risultati di classificazione in generale su dati molti sparsi, in cui il numero di attributi che definiscono i dati supera di molto il numero delle istanze. In aggiunta, questo risolutore non utilizza "feature selection", una proprietà che permette al classificatore di scartare gli attributi non necessari per le predizioni, selezionando solo quelli ritenuti importanti. Senza questa proprietà, il risolutore R2-L1 può utilizzare tutti gli attributi per eseguire le predizioni. Questo ha permesso di analizzare il comportamento di un classificatore sulla matrice di Score osservando l'intero corredo genetico a disposizione. L'iperpiano definito da SVM mediante l'utilizzo di R2-L1 (in forma primale) può

Tabella 2.3: Suddivisione numero pazienti nelle matrici di Train e Validation

Tipo Tumore	**Score**	**Train**	**Validation**
blca	100	75	25
brca	530	398	132
coadread	513	385	128
gbm	276	207	69
hnsc	306	230	76
kirc	499	374	125
laml	205	154	51
luad	230	173	57
lusc	183	137	46
ov	464	348	116
ucec	248	186	62
Totale Pazienti	3554	2667	887

essere osservato in 2.11.

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}(\max(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i)) \tag{2.11}$$

Tabella 2.4: Media su 10 prove di Liblinear eseguite sulla matrice di Score con risolutore R2-L1 e parametro costo C = 1.

	Accuratezza media (%)									
Tumore	*P.1*	*P.2*	*P.3*	*P.4*	*P.5*	*P.6*	*P.7*	*P.8*	*P.9*	*P.10*
blca	4.00	4.00	4.00	12.00	16.00	16.00	0.00	12.00	4.00	0.00
brca	42.42	40.90	41.66	50.00	42.42	44.69	49.24	35.60	37.87	53.03
coadread	85.93	83.59	86.71	90.63	86.71	82.03	85.93	79.68	81.25	91.40
gbm	60.86	50.72	50.72	59.42	59.42	57.97	60.86	65.21	56.52	53.62
hnsc	60.53	60.52	60.53	51.31	56.57	57.89	46.05	60.52	57.89	61.84
kirc	75.20	72.00	73.60	70.40	72.80	74.40	68.80	72.80	66.40	77.60
laml	68.62	74.50	64.70	54.90	72.54	64.70	64.70	72.54	72.54	74.50
luad	57.89	49.12	54.38	59.64	61.40	33.33	50.87	71.92	45.61	59.64
lusc	34.78	26.08	32.60	21.73	28.26	39.13	30.43	34.78	34.78	28.26
ov	62.06	73.27	58.62	80.17	68.10	83.62	59.48	80.17	67.24	68.96
ucec	62.90	58.06	53.22	58.06	46.77	62.90	59.67	61.29	56.45	50.00
Tutti	61.33	59.97	58.63	62.45	60.76	62.11	58.62	63.02	57.83	63.58

Nella tabella 2.4 sono stati riportati i risultati ottenuti da 10 prove di classificazione effettuati sui dati della matrice di Score, utilizzando il metodo SVM. Per ogni prova è stato rilevato il livello di generalizzazione del classificatore, chiamato d'ora in poi accuratezza. L'accuratezza rappresenta la frazione di istanze della matrice di Validation che sono state classificate in modo corretto dal classificatore. Per uno studio più approfondito, nella tabella viene riportata anche l'accuratezza calcolata per il singolo tumore: essa

permette di evidenziare la precisione del classificatore nei confronti del numero di istanze assegnate ad ogni tipo tumorale.
Complessivamente si è ottenuta un'accuratezza media pari a 60.83%, un valore abbastanza soddisfacente che permette di confermare l'esistenza di associazioni tra i tipi tumorali e una serie di mutazioni genomiche.
Per validare questo risultato esistono diversi metodi. Uno di essi si basa sul confronto dei risultati ottenuti tra due classificatori allenati sugli stessi dati, dove uno dei due utilizza una versione permutata di quest'ultimi. In questo modo, se l'accuratezza tra i due classificatori risulta simile tra loro, si può constatare che tra i dati originali non esistono corrispondenze logiche che il classificatore sia riuscito a rilevare durante la fase di training. Viceversa, se l'accuratezza ottenuta dal classificatore allenato con i dati originali supera di molto quello che utilizza i dati permutati, si può affermare che i dati originali contengono delle corrispondenze logiche e che le previsione effettuate non sono dettate dal caso.
In questa circostanza si è cercato di validare i risultati riportati nella tabella 2.4, allenando un secondo classificatore su una matrice di Score permutata. La permutazione è stata eseguita rimescolando in modo randomico le posizioni dei tipi tumorali definiti nella prima colonna della matrice di Score. Si è constatato dall'osservazione dei risultati di queste prove, riportate nella tabella 2.5, come l'accuratezza risulta essere molto bassa se relazionata con quella riportata in 2.4. Di conseguenza si è confermato che le mutazioni considerate forniscono informazione importante per predire il tipo tumorale.

Tabella 2.5: Media su 10 prove di Liblinear eseguite sulla matrice di Score permutata con risolutore R2-L1 e parametro costo C = 1.

	Accuratezza media (%)			
Label	*P.1*	*P.2*	*P.3*	*P.4*
blca	0.00	4.00	0.00	4.00
brca	12.87	18.18	24.24	15.90
coadread	23.43	21.09	25.78	18.75
gbm	8.69	0.00	2.89	2.89
hnsc	9.21	3.94	2.63	3.94
kirc	23.20	16.80	16.80	18.40
laml	7.84	9.80	1.96	1.96
luad	3.57	8.77	7.02	1.75
lusc	2.17	4.34	2.17	2.17
ov	16.37	11.20	15.51	8.62
ucec	4.83	8.06	0.00	0.00
Tutti	13.30	11.95	12.85	9.80

A seguire, dalla 2.4 si è potuto constatare la scarsa predizione in termini di accuratezza per i pazienti affetti da tumore BLCA, se confrontate con le predizioni definite per gli altri tipi tumorali. Una possibile causa potrebbe essere attribuita al numero effettivo di pazienti con tumore BLCA presenti nella matrice di Train che corrispondono solamente a 75 elementi su 2667, come descritto dalle informazioni in 2.3. Essi rappresentano un

numero abbastanza esiguo rispetto l'insieme complessivo dei pazienti che compongono la matrice e in questo caso SVM, per via dei pochi dati a disposizione, potrebbe non essere in grado di rilevare correttamente delle relazioni importanti che legano il tumore BLCA con delle mutazioni genomiche particolari.
Questo caso ha permesso di evidenziare la difficoltà per il metodo SVM nel creare un classificatore che sia in grado di fare predizioni accurate per ogni tipo tumorale, comportando ad un abbassamento generale dell'accuratezza.

I test di Random Forest sono stati eseguiti variando il numero di alberi decisionali che compongono il classificatore. Questo ha permesso di valutare il numero minimo di alberi che permetta di ottenere un'accuratezza stabile. La tabella 2.6 descrive diversi gruppi di alberi considerati per il classificatore. L'accuratezza e varianza riportate per ogni gruppo sono state calcolate sui risultati ottenuti da 10 prove di classificazione, dalle quali si è osservato una stabilizzazione dei risultati con foreste contenenti 100 o più alberi di decisione. Dai risultati ottenuti non si è osservato nessun vistoso miglioramento in termini di accuratezza rispetto al classificatore allenato con SVM.
Di conseguenza si è potuto affermare che nel caso della matrice di Score i metodi di Random Forest e SVM si eguagliano tra loro, sebbene siano definiti da due approcci di classificazione completamente differenti.

Tuttavia, grazie ai test effettuati, è stato possibile verificare l'esistenza di relazioni tra mutazioni genetiche e tumori utilizzando l'intero corredo genetico a disposizione, lasciando spazio alla possibilità di ottenere un insieme ristretto di geni che permettano ad un classificatore di raggiungere prestazioni pari o migliori a quelle ottenute in 2.4 e in 2.6. Per ottener un tale gruppo è stato necessario definire altri approcci da utilizzare sulla matrice di Score, che si differenziano dal semplice utilizzo di un metodo di classificazione standard.

Tabella 2.6: Prove Random Forest eseguite sulla matrice di Score

N.alberi	**Acc. Media**	**Var.**
10	53.14	0.015
50	58.96	0.013
100	61.87	0.019
250	60.92	0.011
500	61.35	0.007

Capitolo 3

Classificazione mediante Interaction Network

In questo capitolo si cerca di migliorare la classificazione tumorale mediante l'utilizzo congiunto delle informazioni mutageniche e dei legami funzionali che intercorrono all'interno di una cellula. Per questo scopo sono stati ideati due nuovi approcci: nella sezione 3.1 viene definito un nuovo tipo di kernel in grado di unire informazioni mutageniche e funzionali in un unica matrice. Nella sezione 3.2 viene trattato un algoritmo che permette di evidenziare per ogni tumore i potenziali legami funzionali coinvolti nello sviluppo tumorale. Le prestazioni dei due metodi vengono analizzate nella sezione 3.3.

3.1 Definizione del Kernel

Come descritto nelal sezione 2.1, il kernel rappresenta una funzione che permette di operare su spazi con dimensioni uguali o superiori di quelle definite dai dati di partenza senza dover calcolare direttamente le coordinate di quest'ultimi, ottenendo un risparmio elevato in termini computazionali. Esso viene applicato nei casi in cui risulta difficile suddividere i dati correttamente, in modo da migliorare la classificazione. Attualmente esistono già diverse funzioni di trasformazione che permettono di ottenere buoni risultati in generale. Questo non esclude la possibilità di creare nuovi kernel, in base alla tipologia dei dati da studiare.
In questo caso si è cercato di sviluppare due diversi kernel in grado di aggiungere alla matrice di Score le informazioni relative ai legami funzionali che intercorrono tra i diversi geni di un individuo. L'osservazione di legami funzionali avrebbe dei vantaggi rispetto all'analisi dei singoli geni di un individuo. Lo sviluppo del tumore è in genere causato alla variazione di un cluster genico che può provocare modificazioni funzionali nella cellula (quali ad esempio, aumento della proliferazione o diminuita risposta a stimoli di morte cellulare). L'associazione combinata di questi dati assieme a quelli descritti dalla mutation matrix potrebbe potenzialmente migliorare la classificazione tumorale in quanto le mutazioni geniche vengono associate alle diverse funzioni della cellula, in modo da rilevare geni che presentano una correlazione tra loro.
Questi legami funzionali vengono illustrati da una protein-protein interaction network, dove ogni nodo rappresenta un gene distinto e un arco una interazione tra essi. Le diverse interazioni che intercorrono tra i vari geni possono descrivere una funzione particolare di

una cellula, esprimibile nell'interaction network attraverso una distinta sottorete. La rete può essere espressa mediante una matrice di adiacenza: righe e colonne sono rappresentate dai geni della rete, mentre gli elementi interni possono assumere valori pari a 1 o 0 in base alla presenza o meno di una interazione tra due possibili geni. Una matrice di adiacenza così definita risulta simmetrica per costruzione.
Essa viene utilizzata da uno dei due kernel, in modo da unire le informazioni topologiche con quelle mutagene presenti in Score, ottenendo una nuova matrice in cui per ogni paziente vengono evidenziati i geni mutati all'interno dell'interaction network. In questa maniera un classificatore allenato con tale matrice potrebbe ottenere dei miglioramenti in termini di generalizzazione, essendo in grado di utilizzare per la predizione anche le informazioni topologiche dei geni come la posizione nella rete e le varie relazioni tra essi.
Il secondo kernel, rispetto al precedente, stabilisce una gerarchia tra le diverse interazioni che compongono la rete. In questa maniera si tenta di focalizzare l'attenzione solamente su interazioni ritenute interessanti da una determinata procedura. Per ottenere queste informazioni è necessario applicare sulla rete un algoritmo random walk con restart: selezionato il nodo di partenza g per la random walk, ad ogni iterazione essa si muove in uno dei nodi adiacenti a quello corrente g_t con probabilità $(1-\sigma)$, oppure ricomincia dal nodo g con probabilità σ, dove σ è un valore scelto a piacere in $(0 \leq \sigma \leq 1)$. La scelta del nodo successivo viene eseguita in modo randomico tra tutti i possibili candidati, ai quali viene attribuita la stessa probabilità di selezione. Se la rete è connessa, il teorema dell'ergodicità[8] assicura che la random walk eseguita su g raggiunge una situazione stazionaria , ottenendo una distribuzione di probabilità per i vari nodi visitati durante il cammino.
Tali informazioni vengono racchiuse all'interno di una matrice d'influenza (o diffusione), dove righe e colonne sono le medesime della matrice di adiacenza dell'interaction network. Il singolo elemento (i, j) indica la probabilità per il gene i-esimo di raggiungere il gene j-esimo dopo una random walk con partenza dal gene i. La matrice, per via della randomicità del processo, risulta non simmetrica sulla diagonale: la probabilità di arrivare al gene j-esimo partendo dal gene i-esimo potrebbe essere diversa per il percorso inverso.
Unire le informazioni della matrice di influenza con quelle contenute in Score dovrebbe permettere ad un classificatore allenato su tali dati di concentrare l'attenzione in gruppi di mutazioni genetiche che interagiscono molto tra di loro, sempre con il fine di ottenere miglioramenti per la classificazione di nuove istanze.
La creazione dei due Kernel descritti è permessa grazie ad un metodo matematico, descritto in modo dettagliato nell'articolo di NICK[7] (Network-Induced Classification Kernel). Questo metodo permette di includere le informazioni topologiche di una interaction network all'interno del metodo SVM come funzione di regolarizzazione. La regolarizzazione può essere gestita mediante un parametro β: valori alti inducono SVM a tenere maggior considerazione per i dati espressi dalla rete. L'iperpiano regolarizzato è rappresentato in 3.1, dove $\mathbf{A}$ corrisponde alla matrice di adiacenza della rete di dimensione (GxG)

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\beta \sum_{j=1}^{G} \left(\sum_{k=j+1}^{G} (\mathbf{A}_{j,k}(w_j - w_k)^2) \right) \tag{3.1}$$

Il nuovo modello SVM regolarizzato risulta equivalente ad un SVM classico senza tale funzione di regolarizzazione solo se si inducono le informazioni della rete all'interno dei

dati iniziali. L'intero processo può quindi essere espresso mediante la costruzione di una nuova matrice, da analizzare in seguito mediante metodi di classificazione standard. Per una comprensione più semplice, a seguire sono riportate le fasi fondamentali del processo:

1. Definizione parametri:
   - S = matrice di dimensione NxG contenenti i dati iniziali, dove N è il numero di pazienti e G il numero di geni.
   - A = matrice simmetrica non negativa di dimensione GxG contenente informazioni aggiuntive.
   - A' = matrice diagonale di dimensione GxG, dove $A'_{i,i} = \sum_{j=1}^{P} A_{i,j}$
   - I = matrice d'identità di dimensione GxG
2. Esecuzione procedura
   (i) Calcolo della matrice temporanea: $T = (I + \beta(A' - A))^{-1}$
   (ii) Decomposizione di Cholesky su T: esiste una matrice triangolare inferiore L tale che $T = LL^T$
   (iii) Calcolo risultato dell'applicazione kernel: $K = SL$

dove S corrisponde a Score e A alla matrice di adiacenza o influenza.
Per rendere quest'ultima compatibile con il kernel si è deciso di applicare diverse trasformazioni: secondo la formula $(A + A^T)/2$ è stato possibile ottenere una nuova matrice simmetrica dove l'elemento generico (i, j) e il suo corrispettivo (j, i) corrispondono alla media tra gli elementi (i, j) e (j, i) della matrice d'origine. Al contrario, l'altra trasformazione prevede la costruzione di una matrice simmetrica riportando per ogni (i, j) e (j, i) il valore massimo (o minimo) tra gli elementi presenti nella matrice d'origine aventi le stesse coordinate. In questa maniera è stato possibile rendere la matrice di influenza simmetrica senza dover effettuare grossi cambiamenti ai dati contenuti in essa.

3.2 Hotnet2

Il kernel non rappresenta l'unico approccio che permette di utilizzare l'interaction network con il fine di migliorare la ricerca di mutazioni genomiche rilevanti. Piuttosto di unire le informazioni topologiche con quelle mutagene all'interno di un'unica matrice, è possibile analizzare tali dati separatamente. L'idea è stata quella di raffinare la ricerca di cluster genici all'interno dell'interaction network che presentano più mutazioni di quante dettate dal caso, utilizzando come riferimento le informazioni descritte dalla matrice di Score.
Fortunatamente è già stato sviluppato un algoritmo chiamato Hotnet2[8], che permette di eseguire questo tipo di ricerche sulle reti. Hotnet2 utilizza un modello per la diffusione di "calore" attraverso l'intera rete in modo da valutare simultaneamente il valore delle singole mutazioni genomiche relazionate con la sottorete di appartenenza. Il "calore"

di un gene viene espresso mediante un valore numerico e rappresenta la sua frequenza di mutazione. Geni con un "calore" alto (detti nodi caldi) presentano frequenze di mutazioni elevate e sono relazionati con lo sviluppo tumorale, mentre i geni con un valore basso di "calore" (detti nodi freddi) raramente presentano delle mutazioni ma potrebbero interagire con altri geni per la crescita del tumore.
Inizialmente, viene associato ad ogni gene un determinato valore di "calore" in base a determinati criteri come la rilevanza del gene in termini biologici e/o alla sua frequenza di mutazione che intercorre su una popolazione di individui. Successivamente, il "calore" dei vari geni viene propagato ai nodi vicini mediante un processo di random walk con restart, utilizzando la stessa procedura già descritta in precedenza per la creazione della matrice d'influenza. Raggiunta la situazione stazionaria per il processo, l'algoritmo identifica come interessanti le sottoreti composte da geni che hanno dato un contribuito significativo in termini di "calore" inviato e ricevuto. Generalmente, una sottorete interessante è composta sia da nodi con "calore" elevato sia da quelli che presentano una frequenza di mutazioni minore. In tal modo è possibile osservare geni che potrebbero essere associati a diverse forme tumorali, sebbene non presentino frequenti mutazioni tra gli individui. Infine, Hotnet2 calcola il valore di p-value in base al numero di sottoreti scoperte aventi dimensioni $\geq k$, dove k è un valore arbitrario. Il p-value misura il livello di significatività dei risultati ed è definito come la probabilità di ottenere un risultato uguale o più grande di quanto attualmente osservato supponendo che non esistano associazioni tra mutazioni e la rete (ipotesi nulla). Fissando un valore di soglia α, vengono reputate significative tutte le sottoreti di dimensioni $\geq k$ se il valore di p-value $\leq \alpha$, altrimenti si può concludere che tali sottoreti non presentano nessun tipo di relazioni importanti.
Grazie ad Hotnet2, è stato possibile ricavare dei cluster genici associabili ad ogni tipo tumorale presente nella matrice di Score. L'insieme complessivo dei geni ottenuti da vari cluster è stato utilizzato per ridurre la matrice di Score, in modo da poter eseguire dei test di classificazione solo sui geni ritenuti importanti per i vari tumori da Hotnet2. Per ottenere un tale risultato si è sviluppato una procedura, descritta in Alg. 1.

3.3 Risultati

L'interaction network utilizzata per questo studio è conosciuta con il nome HINT+HI2012 (`http://compbio-research.cs.brown.edu/pancancer/hotnet2/#!/data`), una combinazione tra le informazioni contenute nella rete HINT (High-quality INTeractomes) e un insieme HI-2012 contenente iterazioni proteiche. Essa è strutturata secondo 9858 geni distinti e 40704 interazioni tra essi, dove tali informazioni sono racchiuse all'interno di una matrice di adiancenza. La matrice di influenza utilizzata deriva dalla stessa rete mediante una random walk con parametro di restart $\sigma = 0.40$. Esso rappresenta un valido valore per diffondere correttamente il calore lungo la rete in modo da ottenere dei risultati interessanti[8].
Tuttavia il numero di geni in Score (24799) non è equivalente al numero di geni presenti nelle matrici di adiacenza e influenza (9858). Per applicare il metodo del kernel descritto in precedenza è stato necessario adattare le dimensioni di Score eliminando da essa tutte le colonne il cui gene non compariva all'interno della rete HINT+HI2012, in modo che il numero di feature tra le matrici siano identici. La matrice risultante prende il nome di

Data:
- M mutation matrix ;
- Net interaction network ;
- σ valore di restart per la random walk di Hotnet2 ;
- α valore soglia per test P-value ;

$Set = \emptyset$, insieme contenente i geni scoperti da Hotnet2 ;
forall *T, tumore distinto in M* **do**
  creazione MT, sottomatrice di M contenente solo pazienti affetti da tumore T ;
  definizione vettore heat $hT = []$;
  forall *G, gene distinto in Net* **do**
    calcolo *heat*, somma mutazioni geniche di G presenti in MT ;
    aggiunta tupla $(g, heat)$ al vettore hT ;
  end
  esecuzione Hotnet2 su Net, con parametri (hT, σ) ;
  creazione *subNet*, insieme contenente le sottoreti scoperte da Hotnet2 con p-value $\leq \alpha$;
  forall *Cg, sottorete distinta in subNet* **do**
    forall *G, gene distinto in Cg* **do**
      aggiunta gene G all'insieme Set ;
    end
  end
end
creazione matrice H mediante riduzione feature di M utilizzand i geni contenuti in Set ;
applicazione metodi di classificazione su matrice H ;

Algorithm 1: procedura per la riduzione feature di una mutation matrix mediante cluster genici ottenuti da Hotnet2

Score-reduction.
Per via delle dimensioni di Score-reduction, che contiene all'incirca 1/3 delle feature della matrice di partenza (Score), si è deciso di osservare il comportamento di un classificatore allenato su essa senza aggiungere informazioni aggiuntive.

Dai risultati 3.1 e 3.2 si può evincere come un classificatore allenato con la matrice Score-reduction presenta un'accuratezza del tutto simile ad un classificatore allenato con la matrice di Score (tabelle 2.4 e 2.6). Questo rappresenta un risultato importante per la ricerca di cluster genici associabili allo sviluppo tumorale, in quanto si è scoperto che ben 14941 geni presenti nella matrice di Score non portano a nessun contributo significativo per la classificazione tumorale. Di conseguenza, l'esclusione di questi geni ha permesso di restringere il campo di ricerca solo sui geni presenti nella rete HINT+HI2012.
Quindi si è analizzato il comportamento di classificatori allenati con le matrici k-adiacenza e k-influenza, definite mediante l'applicazione del kernel tra la matrice di Score-reduction e le matrici di adiacenza/influenza rispettivamente. Grazie al parametro β, è possibile regolare l'influsso delle informazioni topologiche all'interno delle due matrici, in modo da valutare i potenziali miglioramenti che si potrebbero ottenere dal kernel rispetto ai

Tabella 3.1: Media su 10 prove di Liblinear eseguite sulla matrice di Score-reduction con risolutore R2-L1 e parametro costo C = 1.

	Accuratezza media (%)									
Tumore	*P.1*	*P.2*	*P.3*	*P.4*	*P.5*	*P.6*	*P.7*	*P.8*	*P.9*	*P.10*
blca	0.00	0.00	12.00	4.00	4.00	12.00	12.00	12.00	4.00	4.00
brca	27.27	19.69	51.51	46.96	55.30	47.72	57.57	50.00	57.57	57.57
coadread	41.40	40.62	84.37	82.03	79.68	90.62	82.03	82.03	81.25	91.40
gbm	40.57	39.13	43.47	47.82	43.47	57.97	43.47	46.37	50.72	43.47
hnsc	34.21	27.63	50.00	42.10	44.73	46.05	42.10	39.47	53.94	40.78
kirc	27.20	31.20	80.80	74.40	75.20	70.40	72.80	69.60	76.80	66.40
laml	49.01	37.25	76.47	58.82	56.86	62.74	72.54	62.74	72.54	56.86
luad	21.05	35.08	36.84	43.85	35.08	36.84	50.87	49.12	52.63	43.85
lusc	10.86	10.86	21.73	34.78	21.73	23.91	30.43	26.08	17.39	28.26
ov	55.17	52.58	75.86	76.72	75.86	68.96	73.27	70.68	73.27	78.44
ucec	16.13	17.74	70.96	66.12	67.74	54.83	74.19	61.29	67.74	58.06
Tutti	57.15	58.28	62.00	59.41	58.96	58.96	61.78	58.06	62.57	59.97

Tabella 3.2: Prove (10) Random Forest eseguite sulla matrice di Score-reduction

N.alberi	**Acc. Media**	**Var.**
10	0.52	0.008
50	0.56	0.010
100	0.57	0.012
250	0.58	0.009
500	0.58	0.014

risultati ottenuti in 3.1 e 3.2. Inoltre, per la costruzione della matrice k-influenza sono state considerate tutte e 3 le trasformazioni di simmetria della matrice di influenza (media, massimo, minimo). Sono stati eseguiti 10 prove di classificazione su ogni matrice descritta in precedenza. La media dell'accuratezza e relativa varianza di queste prove sono riportate nella tabella 3.3.

Per i classificatori allenati con la matrice k-adiacenza si può osservare un aumento di accuratezza solo a seguito di una diminuzione del valore di β. Si intuisce quindi che forzare SVM ad avere una maggior considerazione per le informazioni espresse dalla matrice di adiacenza comporta solamente ad ottenere classificatori con una precisione inferiore. Viceversa, i classificatori allenati con la matrice k-influenza hanno ottenuto risultati del tutto simili tra loro, indipendentemente dal valore di β e dalla trasformazione di simmetria utilizzata. L'influsso della matrice di influenza all'interno di Score-reduction può essere considerata una operazione superflua in quanto la precisione del classificatore non subisce sostanziali variazioni rispetto a quello definito in 3.1. In questo caso si è potuto concludere come la gestione delle informazioni topologiche attraverso la procedura del kernel non permetta di ottenere vistosi miglioramenti rispetto ad un semplice studio delle sole informazioni mutagene.

La procedura Hotnet2 è stato eseguita due volte, in modo da ottenere due risultati

Tabella 3.3: Media su 10 prove di Liblinear eseguite sulle matrici k-adiacenza e k-influenza con risolutore R2-L1 e parametro costo C = 1.

Matrice	**Beta**	**Acc. Media**	**Var.**
k-adiacenza	10	50.96	1.81
	1	54.72	1.22
	0.1	59.64	1.41
k-influenza (simmetria media	10	60.15	1.32
	1	60.74	1.33
	0.1	60.32	0.78
k-influenza (simmetria massima)	10	60.48	1.30
	1	61.42	0.93
	0.1	61.32	1.58
k-influenza (simmetria minima)	10	61.81	0.90
	1	60.68	1.02
	0.1	59.87	1.03

diversi in base ai valori di soglia per p-value $\alpha = 0.05$ e $\alpha = 0.03$. Il primo è definito come un valore di convenzione per i test di p-value , mentre il secondo è stato considerato per filtrare soluzioni più restrittive. In entrambi i casi si è sempre usufruito della rete HINT+HI2012, recuperando le informazioni mutagene dalla matrice di Score. Come valore di restart per la random walk si è utilizzato lo stesso valore consigliato in precedenza $\sigma = 0.4$.
Come risultato della procedura si sono ottenute due sottomatrici di Score, in base ai valori di soglia $\alpha = 0.05$ e $\alpha = 0.03$: la prima è descritta da 941 geni distinti mentre la seconda contiene un numero minore, solo 808 geni.Prima di eseguire i test di classificazione su queste due matrici, chiamate rispettivamente HotScore-005 e HotScore-003, si è eseguita un'analisi sui risultati ottenuti da Hotnet2.
Le tabelle 3.4 e 3.5 descrivono per ogni tumore il numero di sottoreti scoperte da Hotnet2, il numero complessivo di geni che le compongono e la dimensione della più piccola sottorete tra esse. Grazie a quest'ultimo risultato è stato possibile osservare come l'abbassamento del valore di α consegue ad una maggiore considerazione per sottoreti di dimensioni superiori. Mediante il valore di soglia $\alpha = 0.05$ sono stati scoperti ben 5 tumori con una sottorete di dimensione minima pari a 4 rispetto alla prova con $\alpha = 0.03$ in cui solo 2 di essi (brca, hnsc) presentano il medesimo risultato. Di conseguenza, l'abbassamento di α da 0.05 a 0.03 ha permesso l'eliminazione di alcune sottoreti con dimensioni ininfluenti per alcuni tumori, diminuendo così l'insieme dei geni presenti in HotScore-003 (808) rispetto a HotScore-005 (941).
Confrontando la cardinalità degli insiemi di geni associati ad ogni tumore (nelle tabelle 3.4 e 3.5) con le dimensioni delle due matrici, si è constatata la presenza di diversi geni associati simultaneamente a più tumori distinti. Per valutare questa distribuzione sono stati definiti due istogrammi 3.1 e 3.2 che descrivono rispettivamente la suddivisione dei 941 e 808 geni in base al numero di tumori distinti associati ad ogni gene. Gli istogrammi sono quindi composti da 11 insiemi, dove il primo di essi identifica tutti i geni coperti da un solo tumore, mentre nell'ultimo insieme sono presenti i geni coperti da tutti i tumori considerati in questo studio.

Tabella 3.4: Distribuzione cluster genici ottenuti mediante Hotnet2 con $\alpha = 0.05$ (941 geni distinti trovati). Per ogni tumore viene descritto il numero di sottoreti scoperte da Hotnet2, la dimensione della più piccola sottorete tra esse e il numero complessivo di geni scoperti.

Tumore	**N. sottoreti**	**Dim. min.**	**Geni Tot.**
blca	6	5	49
brca	20	4	134
coadread	14	7	337
gbm	18	5	188
hnsc	16	4	112
kirc	21	7	296
laml	25	4	164
luad	42	4	286
lusc	36	4	261
ov	10	5	94
ucec	22	6	242

Tabella 3.5: Distribuzione cluster genici ottenuti mediante Hotnet2 con $\alpha = 0.03$ (808 geni distinti trovati). Per ogni tumore viene descritto il numero di sottoreti scoperte da Hotnet2, la dimensione della più piccola sottorete tra esse e il numero complessivo di geni scoperti.

Tumore	**N. sottoreti**	**Dim. min.**	**Geni Tot.**
blca	6	5	49
brca	20	4	134
coadread	14	7	337
gbm	18	5	188
hnsc	16	4	112
kirc	21	7	296
laml	16	5	128
luad	7	8	126
lusc	20	5	197
ov	10	5	94
ucec	16	7	206

Entrambi gli istogrammi presentano lo stesso andamento. In particolare si può osservare che più della metà dei geni presenti nelle due matrici sono coperti esattamente da un singolo tumore, secondo Hotnet2. Questo potrebbe essere considerato un vantaggio per migliorare la classificazione tumorale in quanto esisterebbero molti geni esclusivi ad un singolo tipo tumorale.
Per dimostrare tale affermazione, si è analizzato Score-reduction in modo da definire per ogni gene il numero distinto di tumori coperti dalle sue mutazioni. Successivamente, queste informazioni sono state utilizzate per identificare su ogni paziente presente in

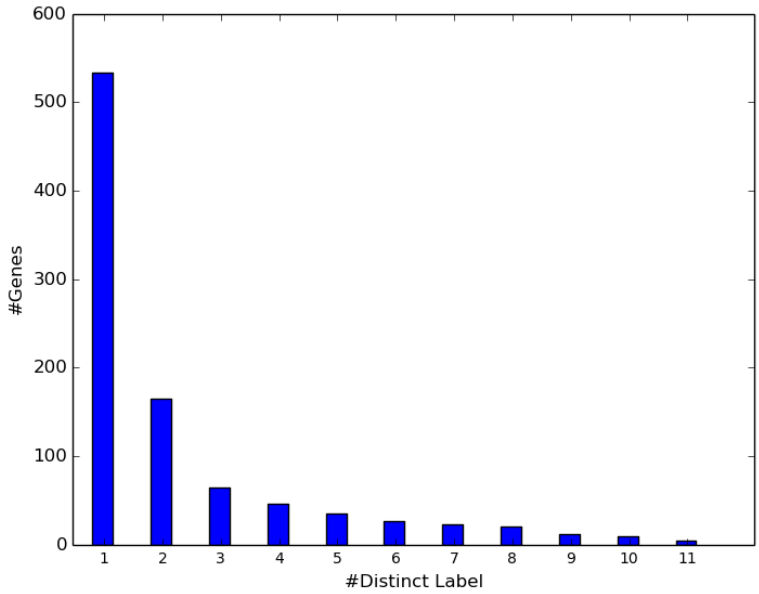

Figura 3.1: Istogramma relativo alla suddivisione dei geni trovati con Hotnet2 con valore di soglia $\alpha = 0.05$, in base al numero di tumori distinti associati ad ogni gene.

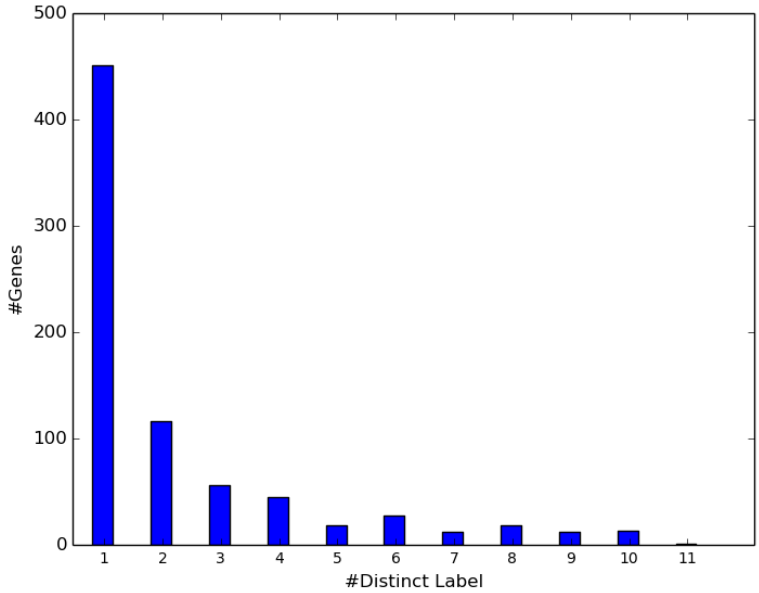

Figura 3.2: Istogramma relativo alla suddivisione dei geni trovati con Hotnet2 con valore di soglia $\alpha = 0.03$, in base al numero di tumori distinti associati ad ogni gene.

HotScore-003 i due geni mutati con la copertura minima e massima. In questo modo si è potuto suddividere i pazienti a seconda del numero di tumori coperti dai due geni identificati in precedenza. Tale suddivisione, descritta nell'istogramma 3.3, ha permesso di verificare che in ogni paziente esiste almeno una mutazione genomica che è associabile

a tutti i tumori considerati in questo studio. Allo stesso tempo si è constatato che non esiste nessuna mutazione genomica esclusiva per un singolo tumore, al contrario di quanto descritto dalla suddivisione dei geni secondo Hotnet2 (3.1 e 3.2).
La motivazione deriva dal fatto che un cluster genico identificato da Hotnet2 per un dato tumore può presentare anche mutazioni in un gruppo ristretto di pazienti affetti da un tumore differente, ma Hotnet2 non è in grado di associare lo stesso cluster anche a quest'ultimo per via del numero troppo trascurabile di individui. Grazie a 3.3 si potuto confermare l'inesistenza di geni in grado di manifestare delle mutazioni solo in presenza di uno sviluppo tumorale specifico e che la distribuzione descritta nei due istogrammi 3.1 e 3.2 è da considerare puramente indicativa e non ha fini pratici per il miglioramento della classificazione tumorale.

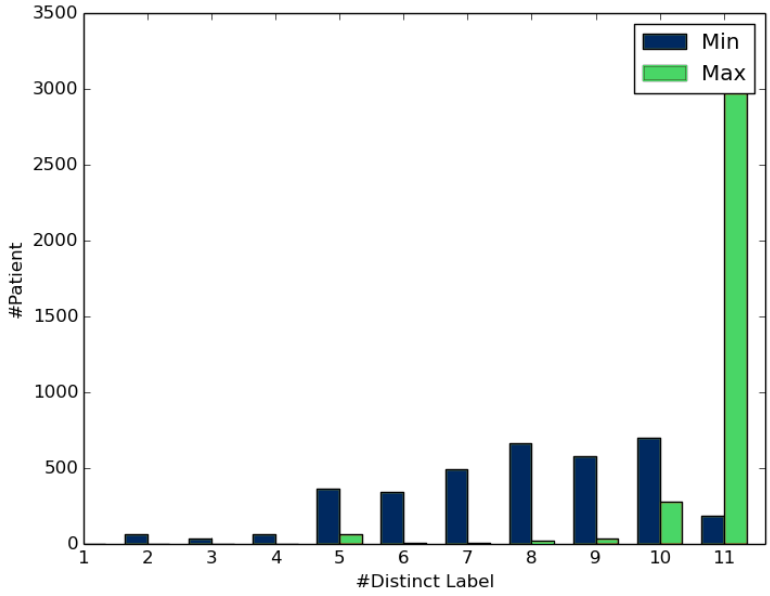

Figura 3.3: Istogramma relativo alle frequenze di associazione minima e massima per le mutazioni geniche definite da ogni paziente in HotScore-003

Dai risultati di classificazione descritti nelle tabelle 3.6 e 3.7 si può osservare come la precisione dei due classificatori sia all'incirca la stessa, sebbene quello allenato su HotScore-003 utilizzi 133 geni in meno. Si intuisce che tali geni fanno parte di cluster genici scartati con l'abbassamento del valore di α da 0.05 a 0.03, evidenziando così l'importanza di considerare soluzioni più restrittive generate da Hotnet2. Tuttavia la precisione dei due classificatori ha subito un peggioramento sostanzioso rispetto a quello allenato su Score-reduction (3.1) che risulta essere ancora il migliore in quanto a precisione e numero di feature utilizzate.
Un ulteriore abbassamento del parametro α non porterebbe a nessuna variazione in termini di accuratezza per il classificatore, ma solo ad un possibile abbassamento del numero di feature considerate.

Hotnet2 e Kernel sono due metodi che hanno permesso di utilizzare le informazioni topologiche con lo scopo di migliorare la classificazione tumorale, ma in questo caso non si sono ottenuti miglioramenti evidenti rispetto ad un'analisi sulle sole informazioni mutagene. Tuttavia è bene osservare che rispetto alla classificazione eseguita sulla matrice di Score si sono compiuti passi in avanti: i classificatori allenati su Score-reduction presentano la stessa precisione utilizzando 1/3 delle feature di Score, mentre il passaggio dalla matrice di Score-reduction a HotScore-003 ha comportato ad una riduzione delle feature di un fattore 10 (circa) a scapito di un'abbassamento della precisione del classificatore di all'incirca 10%. Questi risultati sono incoraggianti in quanto dimostrano che è possibile ridurre la matrice iniziale al fine di ottenere un gruppo ristretto di geni per classificare con una buona precisione i vari casi tumorali. Infine, queste sperimentazione sono servite per ideare un metodo in grado di evidenziare geni che mutano frequentemente in un solo tipo tumorale.

Tabella 3.6: Media su 10 prove di Liblinear eseguite sulla matrice HotScore-005 con risolutore R2-L1 e parametro costo C = 1.

	Accuratezza media +(%)									
Tumore	*P.1*	*P.2*	*P.3*	*P.4*	*P.5*	*P.6*	*P.7*	*P.8*	*P.9*	*P.10*
blca	8.00	4.00	4.00	4.00	4.00	16.00	0.00	4.00	12.00	16.00
brca	34.84	34.84	40.90	37.12	40.15	40.15	34.09	34.84	42.42	41.66
coadread	79.68	84.37	82.81	88.28	83.59	82.81	87.50	85.93	84.37	86.71
gbm	31.88	27.53	30.43	31.88	39.13	31.88	44.92	34.78	27.53	34.78
hnsc	36.84	35.52	38.15	48.68	40.78	34.21	38.15	36.84	36.84	46.05
kirc	55.20	60.00	59.20	60.00	62.40	59.20	63.20	57.60	57.60	61.60
laml	49.01	52.94	49.01	47.05	49.01	47.05	60.78	45.09	43.13	52.94
luad	43.85	47.36	45.61	38.59	52.63	42.10	36.84	43.85	43.85	29.82
lusc	21.73	30.43	21.73	19.56	17.39	26.08	30.43	26.08	19.56	30.43
ov	71.55	70.68	74.13	75.86	71.55	75.86	68.96	74.13	77.58	70.68
ucec	48.38	58.06	56.45	59.67	59.67	50.00	54.83	41.93	51.61	62.90
Tutti	49.83	52.08	52.64	53.77	54.11	52.31	53.66	51.07	52.31	54.67

Tabella 3.7: Media su 10 prove di Liblinear eseguite sulla matrice HotScore-003 con risolutore R2-L1 e parametro costo C = 1.

	Accuratezza media (%)									
Tumore	*P.1*	*P.2*	*P.3*	*P.4*	*P.5*	*P.6*	*P.7*	*P.8*	*P.9*	*P.10*
blca	0.00	4.00	8.00	12.00	28.00	0.00	4.00	20.00	12.00	20.00
brca	38.63	35.60	37.87	40.15	42.42	34.84	37.87	35.60	39.39	39.39
coadread	85.15	83.59	85.93	78.90	86.71	80.46	89.06	77.34	82.81	83.59
gbm	36.23	23.18	30.43	30.43	27.53	34.78	28.98	37.68	27.53	34.78
hnsc	44.73	48.68	42.10	39.47	36.84	39.47	42.10	40.78	36.84	35.52
kirc	64.80	55.20	59.20	59.20	65.60	58.40	60.00	67.20	62.40	60.00
laml	56.86	52.94	41.17	47.05	45.09	45.09	47.05	35.29	49.01	52.94
luad	43.85	45.61	47.36	50.87	42.10	42.10	38.59	42.10	56.14	29.82
lusc	17.39	28.26	23.91	32.60	26.08	26.08	41.30	39.13	23.91	28.26
ov	68.96	73.27	77.58	68.96	73.27	67.24	68.10	74.13	74.13	75.86
ucec	41.93	61.29	45.16	54.83	62.90	46.77	54.83	56.45	51.61	35.48
Tutti	52.76	52.53	52.53	52.31	54.79	49.83	52.98	53.32	53.21	51.52

Capitolo 4

Classificazione mediante problema di maximum coverage

In questo capitolo viene descritto un nuovo metodo basato sul problema di maximum coverage, in grado di rilevare geni che mutano frequentemente solo in pazienti affetti da un determinato tipo tumorale. Una descrizione dettagliata del metodo viene proposta nella sezione 4.1, mentre nelle sezioni 4.2 e 4.3 vengono riportate due possibili implementazioni mediante paradigma Greedy e ILP rispettivamente. Nella sezione 4.4 vengono analizzati i risultati ottenuti dai due algoritmi.

4.1 Definizione del problema

Nel capitolo precedente sono state effettuate alcune analisi riguardanti la frequenza di associazione geni mutati-tumori, dalle quali si è riscontrato che non esiste nessuna mutazione genomica esclusiva ad un solo tipo tumorale. Questo avrebbe aiutato di molto la classificazione tumorale, in quanto la presenza di feature esclusive avrebbe consentito al classificatore di effettuare predizioni più accurate. Tuttavia, anche senza la presenza di feature esclusive, un gene può comunque essere associato ad un determinato tumore in base alla frequenza di mutazione presente in un gruppo di individui. Per fare un esempio, 100 pazienti sono suddivisi equamente in due gruppi, in base alla tipologia del tumore associato. Tutti i pazienti del primo gruppo riportano una mutazione per un determinato gene, mentre nell'altro gruppo solo un paziente su dieci presenta una mutazione per il medesimo. Da questo andamento si può dedurre come il gene considerato abbia una relazione con il tumore associato al primo gruppo. Questo non esclude il fatto che possano esistere geni che presentino mutazioni di rado, ma che danno un contributo rilevante per lo sviluppo tumorale oppure geni con mutazioni frequenti, ma ininfluenti per la crescita del tumore. Infatti, il gene definito nell'esempio presenta solo la proprietà di mutare frequentemente per un tipo tumorale e allo stesso tempo poco per l'altro. Un gruppo formato da tali geni potrebbe essere considerato interessante per migliorare la classificazione tumorale, in quanto si avvicina all'idea di avere un gruppo di geni esclusivi per ogni tipo tumorale.
Per ricercare un gruppo di geni con queste caratteristiche bisogna analizzare le informazione mutagene di una mutation matrix. Questo tipo di ricerca può essere tradotto come una variante del problema *maximum coverage*. Per definizione, dato un numero k e una

serie di insiemi $S_1, S_2, ...S_n$, con ogni $S_i \subseteq U$ con U universo di elementi e con gli S_i che potrebbero avere degli elementi in comune, maximum coverage seleziona al massimo k insiemi in modo da massimizzare il numero di elementi coperti (ottenuti mediante l'unione degli insiemi selezionati). Maximum coverage è un problema NP-Hard.
Al fine di utilizzare correttamente le informazioni contenute all'interno della mutation matrix si è dovuto adattare il problema del maximum coverage: per ogni tipo tumorale T viene ricercato un insieme contenente k geni, le cui mutazioni coprono il maggior numero di pazienti affetti da T e allo stesso tempo il minor numero possibile di pazienti malati da tumori diversi. Per ottenere questo risultato è stato necessario ideare un valore di affinità r, in grado di esprimere la relazione che sussiste tra k geni candidati e il tumore T. Esso viene calcolato focalizzando l'attenzione sui pazienti che presentano una mutazione in uno dei k geni candidati presenti nella mutation matrix. Ai pazienti affetti da tumore T viene assegnato un peso positivo, mentre i pazienti malati da tipologie tumorali differenti assumono un peso negativo. Il valore di affinità r viene quindi espresso dalla somma complessiva dei pesi assegnati ai soli pazienti coperti dalle mutazioni genomiche dei geni candidati. In questo modo si possono valutare diverse combinazioni di k geni, selezionando solamente quelle con un valore di r elevato per il tumore T.
Per questo studio sono stati definiti 4 diversi pesi, riportati di seguito. In questo caso N rappresenta il numero di pazienti totali in una mutation matrix, L il numero di tumori distinti considerati, X rappresenta un qualsiasi tipo tumorale diverso dal tumore T, N_T e N_X rispettivamente il numero di pazienti affetti da tumore T e X all'interno della mutation matrix.

- *Peso Naive*:

$$P = \begin{cases} +1 & \text{se paziente affetto da tumore} = T \\ -1 & \text{se paziente affetto da tumore} \neq T \end{cases}$$

  Naive rappresenta l'approccio più semplice per calcolare il valore di affinità. Esso si limita solamente a valutare la differenza tra il numero di pazienti affetti da T e quelli malati da tipologie tumorali differenti.

- *Peso One vs One for All*:

$$P = \begin{cases} +\frac{1}{N_T} & \text{se paziente affetto da tumore} = T \\ -\frac{1}{(N-N_T)} & \text{se paziente affetto da tumore} \neq T \end{cases}$$

  One vs One for All mette a confronto il numero di pazienti affetti da T con tutti gli altri pazienti con tumori diversi. Rispetto a *Naive*, risulta bilanciato: nel caso in cui venissero coperti tutti e soli i pazienti affetti da T, la somma dei loro pesi darebbe come risultato 1. In modo analogo, se venissero coperti tutti e soli i pazienti affetti da tumori differenti, la somma dei loro pesi darebbe come risultato -1. Questo permette di bilanciare la differenza che esisterebbe tra il numero di pazienti affetti da T (N_T) con quelli malati da tumori diversi ($N - N_T$).

- *Peso One vs One*:

$$P = \begin{cases} +\frac{N_T}{N} & \text{se paziente affetto da tumore} = T \\ -\frac{N_X}{N} & \text{se paziente affetto da tumore} = X \end{cases}$$

  One vs One permette di definire per ogni tipo tumorale un peso corrispondente alla sua frequenza di apparizione all'interno della mutation matrix, in termini di pazienti malati su pazienti totali. Tale approccio non risulta bilanciato.

- *Peso One vs All*:

$$P = \begin{cases} +\frac{1}{N_T} & \text{se paziente affetto da tumore} = T \\ -\frac{1}{N_X}\frac{1}{(L-1)} & \text{se paziente affetto da tumore} = X \end{cases}$$

  One vs All, anch'esso bilanciato, definisce dei pesi che identificano il numero di pazienti malati per ogni tumore. L'unica differenza sta nella gestione dei pesi per i pazienti affetti da un tipo tumorale diverso da T, ai quali viene aggiunto un valore che identifica il numero distinto di tumori diversi da T presenti nella mutation matrix. Quest'ultimo termine permette di ottenere dei pesi bilanciati. Infatti se venissero coperti tutti i pazienti della mutation matrix, la somma complessiva dei pesi darebbe un valore nullo.

Più formalmente, il problema di maximum coverage tra geni e pazienti viene definito nel seguente modo:

- Istanza:
  - mutation matrix M di dimensione (NxG), formata da N pazienti e G geni
  - selezione del tipo tumorale T
  - selezione del peso P
  - selezione della cardinalità k per l'insieme di geni da rilevare
- Obiettivo: ricerca di k geni le cui mutazioni coprono il maggior numero di pazienti affetti da T e allo stesso tempo il minor numero possibile di pazienti malati da tumori diversi.

Per risolvere il problema di maximum coverage tra geni e pazienti descritto in precedenza, sono stati sviluppati due algoritmi distinti: uno di essi utilizza un approccio Greedy mentre l'altro viene definito secondo un modello di ottimizzazione ILP (Integer Linear Problem). Entrambi i metodi associano ad ogni tumore k geni, con valore di affinità elevato mediante l'utilizzo dei pesi.

4.2 Approccio Greedy

L'algoritmo Greedy[3] è un paradigma di programmazione che utilizza un approccio euristico per la risoluzione di problemi di ottimizzazione. Ad ogni iterazione viene eseguita la scelta migliore a livello locale con la speranza di poter raggiungere la soluzione ottimale globale alla fine delle iterazioni. Sebbene la strategia greedy il più delle volte produce soluzioni sub-ottime globali, tali valori possono essere considerati come un'approssimazione della soluzione ottima globale ottenuta in tempi computazionali ragionevoli.
In questo caso è stato sviluppato un'algoritmo greedy in grado di selezionare k geni per ogni tipo tumorale all'interno di una mutation matrix. Esso utilizza uno dei 4 pesi definiti in precedenza (Naive, One vs One for All, One vs All, One vs One), inizializzati mediante le informazioni di distribuzione dei tipi tumorali reperibili dalla mutation matrix. Ad ogni paziente viene assegnato un valore in base al tipo tumorale associato e al peso P utilizzato durante lo svolgimento del metodo. Partendo da un insieme soluzione vuoto che conterrà i k geni per un dato tumore, ad ogni iterazione verrà aggiunto ad esso il gene che da il risultato migliore in termini di valore di affinità r assieme agli altri geni già selezionati nelle iterazioni precedenti, fino al raggiungimento di k geni presenti nell'insieme soluzione. Una descrizione formale dell'algoritmo può essere osservata in Alg.2.

Data:
- mutation matrix M di dimensione (NxG), formata da N pazienti e G geni ;
- numero k di geni da trovare ;
- tipo P di peso da utilizzare ;
- tumore T target ;

inizializzazione peso P in base alle informazioni contenute in M e al tumore T ;
assegnazione peso p_i al paziente i-esimo in base al tumore associato ad esso ;
$Set_T = \emptyset$, insieme contenente k geni rilevati dal Greedy per il tumore T ;
while $|Set_T| \neq k$ **do**
    $r = -\infty$, valore di affinità tra i geni nel Set_T e il tumore T ;
    $g_{cand} = null$, gene candidato per il Set_T ;
    forall *gene* $g \in M$ *AND* $g \notin Set_T$ **do**
        $r_{temp} = \sum_{i \in N} p_i * M_{i,j}$ con j = posizione geni $\in \{Set_T + g\}$
        if $r_{temp} > r$ **then**
            $r = r_{temp}$;
            $g_{cand} = g$;
        end
    end
    aggiunta g_{cand} a Set_T ;
end
return Set_T ;

Algorithm 2: algoritmo Greedy

A seguire viene riportato un esempio esplicativo che descrive l'applicazione dell'al-

goritmo Greedy utilizzando una mutation matrix creata per lo scopo (4.1). La matrice descrive le mutazioni genomiche di 10 pazienti affetti da 3 possibili tumori (Z1,Z2,Z3), focalizzando l'attenzione su 5 geni distinti (A,B,C,D,E). L'algoritmo Greedy viene eseguito per ogni tumore distinto della matrice, utilizzando tutti i pesi definiti in precedenza. Per ogni tumore distinto viene ricercato un gruppo formato da $K = 2$ geni. Nella tabella 4.2 vengono riportati i valori dei 4 pesi assegnati ai pazienti della mutation matrix in relazione al tipo tumorale associato ad essi e al tumore target T considerato dal Greedy. Durante la prima iterazione, l'insieme soluzione dei geni è vuoto. Quindi l'algoritmo si limita a trovare il gene che abbia il valore di affinità r più alto, a seconda del tumore T considerato e al peso P utilizzato. Nella seconda iterazione, Greedy analizza tutte le combinazioni tra il gene trovato nell'iterazione precedente e i geni restanti in modo da trovare quella con il valore r maggiore (sempre in relazione a T e P). Questo processo viene descritto dalle tabelle 4.3 e 4.4.
I risultati per ogni peso P e tumore T considerato vengono riportati nella tabella 4.5. Essa descrive l'insieme dei geni soluzione, il numero di pazienti affetti da tumore T aventi una mutazione nei geni soluzione e il numero di pazienti che presentano una mutazione nei geni soluzione ma non sono affetti da T. Dalla tabella si può constatare come gli insiemi di geni e il numero di pazienti ottenuti risultano uguali al variare dei pesi utilizzati. E' bene osservare che questo è un caso particolare causato dalla struttura della mutation matrix 4.1. Generalmente l'utilizzo di pesi diversi per lo stesso tipo tumorale T comporta ad ottenere insiemi di geni differenti.

Tabella 4.1: Descrizione mutation matrix per esempio Greedy

Tumore	**A**	**B**	**C**	**D**	**E**
Z1	0	0	1	1	1
Z1	0	0	1	0	1
Z1	0	1	1	0	1
Z1	0	1	1	0	0
Z2	1	0	1	1	1
Z2	1	0	1	0	0
Z2	1	0	1	0	0
Z3	0	1	1	1	1
Z3	0	1	1	0	0
Z3	0	1	1	0	1

Tabella 4.2: Calcolo pesi in base alla definizione P e alle informazioni contenute nella mutation matrix per esempio Greedy

	N. Pesi Pazienti			**O. vs O.f.A. Pesi Pazienti**			**O. vs O. Pesi Pazienti**			**O. vs A. Pesi Pazienti**		
T	**Z1**	**Z2**	**Z3**	**Z1**	**Z2**	**Z3**	**Z1**	**Z2**	**Z3**	**Z1**	**Z2**	**Z3**
Z1	+1	-1	-1	+1/4	-1/6	-1/6	+4/10	-3/10	-3/10	+1/4	-1/6	-1/6
Z2	-1	+1	-1	-1/7	+1/3	-1/7	-4/10	+3/10	-3/10	-1/8	+1/3	-1/6
Z3	-1	-1	+1	-1/7	-1/7	+1/3	-4/10	-3/10	+3/10	-1/8	-1/6	+1/3

Tabella 4.3: Prima iterazione dell'esempio Greedy: Calcolo valore affinità r per ogni feature in relazione al tumore T e al peso considerato P

			Geni				
Peso	T	**Insieme Sol.**	A	B	C	D	E
N.	Z1	$\emptyset$	-3	-1	-2	-1	**0**
	Z2	$\emptyset$	**+3**	-5	-4	-1	-4
	Z3	$\emptyset$	-3	**+1**	-4	-1	-2
O. vs O.f.A.	Z1	$\emptyset$	-3/6	0	0	-1/12	**+1/4**
	Z2	$\emptyset$	**+1**	-5/7	0	+1/21	-8/21
	Z3	$\emptyset$	-3/7	**+5/7**	0	+1/21	+2/21
O. vs O.	Z1	$\emptyset$	-9/10	-1/10	-3/10	-2/10	**+3/10**
	Z2	$\emptyset$	**+9/10**	-17/10	-16/10	-4/10	-15/10
	Z3	$\emptyset$	-9/10	**+1/10**	-16/10	-4/10	-9/10
O. vs. A.	Z1	$\emptyset$	-3/6	0	0	-1/12	**+1/4**
	Z2	$\emptyset$	**+1**	-6/8	0	+1/24	-3/8
	Z3	$\emptyset$	-3/6	**+3/4**	0	+1/24	-1/24

Tabella 4.4: Seconda iterazione dell'esempio Greedy: Calcolo valore affinità r per ogni feature in relazione al tumore T e al peso considerato P

			Geni				
Peso	T	**Insieme Sol.**	A	B	C	D	E
N.	Z1	E	-4	**0**	-2	-1	.
	Z2	A	.	-2	-4	**+1**	-2
	Z3	B	-2	.	-4	**-1**	-2
O. vs O.f.A.	Z1	E	-1/12	**+2/6**	0	+1/4	.
	Z2	A	.	+2/7	0	**+5/7**	+2/7
	Z3	B	+2/7	.	0	**+3/7**	+2/7
O. vs O.	Z1	E	-3/10	**+4/10**	-2/10	+3/10	.
	Z2	A	.	-8/10	-16/10	**+2/10**	-9/10
	Z3	B	-8/10	.	-16/10	**-6/10**	-9/10
O. vs. A.	Z1	E	-2/24	**+8/24**	0	+6/24	.
	Z2	A	.	+1/4	0	**+13/24**	-1/24
	Z3	B	+1/4	.	0	**+7/24**	+4/24

Tabella 4.5: Risultati dell'esempio Greedy

		Soluzioni		
Peso	T	*Geni*	# Paz. affetti da T	# Paz. non affetti da T
N.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4
O. vs O.f.A.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4
O. vs O.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4
O. vs. A.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4

4.3 Approccio ILP

ILP (Integer Linear Programming) è un metodo basato sulla risoluzione di problemi di ottimizzazione interi, mediante la massimizzazione/minimizzazione di una funzione obiettivo lineare soggetta a vincoli lineari.Un vincolo esprime una limitazione per l'insieme dei valori che possono assumere alcune variabili sotto determinate condizioni. Lo scopo di ILP è trovare un'assegnazione per tutte le variabili che sia in grado di ottimizzare la funzione obiettivo e allo stesso tempo rispettare tutti i vincoli definiti nel problema. Tuttavia le soluzioni ottenute da un ILP non sempre si possono considerare ottime a livello globale.
Per questo studio è stato definito un modello ILP (4.1) in modo da risolvere il problema della ricerca di k geni per un tumore distinto, utilizzando le definizioni dei pesi P descritti in precedenza e le informazioni contenute in una mutation matrix M (di dimensione NxG). Nel modello, l'elemento generico della matrice viene espresso dal valore a_{ij} che rappresenta la possibile mutazione al gene j-esimo per il paziente i-esimo. In base al peso P utilizzato e alla distribuzione dei tumori all'interno della mutation matrix, vengono inizializzati N pesi p_i da assegnare per ogni paziente. Per identificare l'insieme soluzione dei k geni i relativi pazienti che presentano una mutazione all'interno di tale insieme, il modello associa ad ogni paziente una variabile x_i e ad ogni gene una variabile y_j. Queste variabili sono di tipo binario e servono per identificare i geni e i pazienti che compariranno nella soluzione del problema. Se $x_j = 1$, allora il gene j-esimo comparirà nell'insieme soluzione, mentre per $y_i = 1$ il paziente i-esimo avrà una mutazione in uno dei k geni che compongono l'insieme soluzione. Data la complessità del modello ILP, possono sussistere problemi computazionali in caso di valori elevati per k o di mutation

matrix con dimensioni considerevoli.

$$
\begin{aligned}
\max \quad & \sum_{i \in N} p_i y_i & & (1) \\
& \sum_{j \in G} x_j = K & & (2) \\
& \sum_{j \in G} a_{ij} x_j \geq y_i & \forall i \in N & (3) \\
& \sum_{j \in G} a_{ij} x_j \leq K y_i & \forall i \in N & (4) \\
& x_j \in (0,1) & \forall j \in G & (5) \\
& y_i \in (0,1) & \forall i \in N & (6) \\
& a_{ij} \in (0,1) & \forall (i,j) \in M & (7)
\end{aligned}
\tag{4.1}
$$

Di seguito viene elencata una breve descrizione dei vari vincoli definiti nel problema ILP:

1. Funzione Obiettivo: massimizzare la somma dei pesi dei pazienti y_i selezionati in base al valore del peso p_i associato.

2. Vincolo: il numero di geni x_j selezionati deve essere uguale a k

3. Vincolo: ogni paziente che compare nell'insieme soluzione ($y_i = 1$) deve avere almeno una mutazione genetica in un gene che compare nell'insieme soluzione ($x_j = 1$).

4. Vincolo: tutti i pazienti y_i che presentano una mutazione genetica per almeno uno dei k geni dell'insieme soluzione ($x_j = 1$) devono essere selezionati ($y_i = 1$), indipendentemente dal valore (positivo o negativo) del peso associato a y_i . Questo vincolo permette di prestare più attenzione nella scelta dei x_j, altrimenti verrebbero selezionati solo i pazienti y_i con peso positivo.

Anche in questo caso viene proposto un esempio esplicativo per descrivere il funzionamento del modello ILP, utilizzando i dati contenuti nella stessa mutation matrix definita per l'esempio Greedy (4.1) in modo da evidenziare le differenze tra i due metodi. Di conseguenza, per questo esempio risultano identici anche i valori dei pesi P calcolati in precedenza (4.2). ILP analizza la funzione obiettivo in base a tutte le combinazioni che possono sussistere tra k geni, rispetto all'algoritmo Greedy che costruisce un insieme di k geni ottimali dopo una serie di iterazioni. Questo comportamento viene descritto nella tabella 4.6. Si può notare che per il peso Naive e T =Z1, sono stati rilevati ben due gruppi di geni distinti che permettono di ottenere il valore più alto per la funzione obiettivo. Tuttavia in questo caso viene selezionata la coppia di geni [B,E], in quanto grazie ad essa verranno coperti 4 pazienti con tumore Z1 e 4 con tumore differente, rispetto alla coppia [D,E] (3 pazienti coperti con tumore Z1 e 3 con tumore differente).
I risultati del metodo ILP sono riassunti nella tabella 4.7. Anche in questo caso il numero di pazienti coperti e l'insieme dei geni soluzione risultano uguali al variare dei pesi utilizzati, per via della struttura della mutation matrix. Inoltre, i risultati descritti da ILP sono gli stessi di quelli ottenuti mediante l'utilizzo dell'algoritmo Greedy (4.5). Da ciò si

può constatare che su questo esempio l'algoritmo Greedy ha permesso di ottenere delle soluzioni ottime a livello globale, sempre se i risultati prodotti da ILP possono essere considerati anch'essi ottimi a livello globale. Si può concludere quindi che i due algoritmi su questo esempio permettono di ottenere risultati analoghi per il problema di coverage geni-pazienti, sebbene utilizzino due approcci prettamente differenti.

Tabella 4.6: Valore della funzione obiettivo per ogni possibile soluzione del problema ILP. (con $K = 2$)

		Combinazioni Soluzione									
Peso	T	*A,B*	*A,C*	*A,D*	*A,E*	*B,C*	*B,D*	*B,E*	*C,D*	*C,E*	*D,E*
N.	Z1	-4	-2	-3	-2	-2	-1	**0**	-2	-2	**0**
	Z2	-2	-4	**+1**	-2	-4	-5	-5	-4	-4	-4
	Z3	-2	-4	-3	-4	-4	**-1**	-2	-4	-4	-2
O. vs O.f.A.	Z1	-1/2	0	-5/12	-1/12	0	1/12	**+2/6**	0	0	+1/4
	Z2	+2/7	0	**+5/7**	+2/7	0	-11/21	-11/21	0	0	-8/21
	Z3	+2/7	0	-5/21	-4/21	0	**+3/7**	+2/7	0	0	+2/21
O. vs O.	Z1	-3/10	-2/10	-8/10	-3/10	-2/10	0	**4/10**	-2/10	-2/10	+3/10
	Z2	-8/10	-16/10	**+2/10**	-9/10	-16/10	-18/10	-22/10	-16/10	-16/10	-15/10
	Z3	-8/10	-16/10	-1	-15/10	-16/10	**-6/10**	-1	-16/10	-16/10	-9/10
O. vs. A.	Z1	-1/2	0	-10/24	-2/24	0	+2/24	**+8/24**	0	0	+6/24
	Z2	+1/4	0	**+13/24**	-1/24	0	-13/24	-16/24	0	0	-9/24
	Z3	+1/4	0	-7/24	-5/24	0	**+7/24**	+4/24	0	0	-1/24

Tabella 4.7: Risultati dell'esempio ILP

		Soluzioni		
Peso	T	*Geni*	# Paz. affetti da T	# Paz. non affetti da T
N.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4
O. vs O.f.A.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4
O. vs O.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4
O. vs. A.	Z1	B,E	4	4
	Z2	A,D	3	2
	Z3	B,D	3	4

4.4 Risultati

Per rilevare un insieme di geni in grado di migliorare la classificazione tumorale, gli algoritmi Greedy e ILP sono stati utilizzati con la matrice di Score-reduction (analizzata nel capitolo precedente), in quanto essa risulta la miglior scelta attuale per allenare un classificatore in termini di feature considerate e percentuale di predizioni corrette. (3.1). L'algoritmo Greedy (o ILP) è stato eseguito considerando tutti i tumori definiti all'interno di Score-reduction, in modo da ottenere alla fine un unico insieme F contenente i geni rilevati per ogni tumore. La cardinalità dell'insieme F varia in base alla regolazione del parametro k e dal numero di ripetizioni geniche che possono esistere tra i vari gruppi scoperti dagli algoritmi. In formule, la cardinalità di F può essere espressa come: $k \leq |F| \leq (\#\text{Tumori distinti}) * k$. L'insieme F viene utilizzato per ridurre il numero di feature di Score-reduction, in modo da ottenere una nuova matrice composta dai soli geni scoperti mediante l'algoritmo Greedy o ILP. A seguire è stato eseguito un test di classificazione con lo scopo di analizzare il comportamento predittivo di un classificatore allenato su questa matrice ridotta rispetto al classificatore allenato su Score-reduction. La procedura algoritmo Greedy/ILP, riduzione matrice Score-reduction e test di classificazione è stata eseguita 10 volte per ogni peso P e valore k considerato, al fine di ottenere un numero di dati rilevanti per lo studio.
Per l'algoritmo Greedy sono stati considerati i seguenti valori di k: 300, 100 e 27. Idealmente si potrebbero ottenere matrici ridotte con un numero di feature fino a 3300, 1100 e 297 rispettivamente. Queste dimensioni possono essere viste come una riduzione continua delle feature tra una matrice e l'altra di all'incirca 1/3, partendo da Score-reduction che contiene 9858 geni (la quale a sua volta contiene 1/3 delle feature della matrice di Score). La riduzione è stata effettuata per testare insiemi di geni con cardinalità diverse, in modo da trovare per le feature un intervallo significativo che permetta di allenare classificatori con una precisione accurata.
Nella tabella 4.8 vengono riportati i risultati delle varie prove di classificazione eseguite sulle matrici ridotte. Si può osservare come l'utilizzo dei pesi One vs One for All e One vs All hanno permesso di ottenere un insieme F in grado di allenare classificatori con una precisione sostanzialmente maggiore rispetto a quelli definiti dai pesi restanti (Naive e One vs One). Una possibile motivazione per questo comportamento potrebbe essere dovuta dal fatto che i pesi Naive e One vs One non sono bilanciati, con la conseguenza per l'algoritmo Greedy di selezionare gruppi di geni che non presentano molte mutazioni per i pazienti affetti dal tumore T considerato. Infatti, per un qualsiasi tipo tumorale T gli insiemi dei geni definiti da questi due pesi coprono un numero molto basso di pazienti in Score-reduction rispetto alla totalità di pazienti affetti da T, come viene descritto nelle tabelle 4.9 e 4.11. Viceversa, i pesi One vs One for All e One vs All hanno permesso di ottenere degli insiemi di geni in grado di coprire quasi l'interno gruppo di pazienti per ogni tipo tumorale T (tabelle 4.10, e 4.12). Questi risultati hanno dimostrato che per l'algoritmo Greedy i due pesi One vs One for All e One vs All sono nettamente migliori rispetto a Naive e One vs One.
Sempre dalla tabella 4.8 si è constatato che il migliore insieme F che permetta di ottenere buoni risultati a livello di accuratezza è quello ottenuto attraverso il peso One vs One for All con valore $k = 27$. Questo rappresenta un ottimo risultato in quanto si sono ottenuti classificatori con un accuratezza intorno al 57% considerando solo un insieme F di fea-

ture composto da 258 geni circa, rispetto al classificatore allenato su Score-reduction che presenta precisioni analoghe utilizzando 9858 feature. (3.1).

Tabella 4.8: Risultati medi ottenuti dall'analisi di 10 prove con Liblinear (risolutore R2-L1 e parametro costo C = 1) eseguite sulle matrici ridotte con $|F|$ feature, variabili a seconda del valore di k e peso P considerato nell'algoritmo Greedy.

Peso	k	$\|F\|$ **Media**	**Acc. Media**	**Var. Media**
N.	27	201.6	32.76	0.53
	100	294.0	32.54	0.19
	300	652.3	31.71	1.98
O. vs O.f.A.	27	258.0	57.34	1.47
	100	580.6	56.40	0.14
	300	965.7	56.67	0.93
O. vs O.	27	239.3	35.85	1.35
	100	459.0	35.43	1.17
	300	781.6	37.73	0.89
O. vs. A.	27	257.3	55.69	1.64
	100	591.7	55.47	0.66
	300	960.3	58.10	0.85

Tabella 4.9: Peso Naive: rapporto medio tra il numero di pazienti selezionati affetti da tumore T e il numero di pazienti selezionati affetti da tumore differente, a seconda dei valori di k e T considerati nell'algoritmo Greedy

		k		
T	**Rapporto Max.**	***27***	***100***	***300***
blca	*75/2592*	16.7/7.0	15.0/8.3	26.7/17.7
brca	*398/2269*	58.7/32.7	62.0/38.0	76.7/48.7
coadread	*385/2282*	334.3/74.7	331.3/81.0	335.7/87.0
gbm	*207/2460*	16.3/5.7	15.7/4.0	23.3/12.7
hnsc	*230/2437*	91.3/53.3	66.3/36.0	94.3/61.3
kirc	*374/2293*	242.3/40.3	260.3/56.7	256.7/52.3
laml	*154/2513*	40.3/10.3	42.3/13.0	43.0/14.3
luad	*173/2494*	57.3/12.3	59.0/11.7	76.3/31.7
lusc	*137/2530*	34.7/12.7	32.3/11.3	49.3/28.7
ov	*348/2319*	138.7/121.0	92.3/71.7	81.0/61.7
ucec	*186/2481*	127.7/59.7	127.7/65.7	127.7/65.0

Tabella 4.10: Peso One vs One for All: rapporto medio tra il numero di pazienti selezionati affetti da tumore T e il numero di pazienti selezionati affetti da tumore differente, a seconda dei valori di k e T considerati nell'algoritmo Greedy

		k		
T	**Rapporto Max.**	***27***	***100***	***300***
blca	*75/2592*	75.0/889.7	75.0/745.7	75.0/909.0
brca	*398/2269*	269.0/511.0	374.7/692.7	386.3/757.3
coadread	*385/2282*	372.0/240.7	372.7/223.7	372.7/230.0
gbm	*207/2460*	175.3/526.3	207.0/661.7	207.0/644.3
hnsc	*230/2437*	230.0/1185.0	230.0/1215.7	230.0/1206.0
kirc	*374/2293*	334.7/287.3	368.0/333.7	367.0/332.0
laml	*154/2513*	125.7/258.7	148.3/442.0	149.0/462.3
luad	*173/2494*	173.0/656.3	173.0/628.3	173.0/641.7
lusc	*137/2530*	137.0/1129.3	137.0/1108.7	137.0/1113.7
ov	*348/2319*	347.7/995.3	347.7/982.3	348.0/987.3
ucec	*186/2481*	185.0/372.0	186.0/501.3	186.0/304.0

Tabella 4.11: Peso One vs One: rapporto medio tra il numero di pazienti selezionati affetti da tumore T e il numero di pazienti selezionati affetti da tumore differente, a seconda dei valori di k e T considerati nell'algoritmo Greedy

		k		
T	**Rapporto Max.**	***27***	***100***	***300***
blca	*75/2592*	2.0/0.3	3.0/0.3	5.3/0.3
brca	*398/2269*	102.7/89.0	212.0/227.3	236.3/250.3
coadread	*385/2282*	352.3/119.0	358.3/119.7	353.0/120.7
gbm	*207/2460*	22.7/11.3	25.0/10.7	27.7/13.7
hnsc	*230/2437*	93.7/62.3	111.3/77.3	110.7/77.3
kirc	*374/2293*	275.0/78.7	309.3/117.7	295.3/105.7
laml	*154/2513*	39.0/9.7	41.0/10.7	39.0/8.7
luad	*173/2494*	51.3/11.3	51.7/11.3	54.3/12.0
lusc	*137/2530*	20.33/5.7	30.0/8.3	27.0/7.7
ov	*348/2319*	181.0/170.3	196.0/187.3	201.3/199.0
ucec	*186/2481*	57.7/13.7	67.0/22.3	69.0/18.7

Tabella 4.12: Peso One vs All: rapporto medio tra il numero di pazienti selezionati affetti da tumore T e il numero di pazienti selezionati affetti da tumore differente, a seconda dei valori di k e T considerati nell'algoritmo Greedy

		k		
T	**Rapporto Max.**	***27***	***100***	***300***
blca	*75/2592*	75.0/558.3	75.0/761.3	75.0/465.0
brca	*398/2269*	269.3/516.0	369.0/669.7	380.0/719.7
coadread	*385/2282*	373.0/210.0	372.7/183.7	371.0/210.0
gbm	*207/2460*	164.7/460.7	206.7/609.0	207.0/631.7
hnsc	*230/2437*	229.3/1176.3	230.0/1175.7	230.0/1198.3
kirc	*374/2293*	328.3/235.3	367.0/335.7	366.7/310.7
laml	*154/2513*	125.7/280.0	145.7/411.0	146.3/431.7
luad	*173/2494*	173.0/647.3	173.0/633.0	173.0/664.0
lusc	*137/2530*	137.0/1115.3	137.0/1110.0	137.0/1110.0
ov	*348/2319*	348.0/990.0	348.0/992.0	347.7/990.3
ucec	*186/2481*	186.0/498.0	186.0/390.7	186.0/383.7

Per cercare di migliorare ulteriormente la classificazione tumorale, si è deciso di valutare insiemi F con cardinalità più basse di quanto visto in precedenza. Per ottenere questi insiemi si è eseguito l'algoritmo ILP sulla matrice Score-reduction con $k = 15$ e utilizzando i 4 pesi P. Il valore esiguo di k ha permesso di utilizzare l'algoritmo ILP senza incorrere in nessun problema computazionale. Per un confronto più diretto con l'algoritmo Greedy, si è deciso di eseguire dei test anche con quest'ultimo utilizzando la stessa configurazione scelta per l'algoritmo ILP. I risultati riportati nella tabella 4.13 descrivono l'accuratezza dei vari classificatori allenati sulle matrici ridotte dagli insiemi di geni ottenuti dai due algoritmi, mentre la tabella 4.14 descrive la distribuzione dei pazienti coperti in relazione al tumore T e peso P considerato dai due algoritmi. Si è potuto osservare come i due algoritmi permettono di ottenere insiemi di geni del tutto simili tra loro, in quanto risulta simile il livello di precisione tra i vari classificatori. In questo caso si è potuto constatare che le soluzioni ottenute mediante l'utilizzo dell'algoritmo Greedy possono essere equiparabili alla soluzione ottimale globale ottenuta dall'algoritmo ILP (se raggiunta). L'unica differenza sostanziale tra i due algoritmi ricade sul tempo di esecuzione: come era lecito aspettarsi, ILP impiega un tempo 4 volte superiore rispetto all'algoritmo Greedy per analizzare le informazioni contenute nella matrice di Score-reduction.
L'abbassamento del valore di k non ha migliorato la classificazione tumorale, ma ha permesso di ridurre il numero di feature interessanti. Infatti, il classificatore con $k = 27$ e peso One vs One for All considera 258 geni circa per effettuare predizioni con un accuratezza intorno al 57%, mentre in questo caso si ottiene la stessa precisione considerando solamente un insieme composto da circa 160 geni. In questo modo si è ristretto ulteriormente il campo di ricerca per un gruppo esiguo di geni da poter utilizzare per effettuare predizioni accurate sui vari tipi tumorali considerati.

I due algoritmi Greedy e ILP hanno permesso di ottenere dei risultati rilevanti per la riduzione del numero delle feature di una mutation matrix. Si è deciso quindi di confrontarli con un terzo algoritmo incluso tra i risolutori di Liblinear: L1-regularized L2-loss support vector classification (d'ora in poi R1-L2) permette a SVM di tralasciare

Tabella 4.13: Risultati medi ottenuti dall'analisi di 10 prove con Liblinear (risolutore R2-L1 e parametro costo C = 1) eseguite sulle matrici ridotte ottenute mediante gli algoritmi ILP e Greedy , in base al peso P e con parametro fisso $k = 15$

		N.		***O. vs O.f.A.***		***O. vs. O.***		***O. vs A.***	
	Tempo esec.	**Acc.**	$\|F\|$	**Acc.**	$\|F\|$	**Acc.**	$\|F\|$	**Acc.**	$\|F\|$
ILP	∼400m.	32.88	129.0	58.85	161.0	34.79	136.3	57.64	160.0
Greedy	∼100m.	31.71	145.3	57.00	156.0	34.00	154.6	56.21	155.3

Tabella 4.14: Rapporto medio tra il numero di pazienti selezionati affetti da tumore T e il numero di pazienti selezionati affetti da tumore differente, in base al peso P e al parametro fisso $k = 15$ considerati negli algoritmi Greedy e ILP

		N.		***O. vs. O.f.A.***		***O. vs. O.***		***O. vs. A.***	
T	**Rapporto Max.**	**ILP**	**Gr.**	**ILP**	**Gr.**	**ILP**	**Gr.**	**ILP**	**Gr.**
blca	*75/2592*	25.7/14.3	16.7/7.3	74.3/281.7	75.0/600.7	3.3/0.3	3.7/0.0	74.0/305.7	75.0/748.3
brca	*398/2269*	59.0/38.3	59.7/33.0	226.0/455.7	239.3/465.3	101.3/99.0	86.7/73.0	215.0/409.3	236.3/454.3
coadread	*385/2282*	322.7/77.0	322.3/75.7	364.3/212.0	368.7/223.7	339.3/100.0	338.6/99.3	362.3/201.0	365.7/196.3
gbm	*207/2460*	17.7/6.0	14.7/3.3	144.0/427.3	147.3/465.3	20.7/10.0	24.0/12.0	142.3/482.3	146.3/462.7
hnsc	*230/2437*	78.3/50.7	79.0/53.3	188.3/489.0	220.3/1194.3	88.0/63.7	83.3/56.7	184.7/453.3	214.7/1181.7
kirc	*374/2293*	238.3/47.3	260.7/56.7	310.0/247.3	313.0/235.7	261.7/76.0	268.7/62.0	311.0/265.0	306.3/193.7
laml	*154/2513*	24.7/7.3	40.0/10.3	111.3/228.7	111.3/236.3	10.7/3.3	40.0/8.3	111.7/234.0	113.0/249.3
luad	*173/2494*	47.7/13.7	47.0/8.7	154.7/488.0	162.6/563.6	46.3/8.3	53.0/11.0	158.7/499.3	163.3/614.7
lusc	*137/2530*	14.7/3.3	26.0/8.7	128.0/588.0	137.0/1130.3	25.0/9.0	21.3/4.7	123.0/496.3	137.0/1122.7
ov	*348/2319*	136.3/112.0	37.3/24.0	348.0/974.3	348.0/1002.6	162.7/141.0	168.3/148.3	348.0/984.7	348.0/991.3
ucec	*186/2481*	99.0/52.7	105.6/56.3	175.0/309.0	172.3/248.0	54.7/16.0	45.0/8.7	174.3/306.7	177.3/365.0

tutte le feature non ritenute necessarie per la predizione, a seconda del valore assegnato al parametro C. Valori bassi di C inducono SVM a essere più selettivo nei riguardi delle feature selezionabili, ottenendo di conseguenza insiemi con cardinalità piccole. Viceversa, valori alti per C permetteno di ottenere insiemi di feature interessanti con cardinalità maggiori. L'iperpiano definito da SVM mediante l'utilizzo di R1-L2 (in forma primale) può essere osservato in 4.2 (dove $\|.\|_1$ corrisponde alla norma di uno spazio l^1).

$$\underset{\mathbf{w}}{\text{minimize}} \quad \|\mathbf{w}\|_1 + C\sum_{i=1}^{N}(\max(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i))^2 \tag{4.2}$$

La tabella 4.15 descrive i risultati medi ottenuti da 10 prove usando Liblinear con risolutore R1-L2 sulla matrice di Score-reduction, in base a diverse configurazioni per il parametro C. Si può osservare che il risolutore R1-L2 non ha permesso di raggiungere sostanziali miglioramenti rispetto ai classificatori analizzati in precedenza. Mediante il parametro $C = 0.035$ si sono ottenuti classificatori definiti su un insieme di feature pari a 157, in grado di effettuare predizioni corrette con un'accuratezza intorno al 59%. In termini di precisione e numero feature considerate, questi classificatori risultano simili con quelli allenati sulle matrici ridotte dagli algoritmi Greedy e ILP con peso One vs One for All e $k = 15$.
Questo risultato ha permesso di evidenziare come 3 algoritmi sostanzialmente diversi per la ricerca di feature interessanti(Greedy, ILP, SVM R1-L2) permettono di ottenere classificatori con una precisione simile tra loro considerando insiemi di feature pari a circa 160 geni. Di conseguenza, gli insiemi devono avere in comune un determinato numero di geni che ricoprono un ruolo rilevante per le predizioni tumorali. Per identificare questo insieme di geni in comune si è deciso di analizzare prima di tutto il numero di geni che

compaiono spesso nell'insieme F delle feature secondo i diversi algoritmi. Questo permette di isolare i geni che vengono selezionati in modo costante dagli algoritmi durante diverse esecuzioni. In questo caso per ogni algoritmo sono stati eseguiti 6 test per ottenere diversi insiemi di feature F tutti con cardinalità pari a 160. La frequenza di apparizione nei vari insiemi per tutti i geni ottenuti dai 3 algoritmi viene descritta dalla figura 4.1. Si è scoperto che per ogni algoritmo esiste un numero consistente di geni che vengono sempre selezionati in tutte e 6 i test eseguiti: essi corrispondo a 83, 35 e 24 geni per gli algoritmi SVM (R1-L2), Greedy e ILP rispettivamente. Il divario considerevole che sussiste tra questi gruppi potrebbe essere dovuto dal modo in cui le feature vengono selezionate da SVM(R1-L2) rispetto al Greedy e ILP. L'analisi congiunta dei 3 gruppi ha permesso di identificare ben 20 geni in comune tra loro. Questo risultato è importante in quanto ha permesso di dedurre che gli algoritmi, durante la creazione dell'insieme delle feature F, selezionano sempre 20 geni particolari considerati necessari per effettuare predizioni tumorali accurate. Non meno importati sono i geni in comune tra le coppie di gruppi ma non tra tutti e 3 i gruppi: ILP e SVM (R1-L2) hanno in comune un solo gene, mentre tra Greedy e SVM (R1-L2) ne sono stati scoperti 8. La suddivisione completa dei 3 gruppi di geni è riportata in Figura 4.2.

Tabella 4.15: Media su 10 prove di Liblinear eseguite sulla matrice di Score-reduction con risolutore R1-L2 e parametro costo C variabile.

C	**Geni**	**Acc. Media**	**Var. Media**
10	4265.0	54.87	1.84
1	3151.3	61.25	0.84
0.1	921.7	63.65	1.28
0.05	310.0	61.36	1.48
0.035	157.6	59.10	1.46
0.01	21.3	44.48	1.37

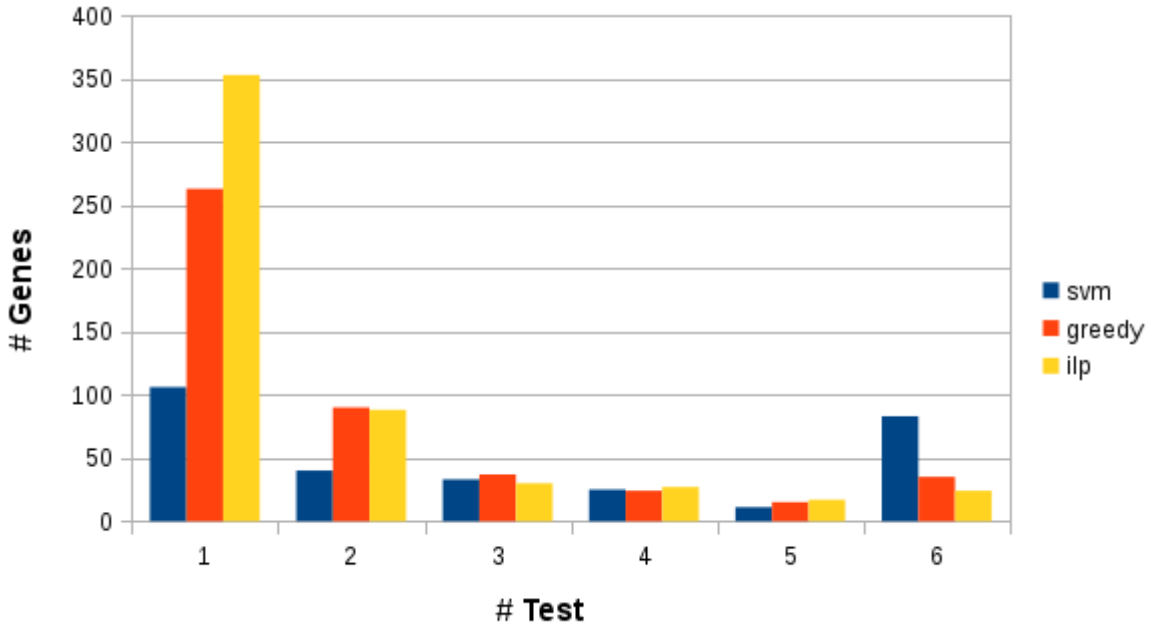

Figura 4.1: Analisi della distribuzione di frequenza per i geni ottenuti da 6 test eseguiti con i 3 metodi ILP(O. vs O.f.A. , K=15), Greedy(O. vs O.f.A. , K=15) e SVM(R1-L2, C=0.035).

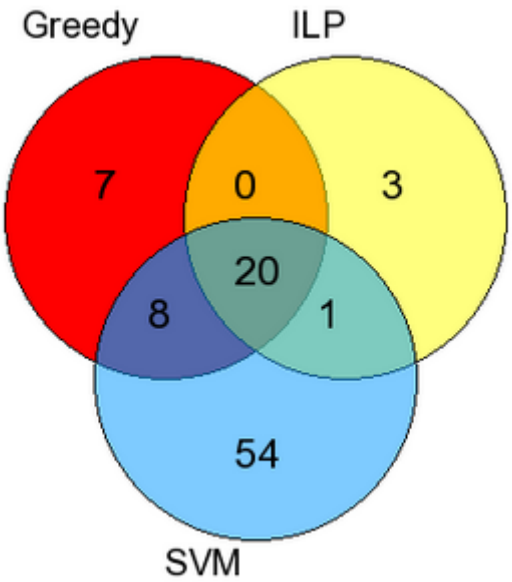

Figura 4.2: Suddivisione dei seguenti gruppi di geni mediante diagramma di Venn: SVM(R1-L2) 83 geni ; Greedy 35 geni ; ILP 24 geni

Mediante questo processo di raffinazione delle feature sono stati scoperti 29 geni potenzialmente interessanti per la predizione tumorale, riportati di seguito per completezza:

- APC
- BAP1
- BRAF
- BRCA1
- BRCA2
- CDH1
- CDKN2A
- CTNNB1
- DNMT3A
- EGFR
- FLT3
- FRG1B
- GATA3
- IDH1
- KDM6A
- KRAS
- MAP3K1
- MUC4
- NOTCH1
- NPM1
- NRAS
- PBRM1
- PIK3CA
- PPP2R1A
- PTEN
- RUNX1
- STK11
- TP53
- VHL

Grazie alle conoscenze attuali nel campo biologico si è potuto constatare che ben 27 geni dei 29 ottenuti sono considerati rilevanti per lo sviluppo tumorale in quanto le eventuali mutazioni che possono presentare sono classificate come mutazioni di tipo driver, secondo le informazioni dal materiale supplementare (tabella S2A) dell'articolo Cancer Genome Landscape[10], reperibile nel sito `http://science.sciencemag.org/content/suppl/2013/03/27/339.6127.1546.DC1?_ga=1.116182469.2126799401.1458945690`. I due geni rimanenti, FRG1B e MUC4, attualmente non sono riconosciuti come geni di tipo driver, ma ciò non esclude la possibilità di essere in qualche modo relazionati allo sviluppo tumorale.
In definitiva questo gruppo di 29 geni presenta delle caratteristiche interessanti: sono quasi tutti di tipo driver e allo stesso tempo fanno parte dei migliori geni che presentano mutazioni frequenti solo per una serie di pazienti affetti da un determinato tipo tumorale. Di conseguenza potrebbe potenzialmente esistere una sorta di correlazione tra il tipo di mutazione (driver, passenger) e la frequenza di mutazione per un gene all'interno di un gruppo di pazienti affetti dallo stesso tipo tumorale.
Per valutare l'effettivo miglioramento di questo gruppo nei confronti della precisione di un classificatore, si è deciso di ridurre le feature di Score-reduction in modo tale da ottenere una matrice ridotta formata dai soli 29 geni. In seguito su tale matrice sono stati eseguiti diversi test di classificazione mediante Liblinear con risolutore R2-L1, ottenendo dei classificatori con una precisione intorno al 55%(tabella 4.16). Questo rappresenta un risultato importante in quanto è stato possibile ottenere una precisione analoga a quella dei classificatori allenati su 160 feature analizzati in precedenza. Per consolidare questo risultato sono stati allenati 1000 classificatori su matrici contenenti 29 feature selezionate in modo randomico da Score-reduction. Dalla figura 4.3 si può osservare la distribuzione di accuratezza dei test eseguiti. Come si voleva dimostrare, l'utilizzo di 29 geni scelti a caso permette di ottenere classificatori con una precisione molto bassa, che non si discosta molto dall'idea di effettuare predizioni tumorali in modo completamente casuale. Di

conseguenza, il gruppo formato da 29 geni può essere considerato il miglior insieme di geni ottenuto in questo studio che permetta di ottenere risultati rilevanti per la predizione tumorale mediante tecniche di machine learning.

Tabella 4.16: Media su 10 prove di Liblinear eseguite su una matrice descritta da 29 geni di tipo driver, con risolutore R2-L1 e parametro costo C = 1.

	Accuratezza media (%)									
Tumore	*P.1*	*P.2*	*P.3*	*P.4*	*P.5*	*P.6*	*P.7*	*P.8*	*P.9*	*P.10*
blca	48.0	32.0	36.0	48.0	56.0	32.0	52.0	40.0	56.0	32.0
brca	44.7	47.73	45.45	46.97	46.21	47.73	44.7	45.45	40.15	46.97
coadread	86.72	86.72	83.59	90.63	84.38	85.16	78.91	87.5	86.72	85.16
gbm	20.29	23.19	20.29	20.29	20.29	21.74	27.54	26.09	20.29	18.84
hnsc	36.84	48.68	47.37	42.11	36.84	39.47	38.16	51.32	34.21	46.05
kirc	73.6	74.4	71.2	71.2	69.6	72.8	71.2	69.6	68.8	71.2
laml	54.9	58.82	62.75	66.67	58.82	64.71	60.78	62.75	62.75	50.98
luad	21.05	17.54	21.05	24.56	15.79	19.3	19.3	15.79	21.05	22.81
lusc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ov	87.07	91.38	87.07	87.07	81.9	93.97	92.24	87.93	88.79	86.21
ucec	53.23	48.39	56.45	59.68	54.84	54.84	59.68	51.61	50.0	59.68
Tutti	55.24	56.82	55.80	57.60	54.11	56.70	55.91	56.48	54.34	55.46

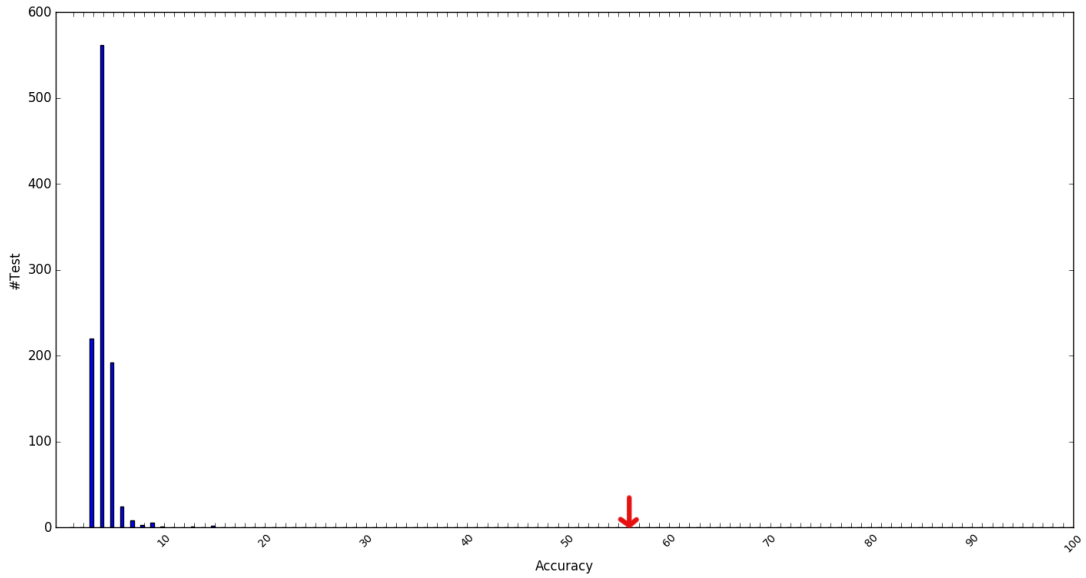

Figura 4.3: Ditribuzione accuratezza per 1000 classificatori allenati con 29 geni scelti a caso. La freccia indica l'accuratezza per il classificatore allenato sui 29 geni di tipo driver scoperti con i metodi ILP/Greedy.

Capitolo 5

Conclusioni

In questo studio sono stati proposti diversi metodi per la ricerca di un gruppo ristretto di geni da considerare come un punto di riferimento per la predizione tumorale, basata sulle sole mutazioni genomiche di un individuo.
Per ricercare un tale insieme sono state applicate diverse tecniche di machine learning (SVM e Random Forest) su mutation matrix, in modo da analizzare la rilevanza delle varie feature che definisco la matrice. La mutation matrix di partenza (Score) contiene le informazioni mutagene dell'intero corredo genetico (25000 geni circa) di 3500 pazienti, suddivisi secondo 11 tumori differenti. Dai primi test di classificazione eseguiti sulla matrice di Score si sono ottenuti classificatori con una precisione intorno al 61%, un valore di partenza tutto sommato positivo dato che nei test di permutazione l'accuratezza media è risultata essere 12%.
Con lo scopo di migliorare questo risultato, sono stati ideati due nuovi metodi (Kernel e Hotnet2) che hanno permesso di sfruttare congiuntamente le informazioni contenute in una interaction network e le informazioni mutagene della mutation matrix. L'interaction network considerata è HINT+HI2012, descritta da 9858 geni e 40704 interazioni tra essi. Per applicare i due metodi è stato necessario ridurre il numero di feature di Score, in modo da far coincidere la cardinalità dell'insieme di geni decritti nella matrice con quelli presenti nell'interaction network. Dai test di classificazione eseguiti sulla matrice risultante Score reduction (9858 geni), sono stati ottenuti classificatori con un'accuratezza pari a circa 59%. Questo risultato ha permesso di eliminare i 2/3 delle feature di Score, in quanto esse non danno nessun contributo per la predizione tumorale. Invece non sono stati osservati sostanziali miglioramenti dai test di classificazione eseguiti sulle matrici risultati dai metodi Kernel e Hotnet2, rispetto a quanto ottenuto utilizzando Score-reduction. Di conseguenza si è constatato che per questo studio le informazioni topologiche di una interaction network non permettono di migliorare in modo sostanziale la classificazione tumorale.
L'ultimo metodo proposto utilizza solo le informazioni mutagene ed è basato sul maximum coverage problem. Esso permette di rilevare un insieme di geni le cui mutazioni sono frequenti solo nei pazienti affetti da un preciso tipo tumorale. Questo metodo è stato implementato secondo due diversi paradigmi: ILP e Greedy. Entrambi sfruttano la definizione di 4 pesi (Naive, One vs One for All, One vs One, One vs All), ideati per valutare l'importanza di un gene in relazione alle sue mutazioni presenti in un gruppo di individui. Dai vari test eseguiti è stato appurato che gli algoritmi sono simili tra loro,

in quanto presentano risultati analoghi. Il risultato migliore è stato ottenuto utilizzando insiemi di geni composti da 160 geni (ottenuti da ILP/Greedy), utilizzati per ridurre le feature di Score-reduction. I classificatori allenati su queste matrici ridotte presentano una precisione intorno al 57%. Rispetto alla matrice di Score iniziale (25000 geni) sono stati compiuti passi in avanti nella riduzione del numero di feature, mantenendo allo stesso tempo l'accuratezza dei classificatori pressoché invariata.
Si è constato che tra i 160 geni ne esiste un numero ristretto che viene sempre selezionato tra diverse esecuzioni degli algoritmi ILP e Greedy. Lo stesso numero ristretto compare anche in altri insiemi (sempre da 160 feature), ottenuti dall'analisi della matrice di Score-reduction attraverso un risolutare SVM con proprietà di feature selection. Questo gruppo è composto da 29 geni, 27 dei quali sono considerati geni di tipi driver. Per i restanti due geni (FRG1B e MUC4) attualmente non si conosce una risposta certa. L'insieme dei 29 geni è stato utilizzato per ridurre ulteriormente la matrice di Score-reduction. I test di classificazione eseguiti su di essa sono stati positivi in quanto si è ottenuta la stessa precisione (55% circa) dei classificatori analizzati in precedenza. Il gruppo di 29 geni rappresenta il miglior insieme di cardinalità minima ottenuta, contenente solo geni rilevanti per le predizioni tumorali. Inoltre si è constato che esiste una sorta di relazione tra il tipo di mutazione di un gene e la sua frequenza di mutamento all'interno di un gruppo di individui affetti dallo stesso tipo tumorale.
In conclusione, questo studio ha permesso di dimostrare che risulta possibile ideare nuovi approcci in grado di eseguire tecniche di machine learning con lo scopo di effettuare delle precisioni tumorali accurate, sulla base dell'eventuale presenza di una mutazione per i geni. Un'applicazione del genere potrebbe essere utile in futuro per eseguire analisi tumorali a basso costo, in modo da diagnosticare per tempo un possibile sviluppo tumorale per gli individui.

Tabella 5.1: Riassunto dei vari risultati migliori ottenuti nel corso dell'opera, mediante classificazione Liblinear con risolutore R2-L1 e parametro C=1 (escluso dove indicato diversamente)

Classificatori allenati su	**Descrizione**	**Num.Geni**	**Acc. Media**
Score	Mutation matrix di partenza	24799	60.83%
Score-reduction	Riduzione da Score. Contiene i geni della interaction network	9858	59.71%
K-adiacenza	Applicazione Kernel tra Score-reduction e matrice di adiacenza della interaction network. Parametro $\beta = 0.1$ (influenza rete bassa)	9858	59.64%
k-influenza	Applicazione Kernel tra Score-reduction e matrice di influenza con restar $\sigma = 0.40$. Parametro $\beta = 0.1$ (influenza rete bassa) e simmetria massima	9858	61.32%
HotScore-003	Riduzione da Score-reduction. Contiene i geni interessanti scoperti da Hotnet2. Valore p-value di soglia $\alpha = 0.03$	808	52.57%
Matrix-coverage	Riduzione da Score-reduction. Contiene i geni interessanti scoperti dall'algoritmo ILP (Greedy). Peso One vs One for All e $K = 15$.	160	58.85%
Score-reduction(R1-L2)	Uso del risolutore R1-L2 per eliminare feature superflue da Score-reduction. C=0.035	158	59.10%
Matrix-driver	Riduzione da Score-reduction. Contiene un insieme comune di geni tra Matrix-coverage e Score-reduction(R1-L2)	29	55.85%

Bibliografia

[1] *Learning From Data by Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin (2012).* AMLBook, 2012.

[2] Global cancer facts and figures 3rd edition. *Atlanta, American Cancer Society*, 2015.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition (MIT Press).* The MIT Press, 3rd edition, 7 2009.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] Rong E. Fan, Kai W. Chang, Cho J. Hsieh, Xiang R. Wang, and Chih J. Lin. LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[6] Tin K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, August 1998.

[7] O. Lavi, G. Dror, and R. Shamir. Network-induced classification kernels for gene expression profile analysis. *J. Comput. Biol.*, 19(6):694–709, Jun 2012.

[8] M. D. Leiserson, F. Vandin, H. T. Wu, J. R. Dobson, J. V. Eldridge, J. L. Thomas, A. Papoutsaki, Y. Kim, B. Niu, M. McLellan, M. S. Lawrence, A. Gonzalez-Perez, D. Tamborero, Y. Cheng, G. A. Ryslik, N. Lopez-Bigas, G. Getz, L. Ding, and B. J. Raphael. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet.*, 47(2):106–114, Feb 2015.

[9] M. Meyerson, S. Gabriel, and G. Getz. Advances in understanding cancer genomes through second-generation sequencing. *Nat. Rev. Genet.*, 11(10):685–696, Oct 2010.

[10] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler. Cancer genome landscapes. *Science*, 339(6127):1546–1558, Mar 2013.

Ringraziamenti

Vorrei ringraziare i miei genitori che con i loro sacrifici e il loro amore mi hanno permesso di raggiungere questo obiettivo importante.

Grazie a mia sorella Lisa che fin dalla sua nascita non ha fatto altro che rendere migliori tutti i giorni della mia vita.

Un grazie a tutti i membri della mia grande famiglia, siete il mio tesoro più prezioso.

Ai miei amici più cari, con cui ho condiviso momenti meravigliosi.

Ringrazio il professore Fabio Vandin che mi ha seguito in questo lavoro di tesi con passione e dedizione.

Infine vorrei fare un ringraziamento speciale a mio nonno Gino che ha sempre creduto nelle mie capacità fin da quando ero bambino, convinto che in futuro sarei riuscito a raggiungere un traguardo come questo.