Tecniche di riconoscimento statistico

Teoria ed applicazioni industriali

Parte 6 – Riduzione features

Ennio Ottaviani
On AIR srl
ennio.ottaviani@onairweb.com
http://www.onairweb.com/corsoPR

A.A. 2018-2019

Complessità di un classificatore

- All'aumentare del numero di *n* features, il progetto di un classificatore presenta alcune problematiche legate ad *n*
  - Complessità computazionale
  - Fenomeno di Hughes

- La complessità computazionale è legata al fatto che il numero di operazioni elementari per classificare un pattern incognito *x* dipende da una potenza di *n*

- L'aumento di complessità porta ad aggravi di tempo/memoria spesso non tollerabili nelle applicazioni

Fenomeno di Hughes

- Ci si aspetta che al crescere di *n* aumenti l'informazione passata al classificatore, e di conseguenza aumentino le prestazioni …
- … invece questo in genere accade solo fino ad un certo valore critico di *n*. Aumenti successivi portano ad un peggioramento
- Questo accade perché al crescere di *n* cresce anche il numero di parametri da stimare, per cui oltre un certo valore, la dimensione fissa del training set non è più adeguata, e le stime diventano meno affidabili

Riduzione delle features

- Nella pratica si preferisce limitare il numero di features *n* ad un valore massimo dipendente dalla applicazione

- La riduzione della dimensione dello spazio porta comunque ad una perdita di informazione, che occorre minimizzare

- Esistono due strategie di base per ridurre *n*:
  - Selezione: ricerca di un sottoinsieme delle features di partenza in modo da ottenere comunque prestazioni sufficienti
  - Estrazione: produzione di nuove features attraverso combinazioni (spesso lineari) in uno spazio a dimensioni ridotta

Selezione delle features

- Impostazione del problema: dato un insieme di features $X=x_1 \ldots x_n$ trovare un sottoinsieme S di dimensione *m* contenuto in X tale da ottimizzare un opportuno criterio (es. l'errore di classificazione)

- Diversi algoritmi di selezione possono essere realizzati al variare del criterio di selezione e della strategia di generazione dei sottoinsiemi

- Una generazione esaustiva non pensabile (per *n* grandi), a causa della complessità combinatorica del problema. S preferiscono strategie di tipo subottimo

Selezione sequenziale

- La strategia di selezione sequenziale in avanti (sequential forward search, SFS) si basa sui passi seguenti:

- Si inizializza S=Ø
- Si calcola uno score (es. errore) per tutti gli insiemi del tipo S U x_i con i=1…n
- Si aggiunge a S la feature migliore
- Si ripete aggiungendo una alla volta nuove features ad S fino al raggiungimento del totale desiderato (od altro criterio di terminazione)

Osservazioni su SFS

- Il metodo è di solito computazionalmente molto efficiente, anche partendo da *n* grande, se *m* è piccolo
- SFS non permette backtracking, cioè una features inserita in S non può più essere rimossa
- L'insieme ottenuto non ha nessuna garanzia di essere quello ottimale
- E' possibile una versione backward (SBS) utilizzabile qualora *m* sia molto vicino ad *n*

Distanza di Bhattacharyya

- A volte si preferisce non utilizzare un criterio di selezione basato sull'errore di classificazione, che richiede di addestrare ogni volta un classificatore diverso, ma si usano criteri più semplici

- La distanza di Bhattacharyya misura quanto un insieme di features è in grado di discriminare tra 2 classi

$$B(S) = -\ln \beta(S) \qquad \beta(S) = \int_{\mathbb{R}^m} \sqrt{p(\mathbf{x}^S \mid \omega_1) p(\mathbf{x}^S \mid \omega_2)}\, d\mathbf{x}^S$$

- Tramite *B* è possibile stabilire limiti inferiori e superiori alla minima probabilità di errore ottenibile con qualunque classificatore (nel caso a 2 classi)

Limite di Bhattacharyya

- Si può dimostrare che, date le probabilità a priori P_1 e P_2 e definito il limite di Bhattacharyya

$$\varepsilon_u = \sqrt{P_1 P_2}\,\exp(-B)$$

la minima probabilità di errore P_e risulta limitata da

$$\varepsilon_u^2 \leq P_e \leq \varepsilon_u$$

- Dovendo selezionare un insieme S (al limite sola 1 feature) per un problema di classificazione, un criterio valido è massimizzare la distanza B utilizzando stime delle probabilità coinvolte

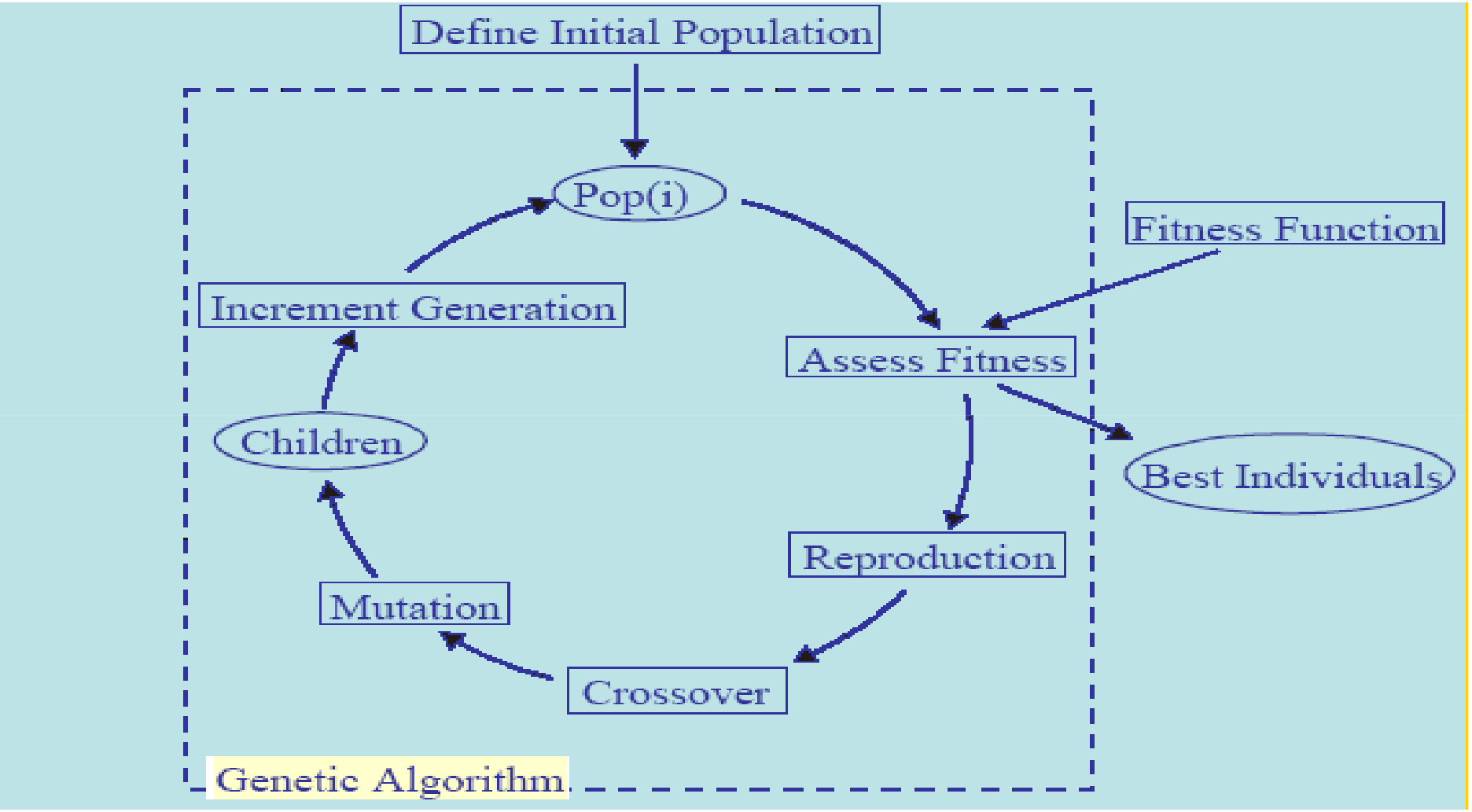
Algoritmi genetici

- I GA si ispirano direttamente alla teoria darwiniana, mutuandone i concetti di base nel campo della ottimizzazione
- Si parte da una popolazione di individui (un sottoinsieme dello spazio delle possibili soluzioni del problema)
- Si definisce una funzione di fitness in grado di descrivere quantitativamente la bontà di una soluzione (cioè il suo grado di adattamento al problema)
- Con un processo iterativo (generazioni successive) vengono selezionate le soluzioni migliori e ad esse vengono applicate variazioni con l'intento di migliorarne la fitness

Algoritmi genetici (2)

- La selezione è condotta secondo i principi darwiniani di sopravvivenza e competizione
- La variazione è realizzata tramite gli operatori di crossover (ricombinazione) e mutazione
  - Il crossover emula la riproduzione, combinando due soluzioni in modo da produrne altre due con caratteristiche mescolate
  - La mutazione altera casualmente le caratteristiche della soluzione che la subisce
- La popolazione evolve nel tempo ottimizzando la fitness per selezione naturale, esplorando diverse regioni dello spazio delle soluzioni

Algoritmi genetici (3)



Modelli a sparsità controllata

- In statistica, un modello è “sparso” quando pochi parametri sono realmente importanti per ottenere buoni risultati
- Il concetto diventa cruciale quando il numero di parametri *p* del modello sovradetermina i dati ($p > N$)
- Esempio: in genetica si classificano le potenziali malattie in base alle diverse espressioni geniche. Di norma p=1000:10000, N~100
- Una riduzione implicita delle features (e quindi dei parametri del modello) si ottiene con tecniche di regolarizzazione, sviluppate nel contesto della regressione, ma applicabili anche alla classificazione

Regolarizzazione tramite LASSO

- Consideriamo un classificatore/regressore di tipo lineare

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + e_i,$$

- I pesi β si ottengono tramite minimizzazione dell'errore q.m. Il LASSO impone un vincolo extra

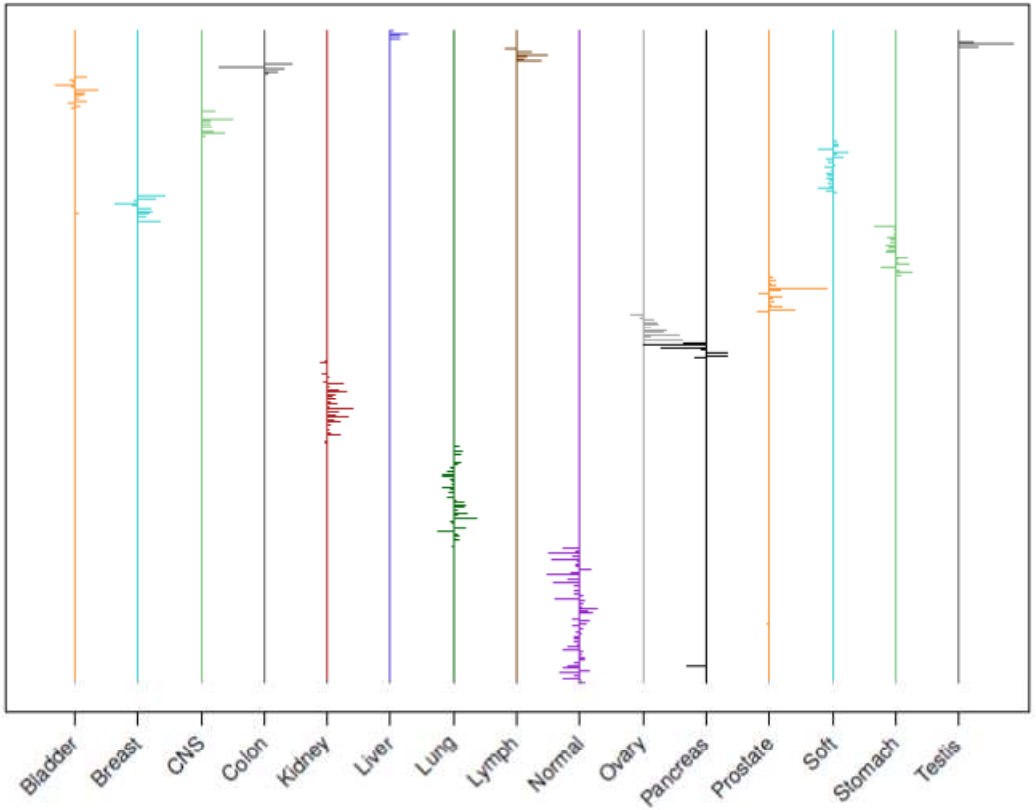
$$\underset{\beta_0,\beta}{\text{minimize}} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \text{ subject to } \|\beta\|_1 \leq t$$

- L'suo della norma L1 complica gli aspetti computazionali, pur rimanendo nell'ambito dei problemi di tipo convesso, ma genera soluzioni in cui molti *β* sono nulli

- Assumiamo quindi che le feature con peso non nullo siano quelle importanti

Esempio di uso del LASSO

- 15 classi (tipologie di cancro
- 4718 espressioni geniche (features)
- 349 pazienti
- LASSO: solo 254 geni hanno peso ≠ 0 per almeno una classe



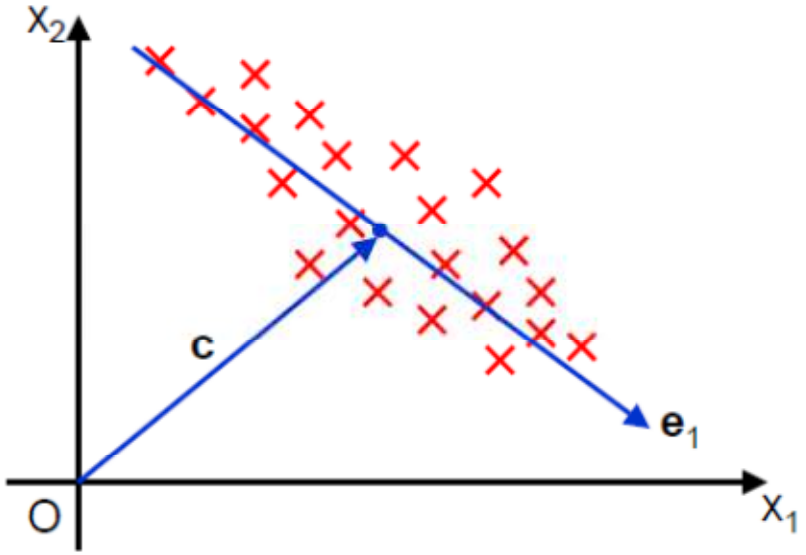
Estrazione delle features

- Impostazione del problema: dato un insieme di features $X=x_1 \ldots x_n$ trovare una trasformazione T che generi un insieme di features $Y=y_1 \ldots y_m$ tale da ottimizzare un opportuno criterio
- Se la trasformazione *T* è lineare, abbiamo in generale $y = Tx$ con *T* matrice *mxn*
- Vengono spesso usate trasformazioni ortonormali, in modo da proiettare *X* su un opportuno sottospazio sotteso da *m* assi ortogonali
- Esistono diverse scelte possibili per la matrice *T*, di efficacia e complessità molto diverse

Analisi delle componenti principali

- L'analisi delle componenti principale (PCA o trasformata di Karhunen-Loewe) è un metodo non supervisionato di estrazione delle features basato su una semplice misura di distorsione

- Si descrive lo spazio delle features in base ad un insieme di n vettori ortonormali $e_1 \ldots e_n$, e si riduce la dimensione selezionando un numero *m* di tali vettori (ed una costante additiva *c*)



Analisi delle componenti principali

- Scartando *n-m* assi si commette un errore, che va minimizzato, per cui il criterio adottato dalla PCA per la scelta è lo scarto quadratico medio

$$\mathcal{J} = \frac{1}{N}\sum_{k=1}^{N}\|\mathbf{x}_k - \tilde{\mathbf{x}}_k\|^2 = \frac{1}{N}\sum_{k=1}^{N}\left\|\mathbf{x}_k - \mathbf{c} - \sum_{i=1}^{m} y_{ik}\mathbf{e}_i\right\|^2$$

- Tenendo conto della ortonormalità dei vettori *e*, si ha

$$\mathcal{J} = \frac{1}{N}\sum_{k=1}^{N}\left[\|\mathbf{x}_k - \mathbf{c}\|^2 - 2\sum_{i=1}^{m} y_{ik}\mathbf{e}_i^t(\mathbf{x}_k - \mathbf{c}) + \sum_{i=1}^{m} y_{ik}^2\right]$$

Analisi delle componenti principali

- Per minimizzare la distorsione J si impone la stazionarietà rispetto alle componenti y_{ik} ottenendo

$$\frac{\partial \mathcal{J}}{\partial y_{ik}} = 0 \Rightarrow y_{ik} = \mathbf{e}_i^t(\mathbf{x}_k - \mathbf{c}), \quad i = 1,2,...,m, \quad k = 1,2,...,N$$

$$\mathcal{J} = \frac{1}{N}\sum_{k=1}^{N}\|\mathbf{x}_k - \mathbf{c}\|^2 - \sum_{i=1}^{m}\mathbf{e}_i^t\Sigma\mathbf{e}_i \qquad \Sigma = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{x}_k - \mathbf{c})(\mathbf{x}_k - \mathbf{c})^t$$

- J si decompone quindi in due termini, uno dipendente dal centro c e l'altro dalla covarianza Σ

Analisi delle componenti principali

- La distorsione risulta quindi minima quando

$$\min_{\mathbf{c}} \frac{1}{N}\sum_{k=1}^{N}\|\mathbf{x}_k-\mathbf{c}\|^2 \Rightarrow \mathbf{c}=\frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k \equiv \boldsymbol{\mu}$$

$$\begin{cases} \max_{\mathbf{e}_i} \mathbf{e}_i^t \Sigma \mathbf{e}_i \\ \|\mathbf{e}_i\|^2 = \mathbf{e}_i^t\mathbf{e}_i = 1 \end{cases} \qquad (\Sigma-\lambda_i I)\mathbf{e}_i = \mathbf{0}$$

- La matrice di covarianza è simmetrica quindi gli autovalori λ_i sono tutti reali e positivi (o nulli)

Analisi delle componenti principali

- Inserendo il risultato ne segue che il valore minimo di J vale

$$\mathcal{J}^* = \frac{1}{N}\sum_{k=1}^{N}\|\mathbf{x}_k - \boldsymbol{\mu}\|^2 - \sum_{i=1}^{m}\lambda_i \mathbf{e}_i^t \mathbf{e}_i = \frac{1}{N}\sum_{k=1}^{N}\|\mathbf{x}_k - \boldsymbol{\mu}\|^2 - \sum_{i=1}^{m}\lambda_i$$

- Esso viene assunto quando gli autovalori selezionati sono quelli più grandi, e la trasformazione risultante è quindi

$$y_{ik} = \mathbf{e}_i^t(\mathbf{x}_k - \boldsymbol{\mu}) \quad \forall i,k \Rightarrow \mathbf{y}_k = T(\mathbf{x}_k - \boldsymbol{\mu})$$

Riepilogo della PCA

- Riepilogando, la PCA prevede le seguenti operazioni

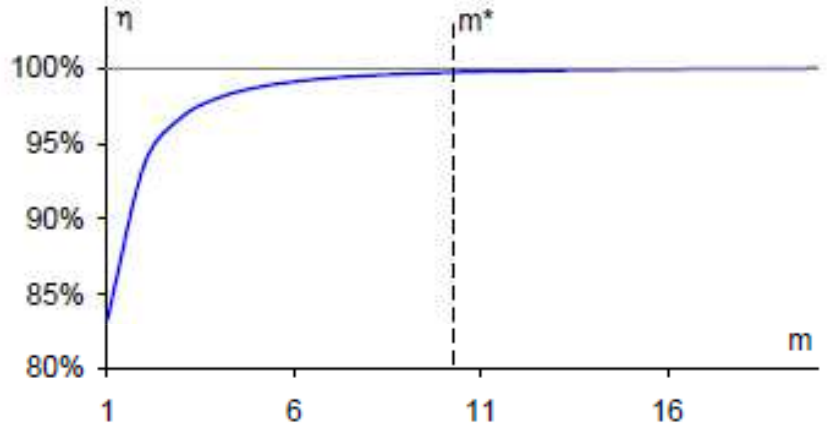
- Calcolo della media e della covarianza del campione
- Calcolo degli autovalori/autovettori della matrice di covarianza
- Ordinamento decrescente degli autovalori
- Costruzione di T mediante gli autovettori corrispondenti agli *m* autovalori maggiori
- Proiezione dei dati mediante T

- La trasformazione può essere interpretata come una rototraslazione nello spazio delle features

Osservazioni sulla PCA

- Si può valutare empiricamente la quantità di informazione persa mediante il calcolo del fattore di efficienza

$$\eta = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$



- La PCA assume quindi che i dati importanti siano quelli sui cui assi si osserva una varianza elevata

- Non si fa uso di alcuna informazione di classe, per cui la PCA è un metodo non supervisionato
- Non è detto quindi che le singole classi siano ortogonalizzate !

PCA su immagini

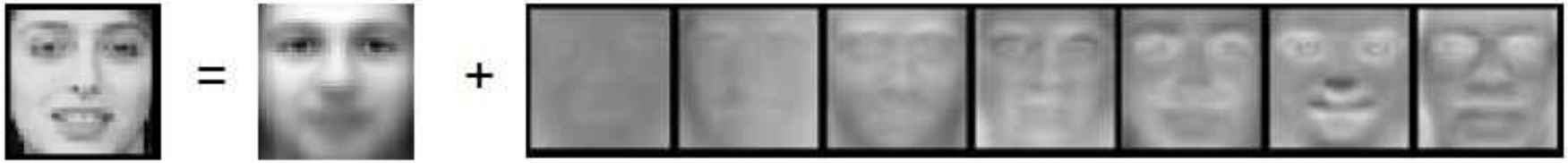
- Cosa accade quando le componenti del vettore sono i pixel di una immagine che rappresenta un oggetto (es. un volto)?
- Gli autovettori corrispondenti agli autovalori più grandi definiscono a loro volta immagini su cui l'originale può essere proiettata (eigen-objects)
- Ai pixel originari si può sostituire un vettore di (pochi) valori pari ai pesi della proiezione sul sottospazio sotteso dagli eigen-objects
- L'approccio si adatta a tutti i casi in cui le immagini condividono tutte la medesima dimensione e struttura morfologica

Eigen-faces

- Un volto X espresso come vettore di pixel

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$
$$= \quad w_1, \ldots, w_k$$

- La sua ricostruzione tramite eigen-faces

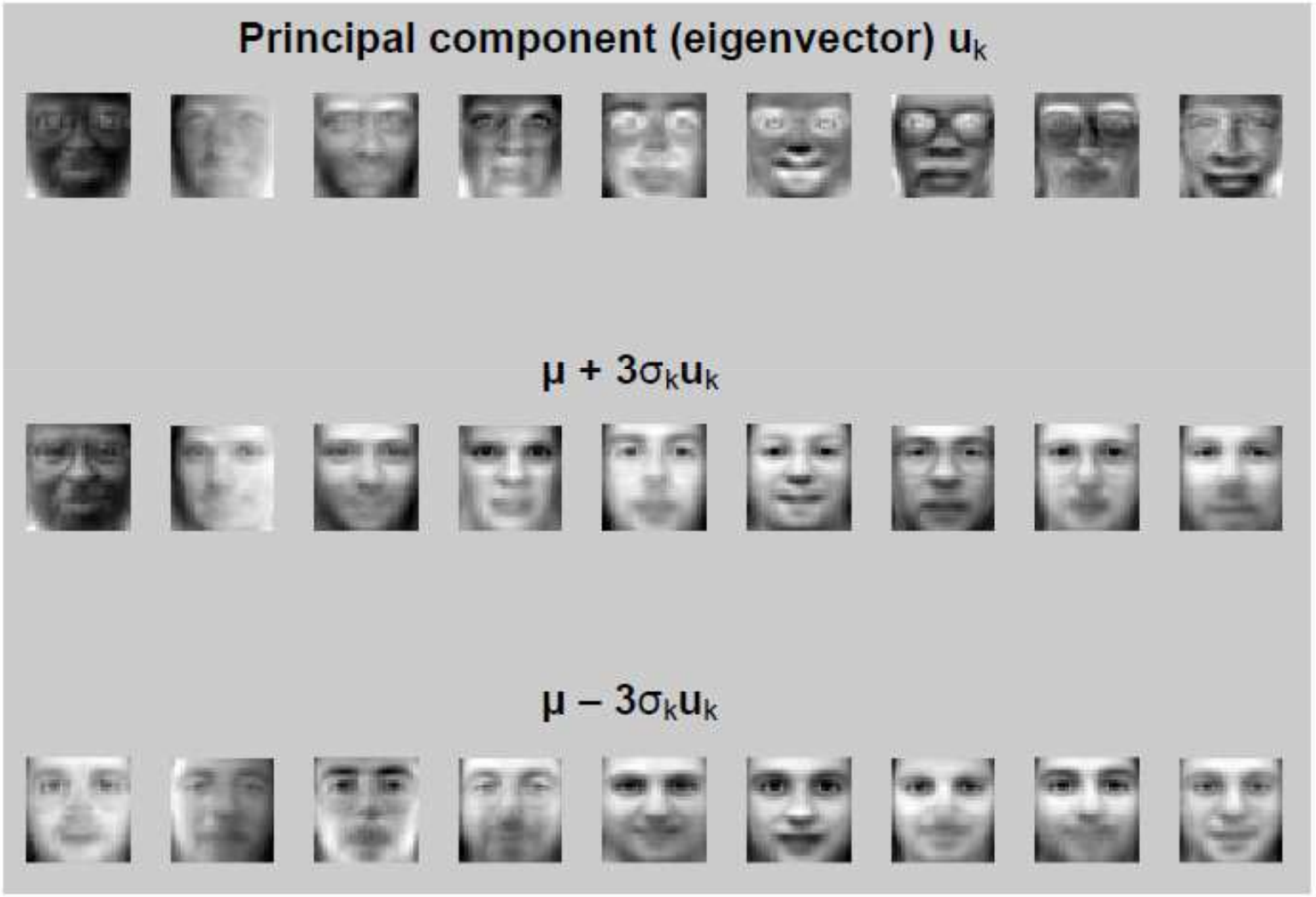


$$\hat{\mathbf{x}} = \boldsymbol{\mu} + w_1\mathbf{u}_1 + w_2\mathbf{u}_2 + w_3\mathbf{u}_3 + w_4\mathbf{u}_4 + \ldots$$

- I pesi $w_1 \ldots w_k$ codificano l'intera immagine

Eigen-faces

- Ogni autovettore codifica una qualche proprietà del volto



Independent Component Analysis

- Consiste nella ricerca di componenti statisticamente indipendenti. Questo può essere fatto in diversi modi:
    - minimizzando la mutua informazione
    - massimizzando la non-gaussianità

- La mutua informazione viene definita tramite la divergenza Kullback-Leibler (generalizza la definizione di entropia classica)

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x,$$

- La ricerca di non-gaussianità è giustificata dal teorema del limite centrale.
- Se i dati combinano sorgenti generiche, il loro mix tende ad essere più gaussiano. Ma allora una buona decomposizione deve massimizzare la non-gaussianità.

Projection pursuit per ICA

- La non-gaussianità di una distribuzione può essere valutata in base alla
  - negentropia (differenza di entropia rispetto ad una gaussiana di pari varianza, che ha entropia massima)
  - curtosi (momento del quarto ordine normalizzato)

$$K = \frac{\mathbf{E}[(\mathbf{y} - \overline{\mathbf{y}})^4]}{(\mathbf{E}[(\mathbf{y} - \overline{\mathbf{y}})^2])^2} - 3$$

- Il projection pursuit ricava iterativamente le componenti indipendenti secondo un dato indice di non-gaussiantità *K* in modo iterativo
  - si cerca una combinazione lineare w_0 che massimizza *K* (tramite discesa del gradiente)
  - nel sottospazio ortogonale a w_0 si cerca una combinazione w_1 che massimizza *K* (Gram-Schmidt)
  - si itera fino ad esaurimento

Analisi discriminante lineare

- L’analisi discriminante lineare (LDA) cerca di estrarre features migliori della PCA utilizzando anche le informazioni di classe (supervisione)

- Viene detta anche trasformata di Fischer, in quanto è basata su una generalizzazione dell’indice di separabilità tra classi proposto da Fisher

- L’idea di base è cercare proiezioni lineari che allo stesso tempo:
  - Minimizzano la varianza intra-classe
  - Massimizzano la dispersione inter-classe

Analisi discriminante lineare

- Per estendere l'indice di separazione di Fisher al caso con M classi si usa una misura della dispersione delle classi basata sulle distanze tra i vettori medi di ciascuna

$$\mathcal{J}(T) = \frac{\left| \sum_{i=1}^{M} N_i (\tilde{\mu}_i - \tilde{\mu})(\tilde{\mu}_i - \tilde{\mu})^t \right|}{\left| \sum_{i=1}^{M} \tilde{S}_i \right|}$$

$$\tilde{\mu} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{y}_k = T\mu$$

- Il numeratore è una media pesata della distanza tra la media di ogni classe e la media dell'intero campione dopo la trasformazione

Analisi discriminante lineare

- Esprimendo tutte le quantità in funzione dei dati e della matrice di trasformazione T si ottiene

$$\mathcal{J}(T) = \frac{\left|TS_bT^t\right|}{\left|TS_wT^t\right|} \qquad S_b = \sum_{i=1}^{M} N_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t \qquad S_w = \sum_{i=1}^{M} S_i$$

- La condizione di stazionarietà di J porta a concludere che gli assi migliori sono quelli per cui vale

$$(S_w^{-1}S_b - \lambda_i I)\mathbf{e}_i = \mathbf{0}$$

- Il risultato è simile a quello della PCA ma ora la scelta dipende anche dalla dispersione inter-classe e non solo dalla covarianza

Osservazioni sulla LDA

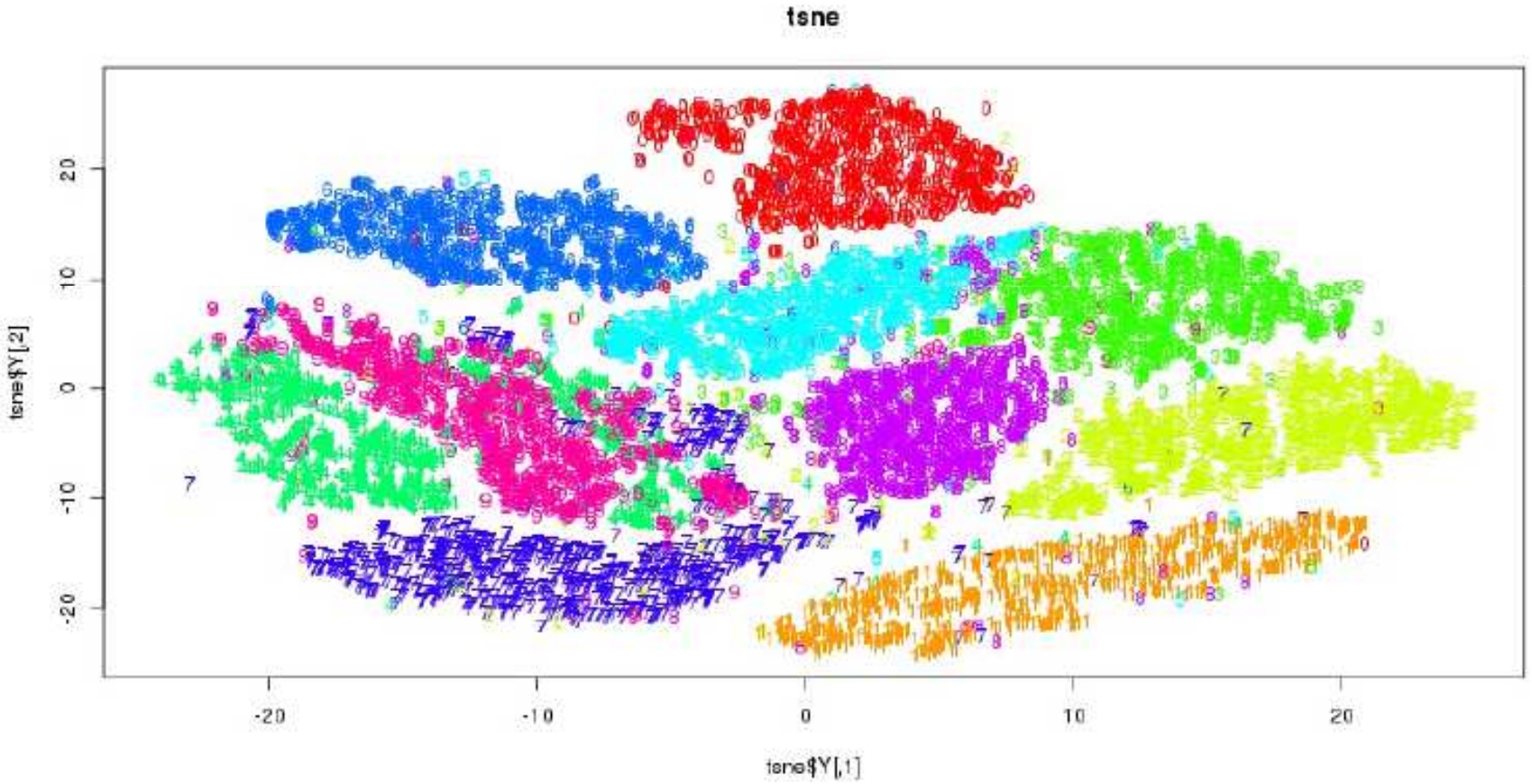
- La matrice di dispersione inter-classe ha al più rango *M-1* (somma di *M* matrici di rango *1* con *1* vincolo sulla media che introduce dipendenza)
- Il metodo è quindi in grado di fornire al massimo *M-1* features, per cui non è di uso generale come la PCA
- Nel caso *M=2* fornisce *1* sola feature, che quantifica la distanza dall'iperpiano ottimale di separazione tra le classi (secondo Fisher)

Tecniche non lineari

- Risultati migliori rispetto a PCA e similari si ottengono utilizzando tecniche non-lineari di estrazione features
- Un approccio neurale si base sull'uso di autoencoders
- Un metodo probabilistico, T-distributed Stochastic Neighbor Embedding (t-SNE), riesce a trovare ottime features minimizzando la divergenza Kullback-Leibler tra la distribuzione originale e quella nel sottospazio di arrivo.
- t-SNE preserva in modo ottimale la struttura locale dello spazio (punti vicini restano vicini)
- Viene usato anche per mappare visivamente spazi N-dimensionali su mappe a 2-3 dimensioni

Esempio con t-SNE

- t-SNE riesce a mappare efficacemente in 2D un problema complesso come MNIST (riconoscimento caratteri manoscritti)



Conclusioni

- I metodi di estrazione sono in generale superiori ai metodi di selezione (che ne sono un caso particolare), in quanto forniscono una maggiore flessibilità in termini pratici

- Tuttavia le features che essi producono non hanno in generale significato fisico e non semplificano il processo di misura (tutte le features originali vanno comunque calcolate)

- La scelta del metodo da adottare è quindi in generale frutto dei vincoli applicativi e dei vantaggi/svantaggi che esso produce nel caso particolare

- Gli approcci basati su LASSO o su algoritmi genetici consentono di produrre ottime scelte qualora la selezione sia necessaria (e non banale)