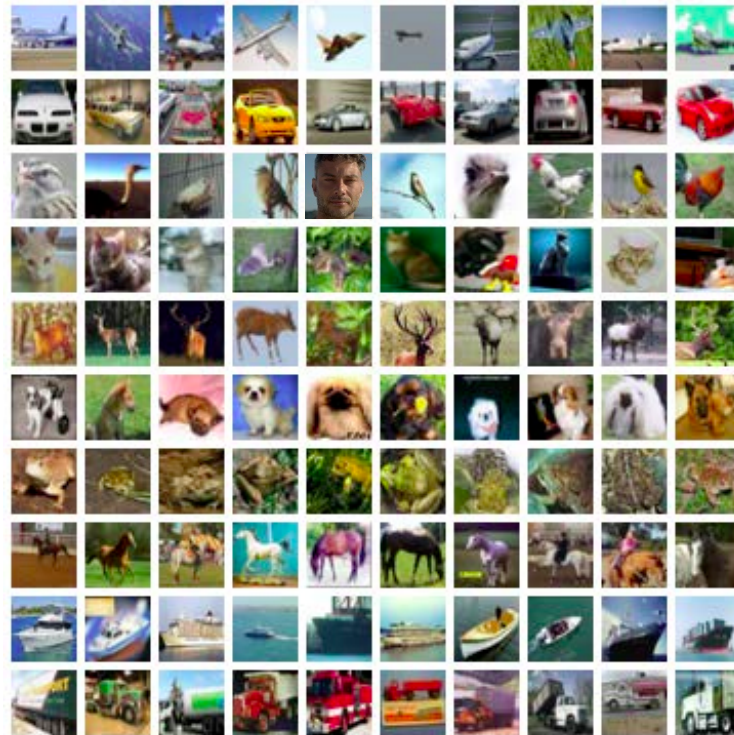


Sistemi avanzati per il Riconoscimento



Riconoscimento di oggetti e scene – metodi generativi

Dr. Marco Cristani



Modelli generativi - recap

- Un modello generativo studia una classe per volta, indipendentemente (*class specific*)
- C classi \rightarrow C modelli generativi *della stessa famiglia*, per esempio
 - Gaussiane
 - misture di Gaussiane,
 - Hidden Markov models...



Modelli generativi – recap (2)

- In fase di classificazione, preso un elemento di test x , scelgo il modello c_{best} (e quindi la classe c_{best}) che mi restituisce la likelihood $P(x/ c)$ massima
- Se ho anche una conoscenza a priori $P(c)$, considero la massima probabilità a posteriori $P(c/x) = P(x/ c) P(c)$ (*Regola di decisione di Bayes*)



Modelli generativi - analisi

- Ma perché il termine *generativo*?
- I modelli generativi studiano le osservazioni, o *dati visibili* v , assumendo essi siano stati generati da *eventi latenti* h
- Tali eventi latenti h sono:
 - accaduti prima di avere le osservazioni (es. sequenza genetica)
 - inaccessibili con i sensori a disposizione (li posso solo inferire, es. l'esistenza di una stella)



Modelli generativi – analisi (2)

- Un modello generativo è di solito parametrico, con parametri (anch'essi latenti) h^θ
- Nella progettazione di un modello generativo, si definisce innanzitutto la *distribuzione congiunta* $P(v, h/h^\theta)$



Modelli generativi – analisi (3)

- Fattorizzando tale distribuzione congiunta *o completa* $P(v, h/h^\theta)$, ossia generando delle distribuzioni condizionali, si definisce formalmente il *processo generativo* di un modello
- Inoltre, la fattorizzazione ci permette di disegnare il *modello grafico* del modello generativo



BoW e i metodi generativi

- Supponiamo di voler classificare in modo generativo una bag of words (BoW)
- Supponiamo di avere una collezione di documenti (*visibili*)

$$\mathcal{D} = \{d_1, \dots, d_N\}$$

formati da parole (anch'esse *visibili*)
provenienti da un vocabolario

$$\mathcal{W} = \{w_1, \dots, w_M\}$$



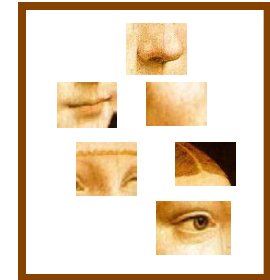
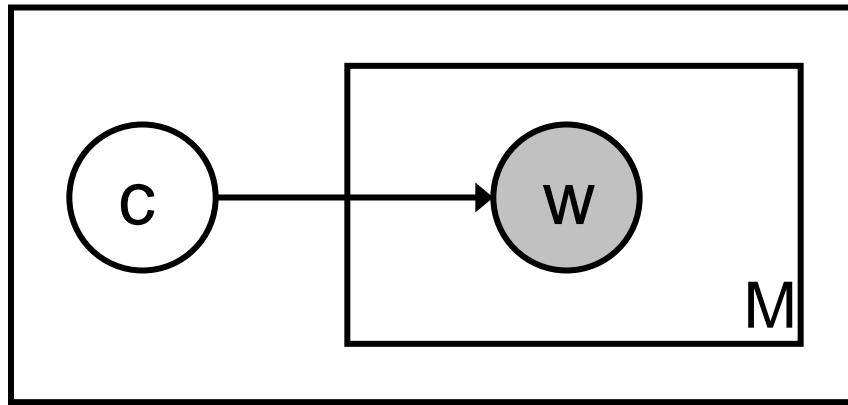
BoW e i metodi generativi (2)

- Supponiamo che ogni documento provenga da una particolare classe c_i presa dall'insieme delle classi H

$$H = \{c_1, \dots, c_C\}$$



Processo generativo

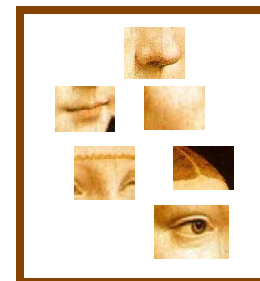
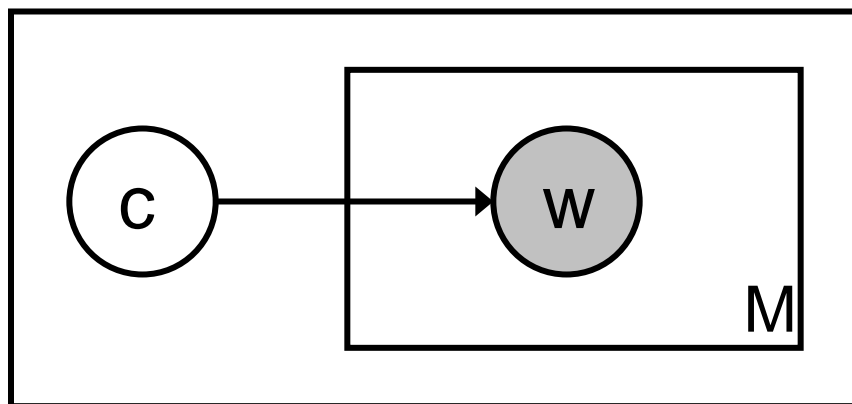


$$P(c, w) = P(w | c)P(c) \propto P(c | w)$$

1. Preso un dato documento ϵD
2. scelgo una particolare classe tra le C a disposizione, usando la distribuzione $P(c)$
3. data la classe, estraggo M variabili visibili usando la distribuzione $P(w/c)$



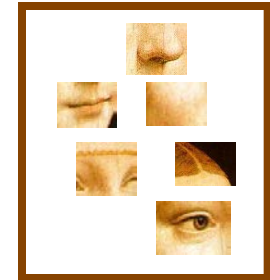
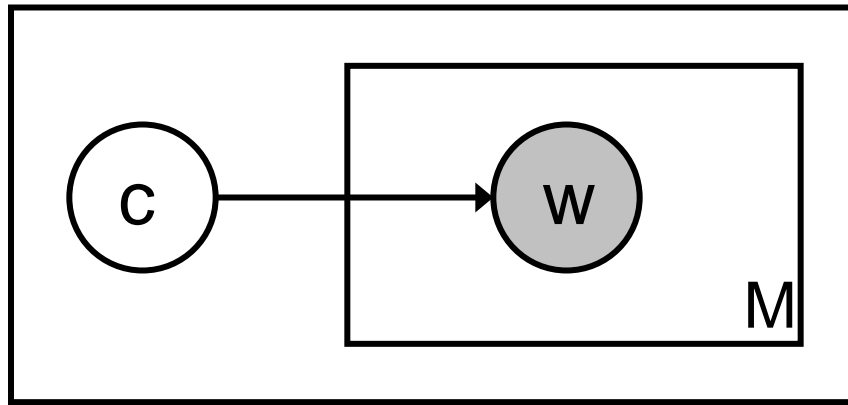
Plate notation di un modello generativo



- In *grigio* le variabili visibili, in *bianco* le nascoste
- Il rettangolo (*plate*) con un numero nell'angolo indica il num. di variabili di quel tipo nel modello
- Una distribuzione condizionale individua una freccia nel modello



Classificazione con il modello generativo



$$i_{best} = \arg \max_i p(c_i | w) \propto p(c_i) p(w | c_i) = p(c_i) \prod_{j=1}^M p(w_j | c_i)$$

Decisione di
classe

Prior per
la classe

Likelihood dell'immagine
data la classe

- c è una variabile nascosta



Probabilistic Latent Semantic Analysis (PLSA)

- Tecnica nata nell'ambito dell'analisi dei testi
- Paper originale:
Unsupervised Learning by Probabilistic Latent Semantic Analysis
– Thomas Hoffman, 1999



<http://cs.brown.edu/~th/>



PLSA - introduzione

- Ideata per risolvere i problemi di
 - **Polisemia**: una parola può avere più significati: quando ne vedo un'occorrenza, che significato vi associo?
 - Esempio: *lama* (del coltello, oppure l'animale)
 - **Sinonimia**: più parole possono avere lo stesso significato: quando vedo due parole, come faccio a capire se sono sinonimi?
 - Esempio: *parlare, colloquiare*



LSA \rightarrow PLSA

- PLSA è vista come l'estensione probabilistica della LSA, Latent Semantic Analysis
- LSA è uno strumento di estrazione delle feature che riduce la dimensionalità
 - Proiezione in uno spazio a più bassa dimensionalità, chiamato anche *spazio latente*
 - Il problema di LSA è che non è giustificato formalmente



PLSA

- PLSA associa ad ogni parola una *variabile latente di contesto*, in grado di tenere in considerazione la polisemia
- Supponiamo di avere una collezione di documenti

$$\mathcal{D} = \{d_1, \dots, d_N\}$$

formati da parole provenienti da un
vocabolario $\mathcal{W} = \{w_1, \dots, w_M\}$



PLSA (2)

- Ignorando l'ordine con cui le parole occorrono in un documento, si possono organizzare tutti i dati in una *matrice di count NxM* (o *co-occurrence table*)

$$\mathbf{N} = (n(d_i, w_j))_{ij}$$

dove $n(d_i, w_j)$ denota il numero di volte che la parola w_j occorre nel documento d_i



PLSA – matrice di count

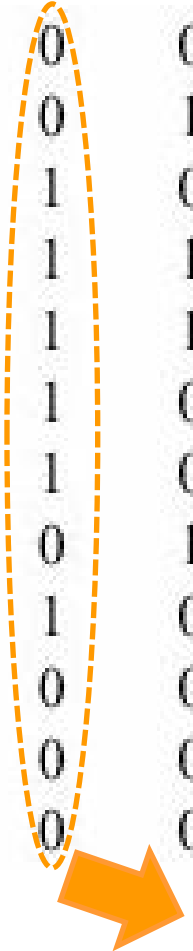
		i N documenti									
j parole M	human	1	0	0	1	0	0	0	0	0	0
	interface	1	0	1	0	0	0	0	0	0	0
	computer	1	1	0	0	0	0	0	0	0	0
	user	0	1	1	0	1	0	0	0	0	0
	system	0	1	1	2	0	0	0	0	0	0
	response	0	1	0	0	1	0	0	0	0	0
	time	0	1	0	0	1	0	0	0	0	0
	EPS	0	0	1	1	0	0	0	0	0	0
	survey	0	1	0	0	0	0	0	0	0	1
	trees	0	0	0	0	0	1	1	1	1	0
	graph	0	0	0	0	0	0	1	1	1	1
	minors	0	0	0	0	0	0	0	1	1	1

matrice di count N



PLSA – BoW nella count matrix

		i N documenti									
j parole M		human	1	0	0	1	0	0	0	0	0
		interface	1	0	1	0	0	0	0	0	0
		computer	1	1	0	0	0	0	0	0	0
		user	0	1	1	0	1	0	0	0	0
		system	0	1	1	2	0	0	0	0	0
		response	0	1	0	0	1	0	0	0	0
		time	0	1	0	0	1	0	0	0	0
		EPS	0	0	1	1	0	0	0	0	0
		survey	0	1	0	0	0	0	0	0	1
		trees	0	0	0	0	0	1	1	1	0
		graph	0	0	0	0	0	0	1	1	1
		minors	0	0	0	0	0	0	0	1	1

 **Bag of words per il documento i !**



Zero-frequency problem

- La matrice **N** è *sparsa*
 - matrice in cui il numero di entry diverse da 0 è molto basso ($\sim 1\%$)
- Il fenomeno della sparsatezza può essere un problema
 - Nel caso di metriche di similarità tra documenti, in cui si contano i termini comuni
 - Poichè le parole sono moltissime, nel caso di sinonimi vado a perdere un match valido



Zero-frequency problem - example

A		B		C	
<i>human</i>	1	<i>human</i>	0	<i>human</i>	0
<i>interface</i>	0	<i>interface</i>	0	<i>interface</i>	0
<i>computer</i>	0	<i>computer</i>	0	<i>computer</i>	0
<i>user</i>	0	<i>user</i>	1	<i>user</i>	0
<i>system</i>	0	<i>system</i>	0	<i>system</i>	0
<i>response</i>	0	<i>response</i>	0	<i>response</i>	0
<i>time</i>	0	<i>time</i>	0	<i>time</i>	0
<i>EPS</i>	0	<i>EPS</i>	0	<i>EPS</i>	0
<i>survey</i>	0	<i>survey</i>	0	<i>survey</i>	0
<i>trees</i>	0	<i>trees</i>	0	<i>trees</i>	0
<i>graph</i>	0	<i>graph</i>	0	<i>graph</i>	1
<i>minors</i>	0	<i>minors</i>	0	<i>minors</i>	0

Quali sono i due documenti tra loro più simili?



Zero-frequency problem - osservazioni

- Nel precedente esempio, costruendo criteri di similarità basati su prodotti interni risultava poco efficace (zero in tutti i casi)
- Tenendo conto però dei sinonimi (human e user) avrei potuto dire che tra di loro **A** e **B** sono più simili



PLSA - Notazione

- PLSA è un modello *latente*, ossia ha delle variabili nascoste, ossia non visibili direttamente dalle osservazioni (parole e documenti)
- Ad ogni osservazione (documento) d_i associo una variabile latente

$$z_k \in \{z_1, \dots, z_K\}$$



PLSA – Notazione (2)

- Introduciamo le seguenti probabilità:

$P(d_i)$ avere il particolare documento d_i

$P(w_j, d_i)$ avere la particolare parola w_j
e il documento d_i

$P(w_j \mid z_k)$ avere la particolare parola w_j
conoscendo z_k

una distribuzione specifica per il
 $P(z_k \mid d_i)$ documento d_i di avere l'istanza
della variabile z_k



Processo generativo di PLSA

- PLSA è un modello generativo
- Come tale, sottintende un processo generativo di produzione dei dati (come i dati sono stati prodotti)
- Tale processo generativo è descritto da distribuzioni di probabilità
- Le distribuzioni le abbiamo specificate nella slide precedente



Processo generativo di PLSA (2)

1. Seleziona un documento d_i con probabilità $P(d_i)$
2. Seleziona una classe latente z_k con probabilità

$$P(z_k | d_i)$$

3. Genera una parola w_j con probabilità


$$P(w_j | z_k)$$



Fattorizzazione di PLSA

1. La probabilità congiunta di un modello generativo (necessaria quando si deve trattare un modello generativo) è

$$P(d_i, w_j) = P(d_i)P(w_j | d_i)$$


$$P(w_j | d_i) = \sum_{k=1}^K P(w_j | z_k)P(z_k | d_i)$$




Fattorizzazione di PLSA (2)

- In pratica riscrivo $P(w_j/d_i)$ come una combinazione convessa di *aspetti*

$$P(d_i, w_j) = P(d_i) P(w_j | d_i)$$

$$P(w_j | d_i) = \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i)$$




PLSA – leggere tra le righe...

- Analizziamo a cosa servono i vari componenti di PLSA

$$P(d_i)$$

Probabilità a priori sui documenti. Di solito

$$P(w_j | z_k)$$

uniforme (tutti i documenti che analizzo sono

$$P(z_k | d_i)$$

ugualmente probabili, non ci sono documenti «rari»)



PLSA – leggere tra le righe... (2)

- Analizziamo a cosa servono i vari componenti di PLSA

- $P(w_j | z_k)$
- modella l'accorpamento di termini simili o co-occorrenti (risolve il problema dei sinonimi)
 - z_k viene chiamato anche **topic**, da cui topic model



PLSA – leggere tra le righe... (3)

$$P(w_j | z_k)$$

- in pratica, un topic raccoglie parole che fanno riferimento ad un contesto semantico ben definito, distinguibile da altri
- Per capire cosa c'è in un topic, fisso z_k e valuto i w_j per cui $P(w_j | z_k)$ è alta



PLSA – leggere tra le righe... (4)

$$P(w_j | z_k)$$

- Fissando una soglia su $P(w_j | z_k)$, assegno differenti parole a differenti topic
- ATTENZIONE: *la stessa parola può essere assegnata a differenti topic* (risolvo il problema della polisemia)



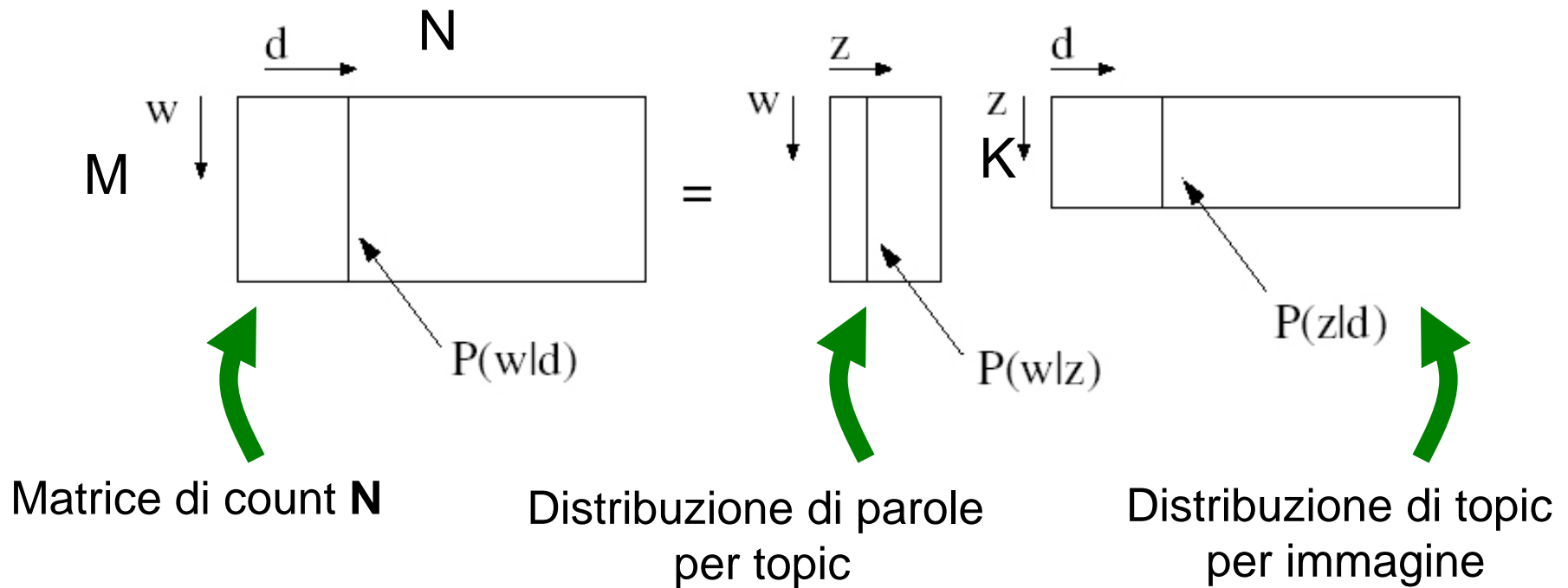
PLSA – leggere tra le righe... (5)

- $P(z_k | d_i)$
- per capire in un documento quali topic ci sono
 - rappresentazione a più bassa dimensionalità del documento, in sostituzione del BoW
 - Posso usare questo per fare classificazione... *see later*



PLSA, graficamente

$$p(w_i | d_j) = \sum_{k=1}^K p(w_i | z_k) p(z_k | d_j)$$



$$K \ll \min(N, M)$$



Zero-frequency problem - solution

	A		B		C	
z_1	human	1	human	0	human	0
	interface	0	interface	0	interface	0
	computer	0	computer	0	computer	0
	user	0	user	1	user	0
z_2	system	0	system	0	system	0
	response	0	response	0	response	0
	time	0	time	0	time	0
z_3	EPS	0	EPS	0	EPS	0
	survey	0	survey	0	survey	0
z_4	trees	0	trees	0	trees	0
	graph	0	graph	0	graph	1
	minors	0	minors	0	minors	0

Quali sono i due documenti tra loro più simili?



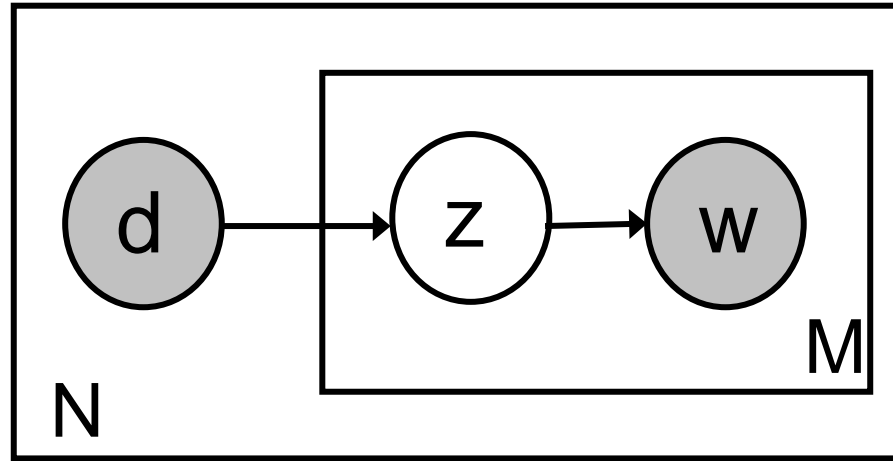
Zero-frequency problem – solution (2)

	A	$P(z A)$	B	$P(z B)$	C	$P(z C)$
z_1	human	1	human	0	human	0
	interface	0	interface	0	interface	0
	computer	0	computer	0	computer	0
	user	0	user	1	user	0
z_2	system	0	system	0	system	0
	response	0	response	0	response	0
	time	0	time	0	time	0
z_3	EPS	0	EPS	0	EPS	0
	survey	0	survey	0	survey	0
z_4	trees	0	trees	0	trees	0
	graph	0	graph	0	graph	1
	minors	0	minors	0	minors	0

Se eseguo il prodotto scalare tra i vari $P(z|d)$, trovo la soluzione!



PLSA rappresentazione grafica



- Le variabili visibili $\{v\}$ sono w, d
- Le variabili latenti (nascoste) $\{h\}$ sono z
- I parametri (nascosti) $\{h^\theta\}$ sono i valori assunti dalle varie distribuzioni in gioco



Training di PLSA - Likelihood

- Capiamo come fare il training della PLSA
- Iniziamo con il calcolo della likelihood *dei dati visibili* (perché applicheremo il learning max. lik., che massimizza la lik)

$$\begin{aligned}\mathcal{L} &= \prod_{i=1}^N \prod_{j=1}^M P(d_i, w_j)^{n(d_i, w_j)} \\ &\propto \sum_{i=1}^N \sum_{j=1}^M n(d_i, w_j) \log P(d_i, w_j)\end{aligned}$$



Training di PLSA – Likelihood (2)

- Dove

$$\sum_{j=1}^M n(d_i, w_j) = n(d_i)$$

modella la lunghezza (in termini di parole)
del documento

- Per fare il training, devo stimare i
parametri per cui

$$\frac{\partial \mathcal{L}}{\partial h^\theta} = 0$$



Likelihood - Fattorizzazione

$$\begin{aligned}
 \mathcal{L} &= \sum_{i=1}^N \sum_{j=1}^M n(d_i, w_j) \log P(d_i, w_j) \\
 &= \sum_{i=1}^N \sum_{j=1}^M n(d_i, w_j) \log (P(d_i) P(w_j | d_i)) \\
 &= \sum_{i=1}^N \sum_{j=1}^M [n(d_i, w_j) \log P(d_i) + n(d_i, w_j) \log P(w_j | d_i)] \\
 &= \sum_{i=1}^N n(d_i) \left[\log P(d_i) + \sum_{j=1}^M \frac{n(d_i, w_j)}{n(d_i)} \log \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i) \right]
 \end{aligned}$$

Log di una somma.



Learning della PLSA – EM recap

$$\log P(v | h^\theta) = \log P(v)$$

$$= \log \left(\int_h P(h, v) \right)$$

$$= \log \left(\int_h Q(h) \frac{P(h, v)}{Q(h)} \right)$$

$$\geq \int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v))$$



Learning della PLSA – EM recap (2)

$$\int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v)) \leq \log(P(v))$$

- h = *variabile nascosta*
 - una caratteristica nascosta, latente, del singolo dato visibile
- $P(h, v)$ = *complete data (hidden + visible) likelihood*
 - Spiega come i dati visibili e nascosti sono interconnessi tra loro
- $Q(h)$ = *distribuzione di supporto per le variabili nascoste*
 - una distribuzione sulle variabili nascoste, più semplice della complete data likelihood



Learning della PLSA – EM recap (3)

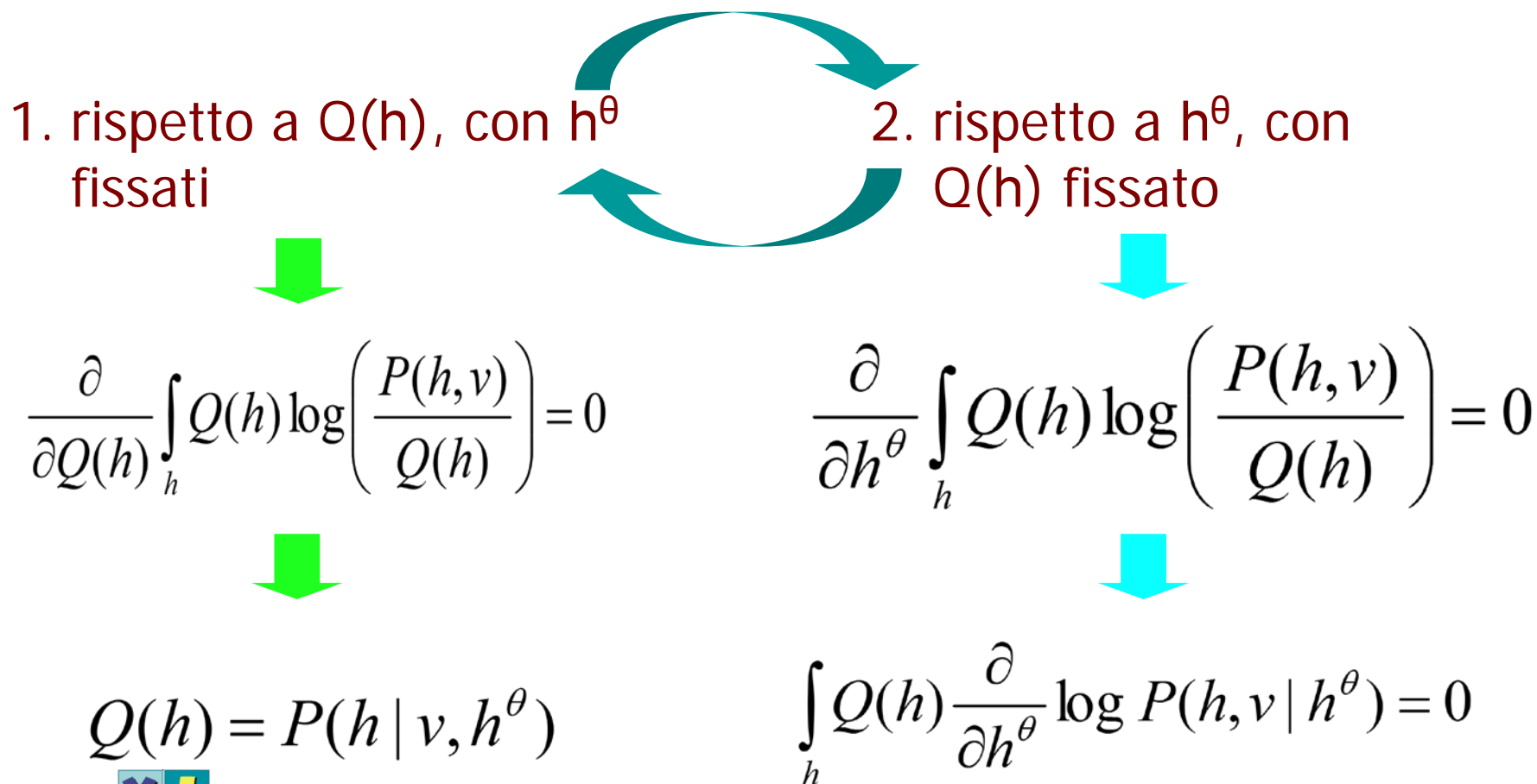
$$\int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v)) \leq \log(P(v))$$

- $F(Q(h), P(h, v))$
 - Una divergenza tra Q, P , un funzionale
 - una approssimazione per difetto della likelihood (cattura peggio quanto bene i dati fittano con il modello)
 - un oggetto con $Q(h)$ sconosciuta
 - un oggetto con h^θ sconosciuta



Learning della PLSA – EM recap (4)

- Minimizzo $F(Q,P)$ *in modo alternato*



PLSA – Expectation-Maximization

- E-step
 - Equivale a trovare una forma funzionale in forma tabellare della funzione di supporto

$$\begin{aligned} Q(h) &= P(h | v, h^\theta) \\ &= P(z_k | d_i, w_j) \end{aligned}$$



EM – Ricavo Q(h)

$$\begin{aligned} P(z_k | d_i, w_j) &= \frac{P(z_k, d_i, w_j)}{P(d_i, w_j)} \\ &= \frac{P(w_j | z_k, d_i) P(z_k | d_i)}{P(d_i, w_j)} \\ &= \frac{P(w_j | z_k) P(z_k | d_i)}{\sum_{l=1}^K P(w_j | z_l) P(z_l | d_i)} \end{aligned}$$



EM – remark importante

- M-step $\int_h Q(h) \frac{\partial}{\partial h^\theta} \log P(h, v | h^\theta) = 0$

NOTA BENE LA DIFFERENZA

$$P(v) = \sum_h P(h, v) \neq P(h, v)$$

$$\log P(v) = \sum_i \sum_j \log P(d_i, w_j)^{n(d_i, w_j)} =$$

$$\sum_i \sum_j n(d_i, w_j) \log \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i)$$

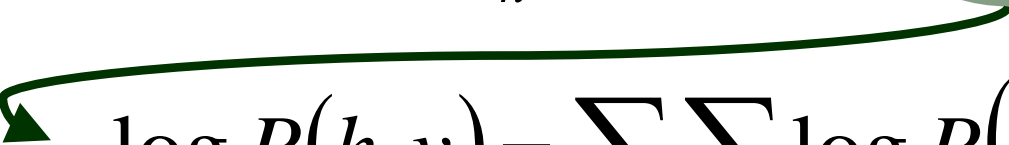


EM – remark importante (2)

- M-step $\int_h Q(h) \frac{\partial}{\partial h^\theta} \log P(h, v | h^\theta) = 0$

NOTA BENE LA DIFFERENZA

$$P(v) = \sum_h P(h, v) \neq P(h, v)$$


$$\begin{aligned} \log P(h, v) &= \sum_i \sum_j \log P(d_i, w_j, z_k)^{n(d_i, w_j)} = \\ &= \sum_i \sum_j n(d_i, w_j) \log [P(w_j | z_k) P(z_k | d_i)] \end{aligned}$$



EM – remark importante (3)

- Nel secondo caso ($P(h,v)$) *assumo di aver selezionato una particolare variabile nascosta!*
- Sparisce il log della somma



EM – Analisi dell'attesa su h

$$\int_h Q(h) \log P(h, v | h^\theta) = 0$$

$$\sum_k \sum_i \sum_j n(d_i, w_j) P(z_k | w_j, z_k) \cdot \\ \cdot \log [P(w_j | z_k) P(z_k | d_i)] = E[\mathcal{L}^c]$$



- A questo punto, i parametri nascosti sono le tavole di probabilità delle distribuzioni

$$P(w_j | z_k), P(z_k | d_i)$$

- Poichè i loro valori hanno dei vincoli, ossia

$$\sum_j P(w_j | z_k) = 1, \sum_k P(z_k | d_i) = 1$$

aggiungo i moltiplicatori di Lagrange e ottengo la formula finale da minimizzare rispetto ai parametri



EM – formulazione finale dell'M-step

$$\mathcal{H} = E[\mathcal{L}^c] + \sum_k \tau_k \left(1 - \sum_j P(w_j | z_k) \right) + \\ + \sum_i \rho_i \left(1 - \sum_k P(z_k | d_i) \right)$$

- Risolvendo il sistema ottengo un set di MK + NK equazioni, che mi permettono di ricavare le formule di ri-stima



Formule di ristima

- M-step

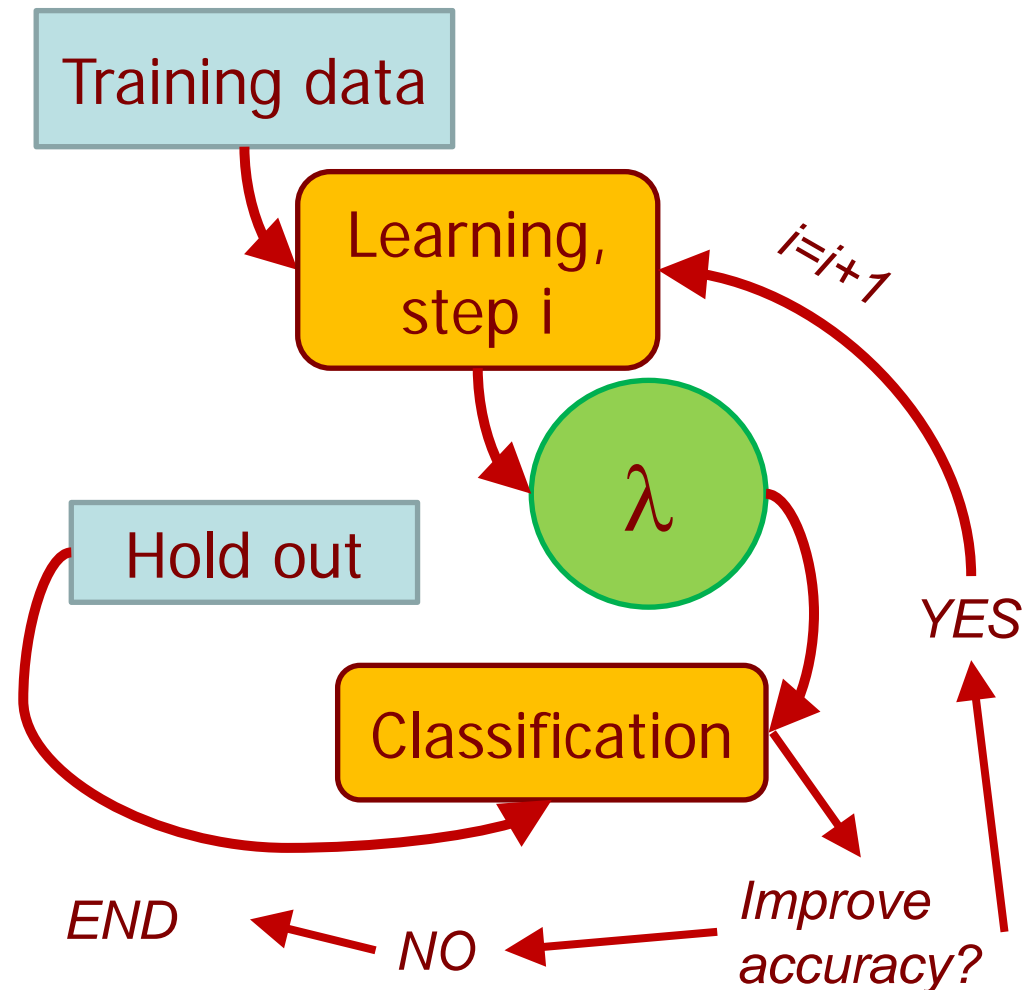
$$P(w_j | z_k) = \frac{\sum_{i=1}^N n(d_i, w_j) P(z_k | d_i, w_j)}{\sum_{m=1}^M \sum_{i=1}^N n(d_i, w_m) P(z_k | d_i, w_m)}$$

$$P(z_k | d_i) = \frac{\sum_{j=1}^M n(d_i, w_j) P(z_k | d_i, w_j)}{n(d_i)}$$



EM – come lo applico

- L'E-step e l'M step sono ripetuti fino a soddisfare una condizione di terminazione
- In alternativa, si può effettuare una procedura chiamata early stop, che in pratica lavora su cross-validazione, ed elimina l'overfitting



OK, so come fare learning con PLSA. E adesso?

- Diverse modalità di azione:

1. Learning trasduttivo + Object/ Scene classification

- a. si usa PLSA come un riduttore di features, si prendono tutti i dati, training e learning, e si addestra il modello
- b. Si usa $P(z/d)$ come rappresentazione a bassa dimensionalità (si passa da M a K dimensioni)
- c. Si applica un qualsiasi metodo di classificazione



OK, so come fare learning con PLSA. E adesso? (2)

2. Learning con trick + Object/ Scene classification

- a. si traina PLSA *solo* sui dati di training, ottenendo $P(w/z)$ e $P(z/d)$
- b. Si addestra(no) il(i) classificatore(i) sui dati di training
- c. Quando arriva un elemento di test, si stima $P(z/d)$ come rappresentazione a bassa dimensionalità , *mantenendo inalterato* $P(w/z)$



OK, so come fare learning con PLSA. E adesso? (3)

- d. In pratica, faccio l'EM, ma non applico la formula di ri-stima per $P(w/z)$, aggiornando esclusivamente $P(z/d)$
- e. Stimato $P(z/d)$ lo si dà in pasto al classificatore, che mi darà come output l'appartenenza ad una classe



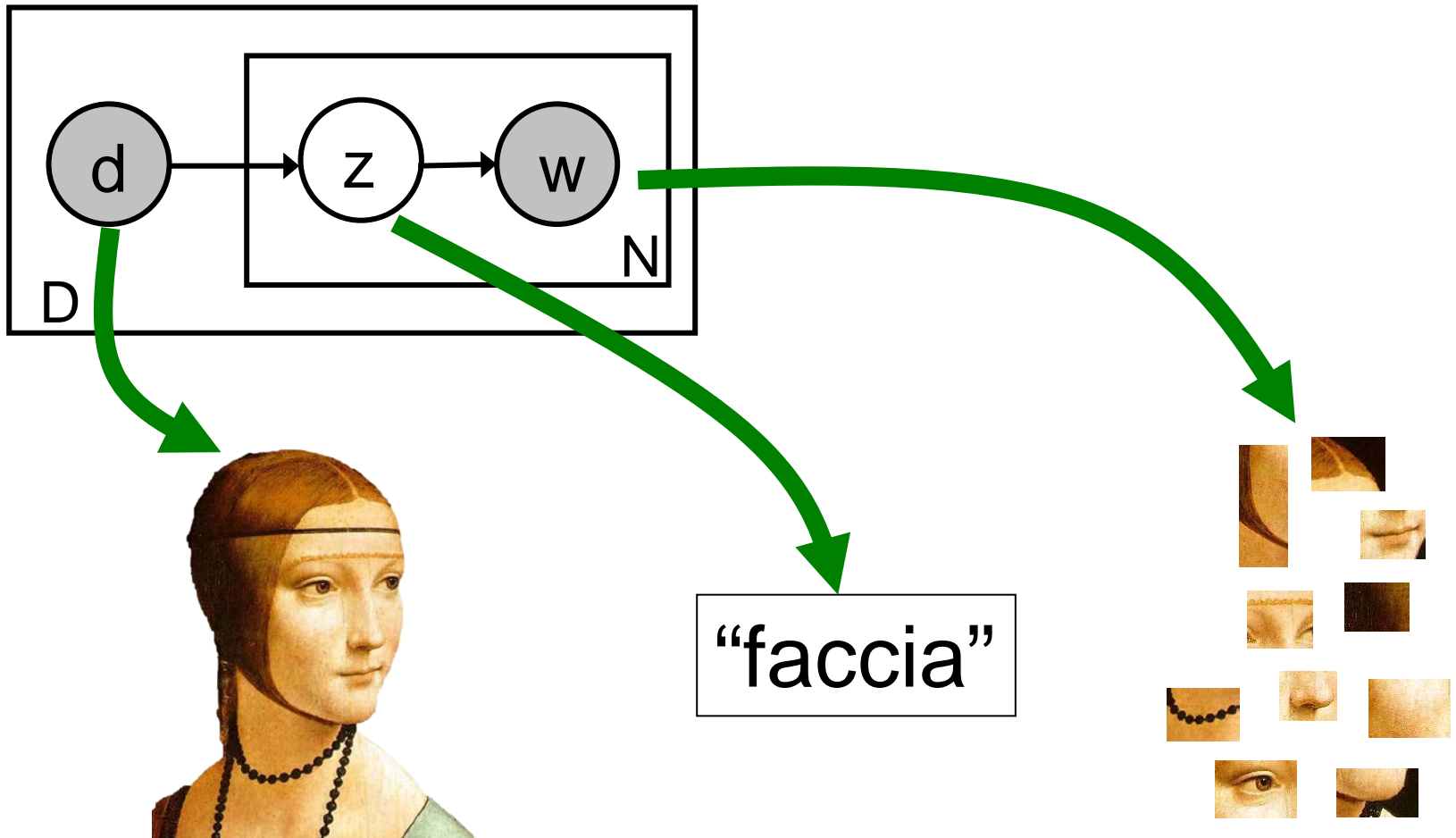
OK, so come fare learning con PLSA. E adesso? (4)

- d. **Topic analysis – trovo il topic maggiormente rappresentativo in una immagine** (una sorta di topic classification!)

$$z^* = \arg \max_z p(z | d)$$



OK, ma le immagini?

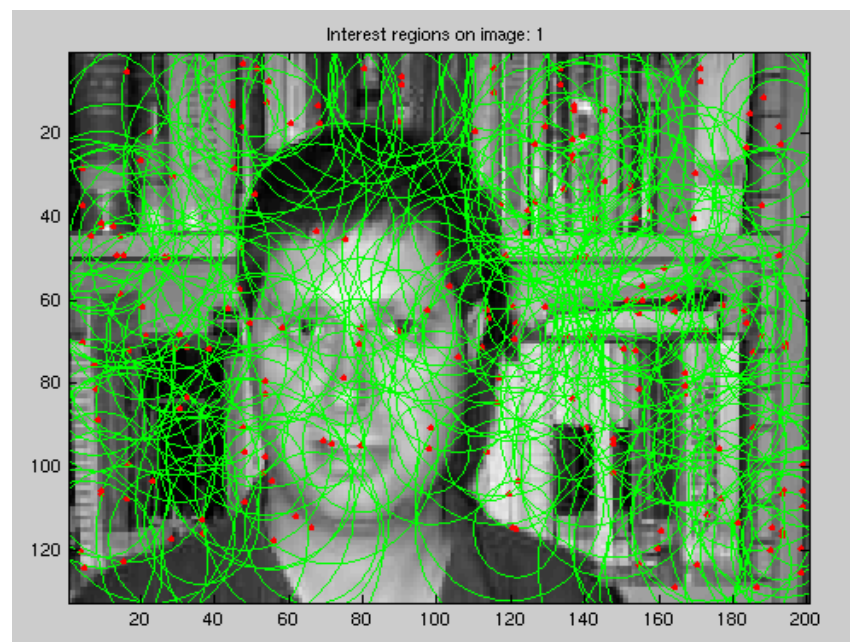
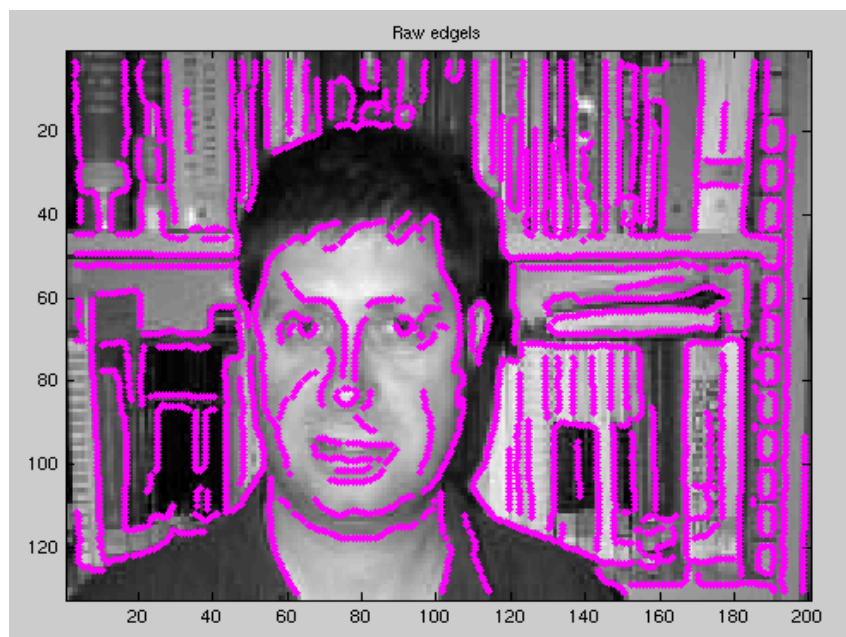


Topic Analysis (= face detection...)



Topic Analysis (= face detection...) (2)

- Raccogli le word (edgelet, SIFT)

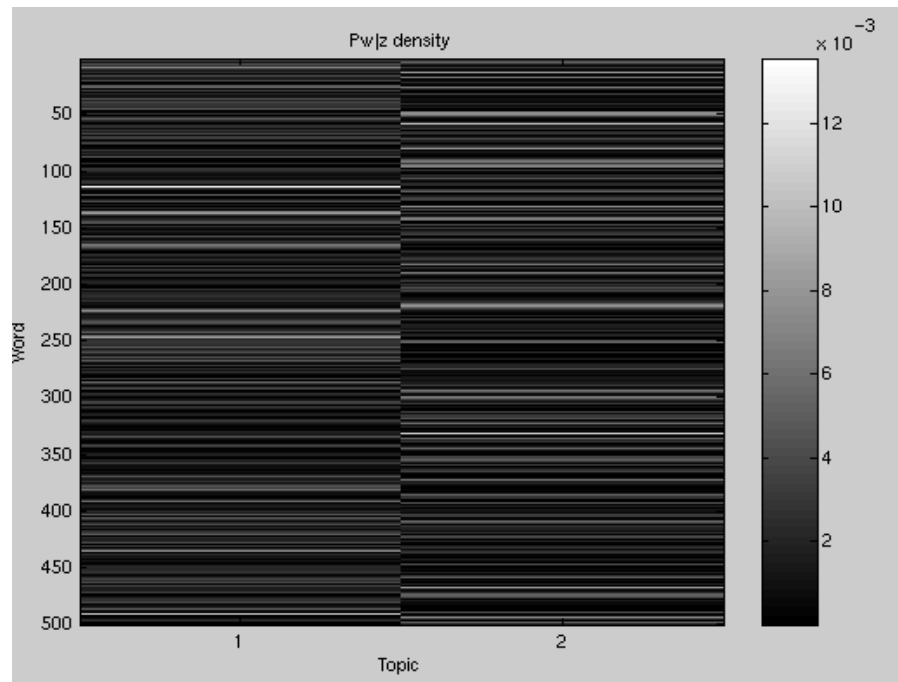


Topic Analysis (= face detection...) (2)

- Applica l'EM (2 topic)

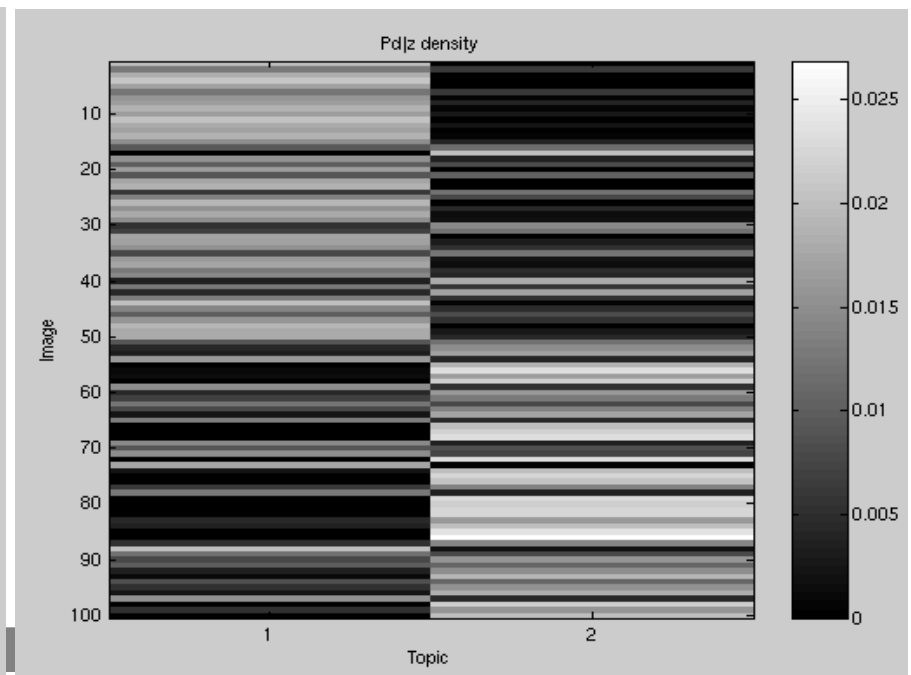
Distribuzione di parole per topic

$$p(w | z)$$



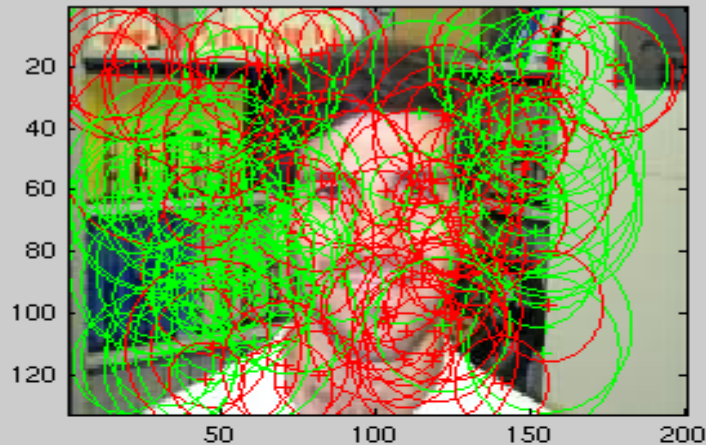
Distribuzione di topic per immagine

$$p(z | d)$$

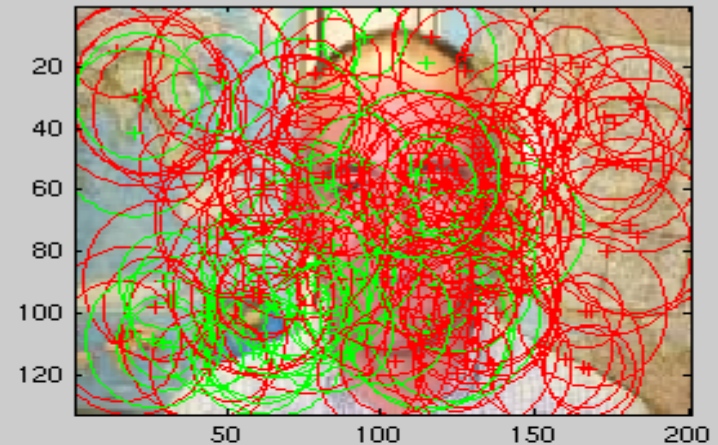


Topic Analysis (= face detection...) (3)

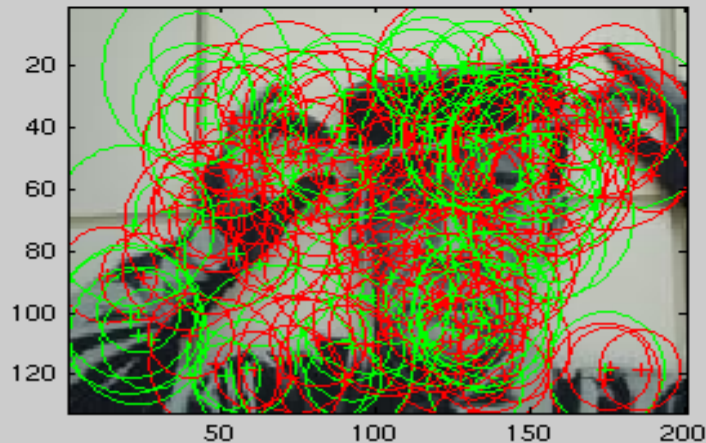
Correct - Image: 1 $P(z|d)=0.3662$ 0.6338



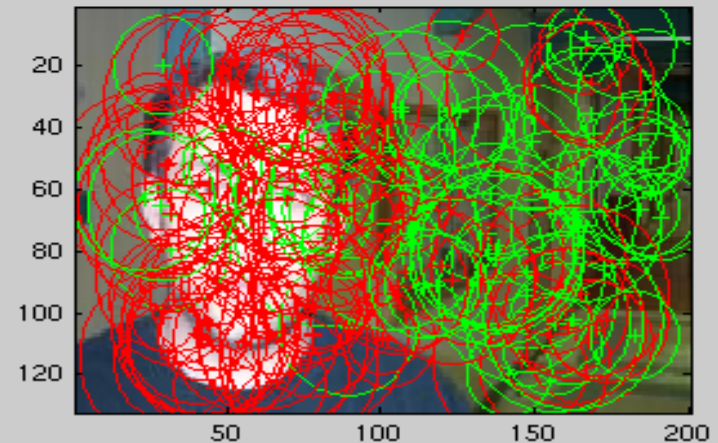
Correct - Image: 2 $P(z|d)=0.83087$ 0.16913



Correct - Image: 3 $P(z|d)=0.59906$ 0.40094



Correct - Image: 5 $P(z|d)=0.68534$ 0.31466



Materiale aggiuntivo

- Sui modelli grafici, metodi generativi (difficile ma completo)
 - [genmod.pdf](#)
- Su PLSA
 - [plsa.pdf](#)
- Su applicazioni di PLSA a object e scene classification
 - [Sivic05b.pdf](#)
 - [Scene Classification via pLSA.pdf](#)

