

# IN550 Machine Learning

## Altri modelli di apprendimento

Vincenzo Bonifaci

# Stima di densità: Kernel density estimation

- Modello generativo
- Scopo: ricostruire la distribuzione di probabilità dei dati

Definisce

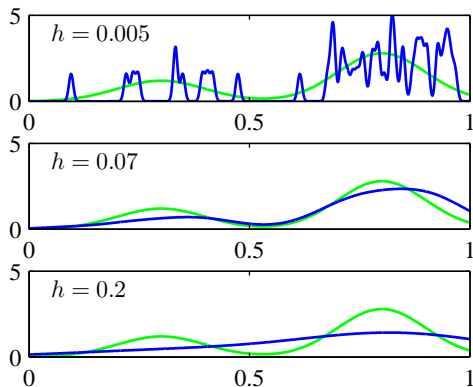
$$p(x) \propto \frac{1}{m} \sum_{i=1}^m k\left(\frac{x - x^{(i)}}{h}\right),$$

dove  $h > 0$  e la funzione  $k : \mathcal{X} \rightarrow \mathbb{R}$  (detta *Parzen kernel*) deve soddisfare

$$k(u) \geq 0, \int k(u) du = 1.$$

# Stima di densità: Kernel density estimation

Esempio: kernel gaussiano  $k(u) = \exp(-\|u\|^2/2)$



Rif.: C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006

# Individuazione di anomalie [anomaly detection]

- Applicazione: riconoscere osservazioni “anomale” (esempio: difetti di fabbrica, comportamenti sospetti, attacchi informatici ecc.)



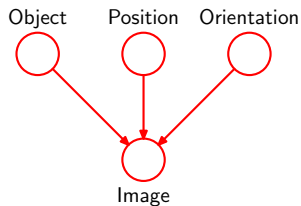
Idea: ricostruisci la distribuzione  $p(x)$  degli esempi “normali” e segnala anomalia se

$$p(x) < \epsilon$$

per una appropriata soglia  $\epsilon > 0$

# Modelli bayesiani a grafo [graphical models]

- Modello generativo
- Sfrutta le dipendenze causali note tra variabili

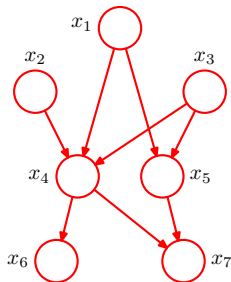


Esempio: osservo Image e voglio ricostruire Object, Position, Orientation conoscendo le distribuzioni condizionate  $p(\text{Image}|\text{Object})$ ,  $p(\text{Image}|\text{Position})$ ,  $p(\text{Image}|\text{Orientation})$  e le distribuzioni a priori

# Modelli bayesiani a grafo [graphical models]

$$p(x) \stackrel{\text{def}}{=} \prod_{j=1}^d p(x_j | \text{parents}_j)$$

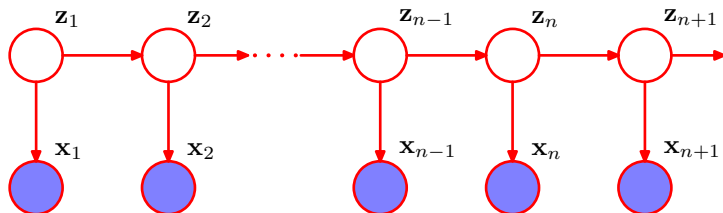
dove  $x = (x_1, \dots, x_d)$  e  $\text{parents}_j$  sono i genitori del nodo  $j$



$$p(x) = p(x_1)p(x_2)p(x_3)p(x_4|x_2, x_3)p(x_5|x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

Rif.: C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006

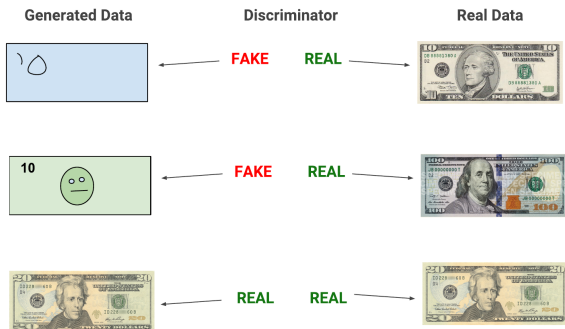
- Modello generativo per dati sequenziali



Rif.: C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006

# Generative Adversarial Networks (GANs)

- Modello generativo
- Due reti neurali in **competizione**: un *generatore* e un *discriminatore*





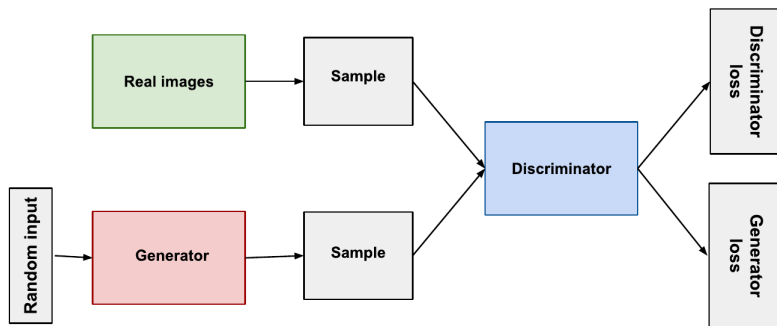
# Generative Adversarial Networks (GANs)

Esempio: generazione di volti



`thispersondoesnotexist.com`

# Generative Adversarial Networks (GANs)



$$g^* = \operatorname{argmin}_g \max_d v(g, d)$$

Rif.: I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press 2016

# Problema degli “esperti” [prediction with expert advice]

- Apprendimento con rinforzo
- Insieme di  $m$  “esperti” che forniscono previsioni
- Esempio: l'indice MIB (borsa di Milano) domani salirà o scenderà?

giorno	$E_1$	$E_2$	$E_3$	realtà
1	scende	sale	scende	scende
2	sale	sale	scende	sale
3	sale	scende	sale	scende
...	...	...	...	...

# Problema degli “esperti”

- $T$  intervalli di tempo: ad ogni intervallo  $t \leq T$ ,
  - L'esperto  $E_i$  fornisce una predizione  $\mathcal{E}_i^t \in \{0, 1\}$
  - Sulla base del vettore  $\mathcal{E}^t$ , prendiamo una decisione  $p^t \in \{0, 1\}$
  - Ci viene comunicato l'esito reale  $o^t \in \{0, 1\}$
- Obiettivo: minimizzare la discrepanza tra il numero di errori

$$\sum_{t=1}^T |p^t - o^t|$$

e gli errori del *miglior esperto a posteriori*:

$$\min_{i=1}^m \sum_{t=1}^T |\mathcal{E}_i^t - o^t|$$

Rif.: N. Cesa-Bianchi, G. Lugosi, *Prediction, Learning, and Games*, Cambridge University Press 2006