Gaussian Process Regression e Classification su Grafi
bozza 12/11/2018

Filippo Maria Castelli

***Abstract*— La Gaussian Process Regression (GPR) e Gaussian Process Classification (GPC) sono tecniche di regressione e classificazione Bayesiana storicamente utilizzate in spazi euclidei $\mathbb{R}^n$. In letteratura si trovano esempi di kernels su grafi definiti a partire dal Laplaciano, il cui utilizzo nel framework dei Gaussian Process rende possibile utilizzare le tecniche di GPC e GPR, già diffuse per gli spazi euclidei, per ottenere risultati per processi definiti sui vertici di un grafo.**

I. Introduzione

Gli algoritmi di Gaussian Process Regression and Gaussian Process Classification, che qui qui presentiamo in un'applicazione su grafi, sono algoritmi di learning Bayesiano: a differenza degli algoritmi di learning "classici", che

1) risolvono un problema di ottimizzazione convessa per identificare un modello di "best fit" per spiegare i dati e poi
2) utilizzano tale modello per effettuare predizioni su punti di input futuri

gli algoritmi Bayesiani non cercano il modello di "best fit" ma calcolano una distribuzione a posteriori sui modelli condizionata alle misure effettuate. La differenza di approccio permette agli algoritmi Bayesiani di utilizzare queste distribuzioni, oltre che per trovare un modello che spieghi i dati, anche di stimare l'incertezza legata alle previsioni del modello stesso.

Gli algoritmi di Gaussian Process Regression e Gaussian Process Classification sono algoritmi di regressione (l'obiettivo è imparare un mapping di un certo spazio di input $A \in \mathbb{R}^n$ ad uno spazio $B \in \mathbb{R}$ di target a valori reali) e di classificazione (concettualmente non troppo dissimile, se non per il fatto che lo spazio di output rappresenta una probabilità e i valori di training sono label binarie).

II. Premesse

Di seguito introduciamo alcuni concetti necessari a comprendere il problema della GPR e GPC.

A. Gaussiane Multivariate

Per cominciare partiamo dalla definizione di *Gaussiana Multivariata* e di alcune sue importanti proprietà:

Una variabile vettoriale stocastica $x \in \mathbb{R}^n$ è detta avere una *distribuzione normale (Gaussiana) multivariata* con media $\mu \in \mathbb{R}^n$ e matrice di covarianza $\Sigma \in \mathbb{S}^n_{++}$ [1] se

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}\,|\Sigma|} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{1}$$

[1]Con $\mathbb{S}^n_{++}$ indichiamo lo spazio delle matrici definite positive $n \times n$

scriviamo in forma abbreviata $\mathcal{N}(\mu,\Sigma)$

Considerando un vettore random $x \in \mathbb{R}^n$ con $x \sim \mathcal{N}(\mu,\Sigma)$, ipotizziamo che le variabili in x siano divise in due set:

$$\begin{aligned} x_A &= [x_1,\dots,x_r]^T \in \mathbb{R}^r \\ x_B &= [x_{r+1},\dots,x_n]^T \in \mathbb{R}^{n-r} \end{aligned} \tag{2}$$

in modo tale da avere

$$\begin{aligned} x = \begin{bmatrix} X_A \\ X_B \end{bmatrix}, \mu = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \\ \Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \end{aligned} \tag{3}$$

per il fatto che $\Sigma = E\left[(x-\mu)(x-\mu)^T\right]$ abbiamo che $\Sigma_{AB} = \Sigma_{BA}^T$.
Valgono le seguenti proprietà:

1) **Normalizzazione** : $\int_x p(x;\mu,\Sigma)dx = 1$
2) **Marginalizzazione** : Le densità di probabilità marginali

$$\begin{aligned} p(x_A) &= \int_{x_B} p(x_a,x_b;\mu,\Sigma)dx_B \\ p(x_B) &= \int_{x_A} p(x_a,x_b;\mu,\Sigma)dx_A \end{aligned} \tag{4}$$

sono Gaussiane.
3) Sono Gaussiane:

$$\begin{aligned} x_A &\sim \mathcal{N}(\mu_A,\Sigma_{AA}) \\ x_B &\sim \mathcal{N}(\mu_B,\Sigma_{BB}) \end{aligned} \tag{5}$$

4) **Distribuzioni Condizionali**: le densità di probabilità condizionali

$$\begin{aligned} p(x_A|x_B) &= \frac{p(x_a,x_b;\mu,\Sigma)}{\int_{x_A} p(x_a,x_b;\mu,\Sigma)dx_Bdx_A} \\ p(x_B|x_A) &= \frac{p(x_a,x_b;\mu,\Sigma)}{\int_{x_B} p(x_a,x_b;\mu,\Sigma)dx_Bdx_A} \end{aligned} \tag{6}$$

sono anch'esse Gaussiane con medie e varianze date da:

$$\begin{aligned} x_A|x_B &\sim \mathcal{N}(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B-\mu_B), \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}) \\ x_B|x_A &\sim \mathcal{N}(\mu_B + \Sigma_{BA}\Sigma_{AA}^{-1}(x_A-\mu_A), \Sigma_{BA}\Sigma_{AA}^{-1}\Sigma_{AB}) \end{aligned} \tag{7}$$

5) **Somma di Distribuzioni**: La somma di variabili indipendenti $y \sim \mathcal{N}(\mu,\Sigma)$ e $z \sim \mathcal{N}(\mu',\Sigma')$ è anch'essa Gaussiana:

$$y + z \sim \mathcal{N}(\mu+\mu',\Sigma+\Sigma') \tag{8}$$

B. Distribuzioni di Probabilità su Funzioni a Dominio Finito

Sia $\mathcal{X} = \{x_1 \dots x_m\}$ un set finito di elementi, ora prendiamo il set $\mathcal{H}$ di possibili funzioni che mappino $\mathcal{X} \to \mathbb{R}$, dal momento che il dominio di una qualsiasi $f(\cdot) \in \mathcal{H}$ è composto di *m* elementi, possiamo rappresentare $f(\cdot)$ come un vettore m-dimensionale $\vec{f} = [f(x_1), f(x_2), \dots, f(x_m)]^T$. Possiamo ipotizzare una distribuzione di probabilità sulle funzioni $f(\cdot) \in \mathcal{H}$ usando la corrispondenza biunivoca tra le $f(\cdot) \in \mathcal{H}$ e le loro rappresentazioni vetoriali $\vec{f}$, in particolare possiamo specificare $\vec{f} \sim \mathcal{N}(\vec{\mu}, \sigma^2 I)$ avendo una distribuzione di probabilità sulle $f(\cdot)$ data da

$$p(h) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{(-\frac{1}{2\sigma^2}(f(x_i)-\mu_i))^2} \tag{9}$$

Possiamo ora descrivere una distribuzione di probabilità su funzioni a dominio finito rappresentandola usando una Gaussiana Multivariata a dimensione finita sulle funzioni di output $f(x_1), \dots, f(x_m)$, per un numero finito di input $x_1 ... x_m$.

Come possiamo ottenere la stessa cosa quando il dominio della funzione ha cardinalità infinita? Per rispondere a questo quesito introduciamo i **Gaussian Process**.

C. Distribuzioni di Probabilità su Funzioni a Domnino non-Finito e Gaussian Processes

Per affrontare il problema ricorriamo ai *processi stocastici*, ed in particolare ai *processi gaussiani*. Un processo stocastico è un insieme di variabili stocastiche $\{f(x) : x \in \mathcal{X}\}$ dipendenti da elementi di $\mathcal{X}$. Un *Gaussian Process* è un processo stocastico tale che ogni sottoinsieme finito di variabili formi una distribuzione Gaussiana multivariata. In particolare, una selezione di variabili random $\{f(x) : x \in \mathcal{X}\}$ è detta estratta da un Gaussian Process con media $m(\cdot)$ e varianza $k(\cdot)$ se, per ogni set finito di elementi $x_1 ... x_m \in \mathcal{X}$, il set associato $f(x_1) ... f(x_m)$ ha distribuzione

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & \dots & k(x_1,x_m) \\ \vdots & \ddots & \vdots \\ k(x_m,x_1) & \dots & k(x_m,x_m) \end{bmatrix} \right) \tag{10}$$

che possiamo indicare in modo più conciso con

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot)) \tag{11}$$

Possiamo, a livello intuitivo, pensare ad una funzione $f(\cdot)$ estratta da un GP come un vettore a dimensionalità estremamente alta estratto da una Gaussiana multivariata a stessa dimensionalità: ogni dimensione della Gaussiana corrisponde ad un elemento x del set $\mathcal{X}$ ed il corrispondente elemento del vettore random rappresenta il valore di $f(x)$. Usando la proprietà di marginalizzazione di Gaussiane multivariate possiamo ottenere la densità di probabilità corrispondente ad ogni sottoinsieme finito di variabili del processo.

Per i processi Gaussiani sono definite una funzione media ed una funzione covarianza in modo tale che

- $m(x) = \mathbb{E}[x]$
- $k(x, x') = \mathbb{E}[(x - m(x))(x' - m(x'))]$

Per quanto riguarda $m(\cdot)$ questa può essere una qualsiasi funzione a valori reali, mentre per $k(\cdot,\cdot)$ è possibile dimostrare che le uniche funzioni di covarianza valide devono generare matrici di Gram

$$K = \begin{bmatrix} k(x_1,x_1) & \dots & k(x_1,x_m) \\ \vdots & \ddots & \vdots \\ k(x_m,x_1) & \dots & k(x_m,x_m) \end{bmatrix} \tag{12}$$

simmetriche e semidefinite positive per un qualasisi set di punti $x_1, \dots, x_m$. Sempre dalla teoria ci viene che, utilizzando il Teorema di Mercer, possiamo utilizzare come matrici di covarianza l'intero repertorio dei kernel semidefiniti positivi, di cui quelli definiti su $\mathbb{R}^n$ sono ben noti nel contesto delle SVM e di cui riportiamo alcuni esempi:

- **Costante**: $k(x, x') = e^{\sigma}$
- **Lineare**: $k(x, x') = \sum_{d=1}^{D} e^{\sigma_d} x_d, x_d$
- **Gaussiano**:$k(x, x') = e^{\sigma_f} e^{-\frac{1}{2} e^{\sigma_l} (x-x')^2}$
- **Periodico**: $k(x, x') = e^{-2e^{\sigma_l} \sin^2 \sigma_\nu \pi (x-x')}$

Un minimo approfondimento sui kernel sarà fornito più avanti quando introdurremo i *Graph Kernels*.

III. Gaussian Process Regression

Vediamo come il concetto di distribuzione di probabilità su funzioni possa essere usato nel contesto della Regressione Bayesiana.

Sia $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ un set di training composto da misure estratte da una distribuzione ignota, il modello di regressione sarà

$$y^{(i)} = f(x^{(i)}) + \epsilon^{(i)}, i = 1, \dots, m \tag{13}$$

dove le $\epsilon^{(i)}$ rappresentano rumore estratto da una distribuzione $\mathcal{N}(0, \sigma^2)$ indipendente. Assumiamo inoltre un GP a priori su funzioni $f(\cdot)$ a media zero

$$f(\cdot) \sim \mathcal{GP}(0; k(\cdot,\cdot)) \tag{14}$$

Sia $T = \{(x_*^{(i)}, y_*^{(i)}\}_{i=1}^{m_*}$ un set di punti di test estratti dalla stessa distribuzione ignota, se per ogni funzione $f(\cdot)$ distribuita dal GP a priori con covarianza $k(\cdot,\cdot)$ la distribuzione marginalizzata per ogni set di punti del set $\mathcal{X}$ deve avere una distribuzione Gaussiana multivariata, questo dovrà essere contemporaneamente vero sia per i punti di training, che per i punti di test, ovvero:

$$\begin{bmatrix} f(x^{(1)}) \\ \vdots \\ f(x^{(m)}) \\ f(x_*^{(1)}) \\ \vdots \\ f(x_*^{(1)}) \end{bmatrix} \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} k(x^{(1)},x^{(1)}) & \dots & k(x^{(1)},x^{(m)}) & k(x^{(1)},x_*^{(1)}) & \dots & k(x^{(1)},x_*^{(m')}) \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ k(x^{(m)},x^{(1)}) & \dots & k(x^{(m)},x^{(m)}) & k(x^{(m)},x_*^{(1)}) & \dots & k(x^{(m)},x_*^{(m')}) \\ k(x_*^{(1)},x^{(1)}) & \dots & k(x_*^{(1)},x^{(m)}) & k(x_*^{(1)},x_*^{(1)}) & \dots & k(x_*^{(1)},x_*^{(m')}) \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ k(x_*^{(m)},x^{(1)}) & \dots & k(x_*^{(m)},x^{(m)}) & k(x_*^{(m)},x_*^{(1)}) & \dots & k(x_*^{(m)},x_*^{(m')}) \end{bmatrix} \right)$$

più conciso:

$$\begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix} \right) \tag{15}$$

Per quanto riguarda l'assunzione che il rumore sia generato da un processo Gaussiano abbiamo

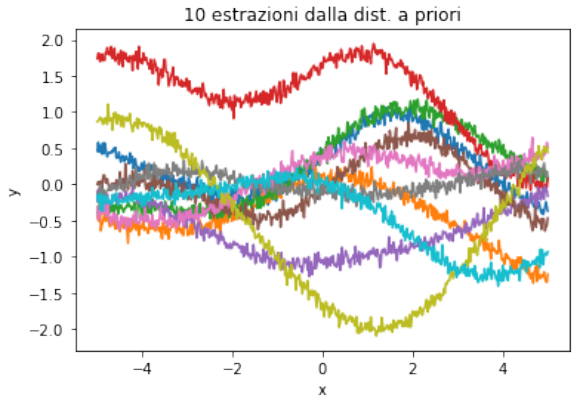

Fig. 1. Estrazioni dalla distribuzione non condizionata: le funzioni sono realizzazioni di processi Gaussiani a media zero sommate ad un termine di rumore.

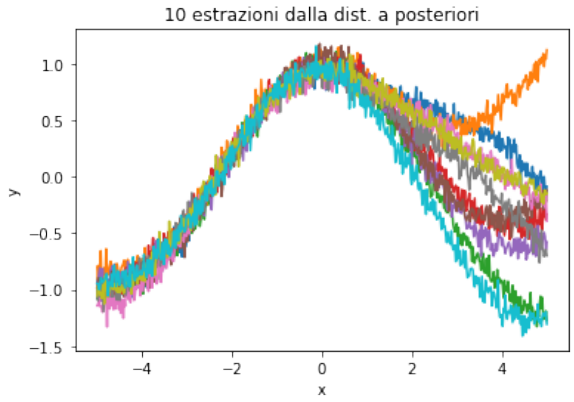

Fig. 2. **Distribuzione a Posteriori**: si vedono chiaramente gli effetti del condizionamento alle misure di training rispetto alle estrazioni di Figura 1.

$$\begin{bmatrix}\vec{\epsilon}\\ \vec{\epsilon}_*\end{bmatrix} \sim \mathcal{N}\Big(\vec{0}, \begin{bmatrix}\sigma^2 I & \vec{0}\\ \vec{0}^T & \sigma^2 I\end{bmatrix}\Big) \tag{16}$$

che sommato al primo processo ci dà

$$\begin{bmatrix}\vec{y}\\ \vec{y}_*\end{bmatrix} = \begin{bmatrix}\vec{f}\\ \vec{f}_*\end{bmatrix} + \begin{bmatrix}\vec{\epsilon}\\ \vec{\epsilon}_*\end{bmatrix} \sim \mathcal{N}\Big(\vec{0}, \begin{bmatrix}K(X,X)+\sigma^2 I & K(X,X_*)\\ K(X_*,X) & K(X_*,X_*)+\sigma^2 I\end{bmatrix}\Big) \tag{17}$$

La Figura 1 dà una visualizzazione di quanto detto finora: una funzione estratta dal processo a priori risulterà avere media nulla ed i termini diagonali determineranno la presenza di rumore gaussiano.

Utilizzando ora le proprietà delle distribuzioni Gaussiane condizionali abbiamo immediatamente che

$$(\vec{y}_*|\vec{y}, X, X_*) \sim \mathcal{N}(\mu^* \Sigma^*) \tag{18}$$

dove

$$\begin{aligned}\mu^* &= K(X_*,X)(K(X,X)+\sigma^2 I)^{-1}\vec{y}\\ \Sigma^* &= K(X_*,X_*)+\sigma^2 I - \\ &K(X_*,X)(K(X,X)+\sigma^2 I)^{-1}K(X,X_*)\end{aligned} \tag{19}$$

??

Il calcolo delle matrici di cui sopra e la stima di μ^* e Σ^* è essenzialmente tutto ciò che è necessario fare per ottenere una predizione con un modello di regressione a GP.

Prendendo un caso reale, impostiamo un problema di regressione misurando alcuni punti di un coseno e utilizziamo queste misure per costruire le matrici di covarianza utilizzando il kernel Gaussiano di cui a II-C: calcolando la distribuzione condizionale utilizzando le regole 6 ed estraendo alcune funzioni da questa otteniamo quello che vediamo in figura 2 tale distribuzione punto per punto avrà media e varianza date dalle **??** e rappresentate in figura 3: la varianza associata ad ogni punto della predizione darà una stima dell'incertezza associata alla stessa, si vede infatti che dove non sono presenti misure questa aumenta.

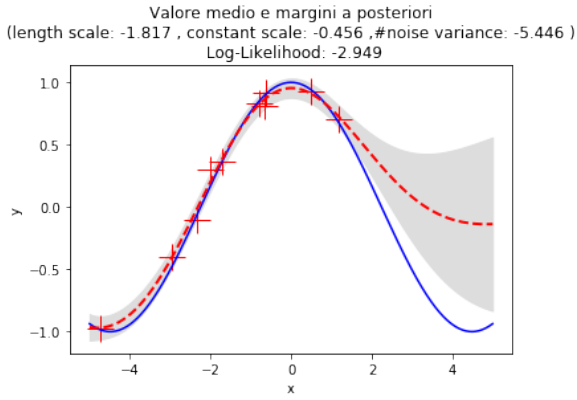

Fig. 3. **Regressione**: sono rappresentati la funzione originale, i punti estratti per impostare il problema di regressione, la media del processo a posteriori come linea tratteggiata e la varianza del processo.

A. *Gaussian Process Classification*

Il problema di classificazione è quello di stimare, data una serie di osservazioni (x, y), dove x sono punti dello spazio, y sono label nel formato $(+1, -1)$, quale possa essere la probabilità $p(y = +1|x_*)$ in un altro punto dello spazio.

L'idea di base è quella di stimare un GP a priori su una "funzione latente" $f(x)$ e successivamente effettuare una trasformazione con una funzione logistica per ottenere una distribuzione a priori su $\pi(x)$[2].

$$\pi(x) = p(y = +1|x) = \sigma(f(x))$$

La funzione latente f gioca un ruolo indiretto: non osserviamo direttamente f (osserviamo invece (X, y)), e neanche siamo interessati a conoscerne i valori, ma piuttosto ci interessa $\pi(x_*)$ nei casi di test x_*: il senso di f è semplicemente quello di consentirci di dare una formulazione conveniente al problema di classificazione, nel framework dei GP.

Possiamo riassumere la soluzione al problema in due step principali:

[2]da notarsi che π è una funzione deterministica di f, e dal momento che f è stocastica, lo è anche π

- In primis viene calcolata la distribuzione sulla variabile latente nei casi di test:

$$p(f_*|X,y,x_*) = \int p(f_*|X,x_*,f)p(f|X,y)df$$

dove, per la Regola di Bayes

$$p(f|X,y) = \frac{p(y|f)p(f|X)}{p(y|X)}$$

- In secundis, la distribuzione a posteriori sulla variabile latente così trovata viene utilizzata per calcolare una predizione probabilstica:

$$\tilde{\pi}_* = p(y_* = +1|X,y,y_*) = \int \sigma(f_*)p(f_*|X,y,x_*)df$$

La differenza rispetto al caso GPR è che, mentre gli integrali equivalenti potevano essere valutati analiticamente in quanto tutte le distribuzioni avevano forma Gaussiana, nel caso della GPC il primo tra gli integrali presentati è decisamente non-Gaussiano e non trattabile analiticamente, così come l'integrale successivo: per certe $\sigma(\cdot)$ potrebbe non esistere nessuna soluzione analitica[3].

Le strade che si prospettano valide per risolvere questo problema sono di due categorie

- Risoluzione numerica con **Metodi MonteCarlo**
- Approssimazioni analitiche, tra cui sono note:
  - **Approssimazione di Laplace**
  - **Expectation Propagation**

L'approccio scelto è quello dell'Approssimazione di Laplace.

B. Approssimazione di Laplace

L'idea è quella di approssimare $p(f|X,y)$ con una ditribuzione Gaussiana $q(f|X,y)$. Effettuando uno sviluppo di Taylor di $\log p(f|X,y)$ attorno al massimo della distribuzione a posteriori otteniamo un'approssimazione Gaussiana:

$$q(f|X,y) = \mathcal{N}(f|\hat{f}, A^{-1}) \propto \exp -\frac{1}{2}(f-\hat{f})^T A(f-\hat{f})) \tag{20}$$

dove

$$\hat{f} = \arg\max_f p(f|X,y) \tag{21}$$

e

$$A = -\nabla\nabla \log p(f|X,y)|_{f=\hat{f}} \tag{22}$$

è l'Hessiana della log-probabilità a posteriori negativa valutata in $\hat{f}$.

Dalla regola di Bayes abbiamo che la distribuzione a posteriori sulle variabili latenti è data da

$$p(f|X,y) = \frac{p(y|f)p(f|X)}{p(y|X)} \tag{23}$$

ma, essendo $p(y|X)$ indipendente da f possiamo unicamente considerare la probabilità a posteriori senza il fattore di normalizzazione nel massimizzare rispetto ad f. Calcolando il logaritmo e ricordando la Gaussianità approssimata di $p(f|X)$ possiamo scrivere

$$\begin{aligned} \Psi(f) &= \log p(y|f) + \log p(f|X) \\ &= \log p(y|f) - \frac{1}{2}f^t K^{-1} f - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \end{aligned} \tag{24}$$

Differenziando rispetto ad f abbiamo

$$\begin{aligned} \nabla\Psi(f) &= \nabla \log p(y|f) - K^{-1}f \\ \nabla\nabla\Psi(f) &= \nabla\nabla \log p(y|f) - K - 1 = -W - K^{-1} \end{aligned} \tag{25}$$

con $W = -\nabla\nabla \log p(y|f)$ che risulta diagonale. [4] Sono possibili diverse forme della likelihood, quelle più usate sono la logistica e la Gaussiana cumulativa, l'esatta formulazione della likelihood e delle sue derivate dipenderà dalla particolare forma scelta.

In corrispondenza del massimo di $\Psi(f)$ abbiamo [5]

$$\nabla\Psi = 0 \rightarrow \hat{f} = K(\nabla \log p(y|\hat{f})) \tag{26}$$

Per trovare il massimo di Ψ usiamo il metodo di Newton iterando

$$\begin{aligned} f_{next} &= f - (\nabla\nabla\Psi)^{-1}\nabla\Psi \\ &= f + (K^{-1} + W)^{-1}(\nabla \log p(y|f) - K^{-1}F) \\ &= (K^{-1} + W)^{-1}(Wf + \nabla \log p(y|f)) \end{aligned} \tag{27}$$

fino a convergenza. Avendo trovato il massimo $\hat{f}$ possiamo specificare

$$q(f|X,y) = \mathcal{N}(\hat{f}, (K^{-1} + W)^{-1}) \tag{28}$$

C. Predizioni

Possiamo esprimere la media a posteriori di f_* in Approssimazione di Laplace in modo analogo al caso precedente, utilizzando il fatto che

$$\hat{f} = K(\nabla \log p(y|\hat{f})) \tag{29}$$

quindi

$$\mathbb{E}_q[f|X,y,x_*] = k(x_*)^T K^{-1}\hat{f} = k(x_*)^T \nabla \log p(y|\hat{f}) \tag{30}$$

Possiamo anche calcolare la varianza $\mathbb{V}_q[f_*|X,y]$ in approssimazione Gaussiana, questa sarà composta di due termini:

$$\begin{aligned} &\mathbb{V}_q[f_*|X,y] = \\ &\mathbb{E}_{p(f_*|X,x_*,f)}[(f_* - \mathbb{E}[f_* - \mathbb{E}[f_*|X,x_*,f])^2] \\ &+\mathbb{E}_{q(f|X,Y)}[\mathbb{E}[f_*|X,x_*,f] - \mathbb{E}[f_*|X,y,x_*])^2] \end{aligned} \tag{31}$$

il primo termine è dovuto alla varianza di f_* ed è dato da

$$k(x_*,x_*) - k(x_*)^T K^{-1} k(x_*) \tag{32}$$

[3] tant'è che nell'implementazione presentata più avanti utilizziamo al suo posto un'approssimazione come combinazione di funzioni integrabili analiticamente

[4] y_i dipende unicamente da f_i e non da $f_{i\neq j}$.

[5] notiamo che, dal momento che $\nabla \log p(y|\hat{f})$ è una funzione non lineare di $\hat{f}$ questa non può essere risolta direttamente.

in modo analogo al caso GPR, il secondo termine è dovuto al fatto che

$$\mathbb{E}[f_*|X, x_*, f] = k(x_*)^T K^{-1} f \tag{33}$$

dipende da f e quindi compare un termine aggiuntivo di

$$k(x_*)^T K^{-1} cov(f|X, y) K^{-1} k(x_*) \tag{34}$$

Sotto approssimazione Gaussiana $cov(f|X, y) = (K^{-1} + W)^{-1}$, quindi

$$\begin{aligned} \mathbb{V}_q[f_*|X, y] = \\ k(x_*, x_*) - k_*^T K^{-1} k_* + k_*^T K^{-1} (K^{-1} + W)^{-1} K^{-1} k_* = \\ k(x_*, x_*) - k_*^T (K + W^{-1})^{-1} k_* \end{aligned} \tag{35}$$

Ottenute media e varianza di f_* rimane solo da trovare le probabilità finali calcolando

$$\tilde{\pi}_* \approx \mathbb{E}_q[\pi_*|X, y, x_*] = \int \sigma(f_*) q(f_*|X, y, x_*) df_* \tag{36}$$

Prendendo un esempio pratico: estraiamo una serie di misure binarie con

$$l(x) = \begin{cases} -1 & \text{se } \cos(x) \leq \frac{1}{2} \\ +1 & \text{altrimenti} \end{cases} \tag{37}$$

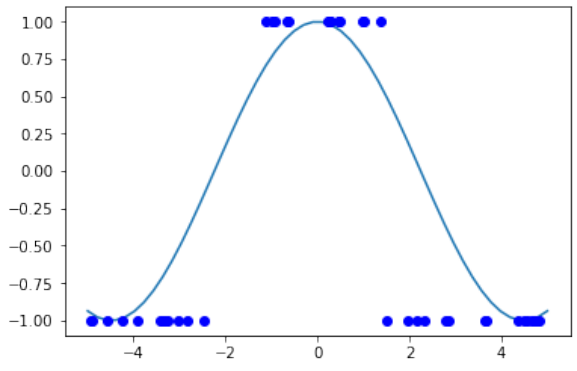

Fig. 4. **Misure Binarie**: le label estratte con la condizione 37 sono in formato $[-1, +1]$, adatte per la GPC.

La media del processo latente e la sua varianza possono essere calcolate con le 30 31 e sono rappresentate in figura 5

Infine possiamo utilizzare la 36 per calcolare la probabilità di appartenenza alle classi, che plottiamo in figura 6

IV. Graph Kernels

Quanto detto finora è perfettamente valido in spazi euclidei $\mathbb{R}^n$ con l'utilizzo dei kernel di cui in II-C, spostandoci in spazi definiti differentemente tali kernel non sono più utilizzabili ed occorre definirne di nuovi.

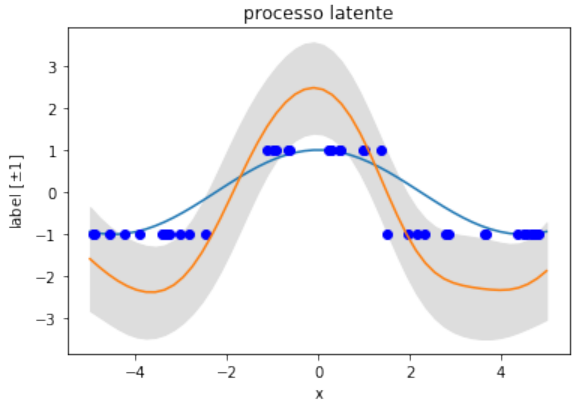

Fig. 5. **Media e Varianza del Processo Latente**

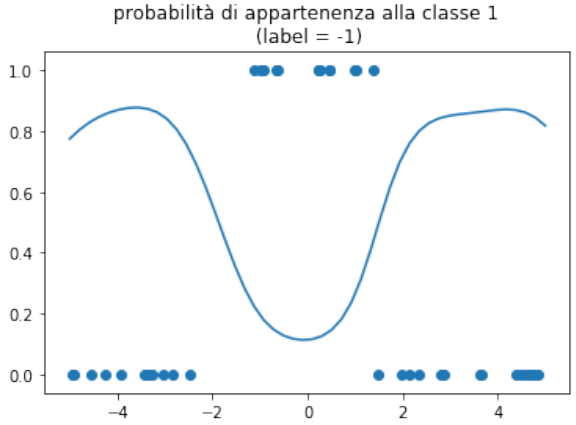

Fig. 6. **Probabilità di Appartenenza alla Prima Classe** ottenuta con la 36

A. Kernels e Teorema di Mercer

Ripartiamo dal concetto di kernel: un kernel, a livello intuitivo, è una funzione K che associa a coppie di elementi di uno spazio χ una misura di *similarità* $K : \chi \times \chi \to \mathbb{R}$, costruendo implicitamente una mappatura $\Phi : \chi \to \mathcal{H}_K$ verso uno spazio di Hilbert $\mathcal{H}_K$ tale che K sia il suo prodotto interno:

$$K(x, x') = < \phi(x), \phi(x') > \tag{38}$$

Vale per i kernel il seguente teorema:

Teorema 1 (Teorema di Mercer): Sia C un sottoinsieme compatto di $\mathbb{R}^n$, se $K(x, x) : C \times C \to \mathbb{R}^n$ è una funzione continua, simmetrica, $K \in L_2(C)$, condizione necessaria e sufficiente perchè essa abbia espansione (può rappresentare un prodotto scalare), ovvero:

$$K(x, x') = \sum_{k=1}^{\infty} a_k \phi_k(x) \phi_k(x')$$

è che K sia semidefinita positiva, ovvero soddisfi la condizione

$$\int_C \int_C f(x) f(x') K(x, x') dx dx' \geq 0$$

per qualsiasi $f(x) \in L_2(C)$

La funzione K per essere un buon kernel, utilizzabile nel senso della 38, deve quindi soddisfare essenzialmente due requisiti:

- deve essere simmetrica:

$$K(x, x') = K(x', x)$$

- deve essere semidefinita positiva.

Il teorema 1 ha analogo discreto, in cui la condizione di semidefinita positività diventa:

$$\sum_{x \in \chi} \sum_{x' \in \chi} f_x f_{x'} K(x, x') \geq 0 \tag{39}$$

per tutti i set di coefficienti reali f_x.

Le **matrici di Gram** associate ai kernel sono definite come $K_{x,x'} = K(x, x')$ ed ereditano, in termini matriciali, le proprietà di simmetria e definita positività.

I problemi di GPR e della GPC su Grafi sono essenzialmente ridotti al problema di trovare una formulazione dei kernel semidefiniti positivi che descrivano la struttura locale degli spazi su cui si vuole operare, è un problema non banale a cui Kondor/Lafferty hanno provato a dare una soluzione [3]

Una forma conveniente per ottenere kernel simmetrici e semidefiniti positivi è suggerita guardando il risultato dell'operazione di esponenziazione di una matrice quadrata:

$$e^{\beta H} = \lim_{n \to \infty} (1 + \frac{\beta H}{n})^n \tag{40}$$

a cui esiste limite pari a

$$e^{\beta H} = I + \beta H + \frac{\beta^2}{2!} H^2 + \frac{\beta^3}{3!} H^3 + \dots \tag{41}$$

vediamo infatti dalla 41 che, quando la matrice H sia simmetrica, la matrice $e^{\beta H}$ risulta simmetrica e semidefinita positiva. Cambiando prospettiva è possibile dimostrare che ogni kernel infinitamente divisibile [6] può essere espresso in forma esponenziale: famiglie di questi kernel possono essere costruite come

$$K(\beta) = K(1)^\beta \tag{42}$$

indicizzate dal parametro β. Scrivendo $K(\beta) = [K(1)^{\frac{\beta}{n}}]^n$ per $n \to \infty$

$$K = \lim_{n \to \infty} (1 + \frac{\beta}{n} \frac{dK}{d\beta}_{\beta=0})^n \tag{43}$$

che diventa la 40 per $H = \frac{dK}{d\beta}_{\beta=0}$

Si intuisce allora che una forma privilegiata in cui cercare i kernel sia quella esponenziale di tipo

$$K = e^{\beta H} \tag{44}$$

[6]Infinita divisibilità significa che per ogni $n \in \mathbb{Z}$ K può essere scritto come $K = K^{\frac{1}{n}} K^{\frac{1}{n}} \dots K^{\frac{1}{n}}$

B. Esponenziali del Laplaciano come Kernel

Sia G un grafo indiretto definito da un set V di vertici ed un set E di edge che connettono coppie v_i, v_j di vertici, indichiamo il fatto che i vertici i e j siano connessi da un edge con $i \sim j$. Definendo la matrice di adiacenza W del grafo come $W_{ij} = 1$ se $i \sim j$ e 0 altrimenti, possiamo definire la matrice dei degree dei vari nodi come una matrice diagonale $D_{ii} = \sum_j W_{ij}$, il **Laplaciano** di G è definito allora come $L = D - W$ ed il **Laplaciano Normalizzato** come $\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. Sono noti dalla teoria spettrale dei grafi i due seguenti teoremi:

Teorema 2 (Spettro di $\tilde{L}$): $\tilde{L}$ è una matrice semidefinita positiva ed i suoi autovalori sono $\lambda_1, \lambda_2, \dots \lambda_n$ con $0 \leq \lambda_i \leq 2$, inoltre il numero dei suoi autovalori nulli è uguale al numero di componenti disgiunte del grafo G che la genera.[7]

L e $\tilde{L}$ possono essere visti come operatori lineari su funzioni $\mathbf{f} : V \to \mathbb{R}$ o equivalentemente su vettori $\mathbf{f} = (f_1, f_2, \dots, f_n)^\intercal$: in questi termini potremmo ridefinire L come

$$< \mathbf{f}, L\mathbf{f} >= \mathbf{f}^\intercal L \mathbf{f} = -\frac{1}{2} \sum_{i \sim j} (f_i - f_j)^2 \forall \mathbf{f} \in \mathbb{R}^n \tag{45}$$

che generalizza rapidamente a grafi con una quantità numerabile di vertici: guardando alla 45 possiamo affermare che L introduce una semi-norma $\|\mathbf{f}\| =< \mathbf{f}, L\mathbf{f} >$ su $\mathbb{R}^n$ e si può intuire come questa quantifichi la variazione locale della funzione f sul suo dominio: tanto più f sarà "mossa", tanto più $\|\mathbf{f}\|$ sarà grande, questo rende il Laplaciano un ottimo candidato per la costruzione di un kernel.

C. Operatori di Regolarizzazione e Laplaciano

Smola e Kondor [11] dimostra come molti dei kernel più utilizzati, come la RBF gaussiana, siano in realtà scrivibili come

$$< f, Pf >= \int |\tilde{f}(\omega)|^2 r(\|\omega\|^2) d\omega =< f, r(\Delta) f > \tag{46}$$

dove $f \in L_2(\mathbb{R}^n)$, $\tilde{f}(\omega)$ indica la trasformata di Fourier di f ed $r(\|\omega\|^2)$ è una funzione di penalizzazione delle componenti in frequenza $|f(\omega)|$ di f (generalmente crescente in $\|\omega\|^2$) e $r(\Delta)$ è l'estensione di r agli operatori ottenuta semplicemente applicando r allo spettro di Δ[2]

$$< f, f(\Delta) f' >= \sum_i < f, \psi_i > r(\lambda_i) < \psi_i, f' > \tag{47}$$

dove λ_i e ψ_i sono rispettivamente gli autovalori e gli autovettori di Δ. I kernel sono ottenuti risolvendo la seguente condizione di consistenza[10]

$$< k(x, \cdot), Pk(x', \cdot) >= k(x, x') \tag{48}$$

[7]Sarà chiaro più avanti, specialmente con l'introduzione degli operatori di regolarizzazione del Laplaciano, che la costruzione di una *norma del Laplaciano* richiederà che esista al più un autovalore nullo, ne consegue i grafi su cui si possono usare i kernel descritti dovranno essere *completamente connessi*.

con questo metodo si riesce a ricavare una corrispondenza tra gli operatori di regolarizzazione $r(\Delta)$ ed i kernel normalmente utilizzati: prendendo ad esempio il kernel RBF Gaussiano:

$$\begin{aligned} r(\|\omega\|^2) &= e^{\frac{\sigma^2}{2}\|\omega\|^2} \\ k(x,x') &= e^{-\frac{1}{2\sigma^2}\|x-x'\|^2} \\ r(\Delta) &= \sum_{i=0}^{\infty} \frac{\sigma^{2i}}{i!}\Delta^i \end{aligned} \tag{49}$$

Tornando al Laplaciano su grafi, la classe di funzionali di regolarizzazione su grafi è definita da

$$< \mathbf{f}, P\mathbf{f} > = < \mathbf{f}, r(\tilde{L})\mathbf{f} > \tag{50}$$

con $r(\tilde{L})$ definita come

$$r(\tilde{L}) = \sum_{i=1}^{m} r(\lambda_i) v_i v_i^{\intercal} \tag{51}$$

e v_i, λ_i sono rispettivamente gli autovettori e gli autovalori ad essi corrispondenti di $\tilde{L}$. Le possibili funzioni di regolarizzazione $r(\lambda)$ includono:

$$\begin{aligned} r(\lambda) &= 1 + \sigma^2\lambda \\ r(\lambda) &= e^{\frac{\sigma^2}{2\lambda}} \\ r(\lambda) &= (aI - \lambda)^{-1} \text{ con } a \geq 2 \\ r(\lambda) &= (aI - \lambda)^{-p} \text{ con } a \geq 2 \\ r(\lambda) &= (\cos \lambda\pi/4)^{-1} \end{aligned} \tag{52}$$

D. Ricavare i Kernel

Facendo riferimento a quanto scritto ad inizio sezione vediamo che è possibile definire uno spazio di Hilbert $\mathcal{H}$ su $\mathbb{R}^m$ in cui valga il prodotto scalare

$$< f, f >_{\mathcal{H}} = < \mathbf{f}, P\mathbf{f} > \tag{53}$$

e si può dimostrare che[8]

$$k(i,j) = [P^{-1}]_{ij} \tag{54}$$

è un kernel valido per un RKHS così definito. A partire dalle 52 i nuovi kernel possono essere semplicemente costruiti come

$$K = \sum_{i=1}^{m} r^{-1}(\lambda_i)\mathbf{v}_i\mathbf{v}_i^{\intercal} \tag{55}$$

ottenendo

$$\begin{aligned} K &= (I + \sigma^2\tilde{L})^{-1} \\ K &= e^{-\frac{\sigma^2}{2\tilde{L}}} \\ K &= (aI - \tilde{L})^{-p} \text{ con } a \geq 2 \\ K &= (\cos \tilde{L}\pi/4)^{-1} \end{aligned} \tag{56}$$

Il secondo dei kernel sopra è stato ricavato[10] in termini di processi diffusivi, si veda l'appendice VII

[8] P^{-1} indica la pseudoinversa quando P non sia invertibile.

V. Esempi di GPR e GPC su Grafi

Introduciamo due esempi pratici: uno di Regressione ed uno di Classificazione su Grafi

A. Regressione

Per semplicità di visualizzazione dei risultati prendiamo come grafo un reticolo quadrato di lato 100 di cui selezioniamo 60 nodi come punti di training e facciamo predizione sui restanti 40. La funzione più semplice che possiamo definire sui nodi è lo shortest path rispetto ad un nodo pivot.

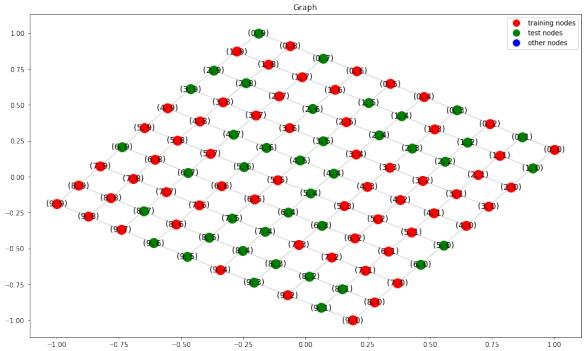

Fig. 7. **Reticolo**: sono indicati in verde i nodi di test ed in rosso i nodi di training

Possiamo impostare un problema di regressione, dove le misure sui nodi di training sono semplicemente le distanze rispetto ad un pivot, possiamo scegliere un kerne di tipo *p-step walk* definito come

$$K = (aI - \tilde{L})^{-p} \text{con } a \geq 2 \tag{57}$$

dove a e p sono i parametri del kernel che in questo caso fissiamo ai valori $a = 2.3$, $p = 4$. Plottando i valori delle predizioni e della varianza associata abbiamo Associando un

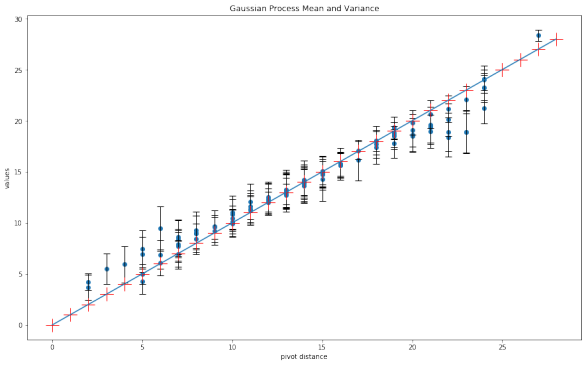

Fig. 8. **Previsioni GPR**: La linea blu indica il valore della distanza dal pivot 0 di ogni singolo nodo, le croci rosse indicano le misure di training, mentre sempre in blu sono indicate le previsioni sui nodi di test con relativa varianza.

colore al valore della funzione sul nodo possiamo visualizzare graficamente in due dimensioni le previsioni:

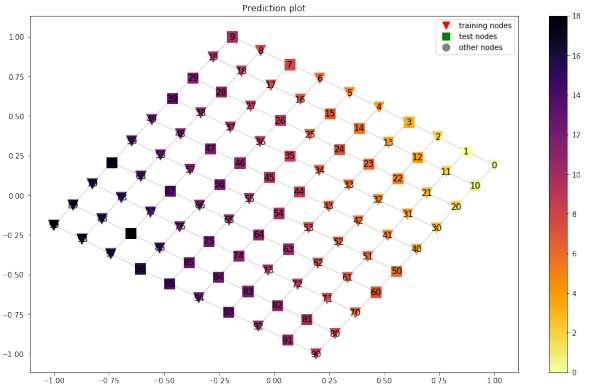

Fig. 9. **Previsioni GPR**: I nodi di training e di test sono rappresentati da simboli diversi, il colore di ogni nodo corrisponde al valore della funzione su di esso.

B. *Classificazione*

Mantenendo lo stesso reticolo, assegnamo ad ognuno dei nodi di training una label binaria definita dalla condizione

$$l_i = \begin{cases} -1 & \text{se } \sin\frac{d_i}{2} \leq 0 \\ +1 & \text{altrimenti} \end{cases} \tag{58}$$

dove d_i indica la distanza dal nodo 0 calcolata con algoritmo di Dijkstra. Inizializzando un classificatore con stesso kernel del caso precedente, parametri $a = 2.1, p = 4$ ed un termine di rumore $\sigma = 0.1$, possiamo rappresentare il processo latente in Figura 10.

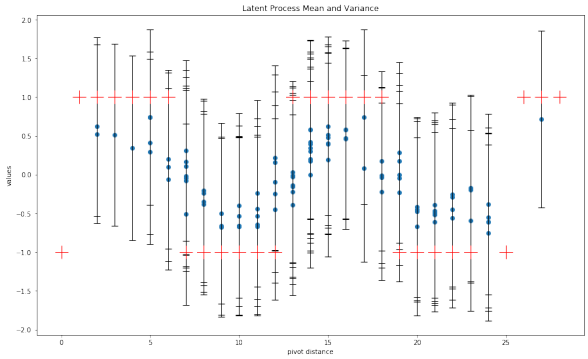

Fig. 10. **Processo Latente**: Le label di training sono indicate in rosso, mentre in blu sono indicate le previsioni sul processo latente con relativa varianza, ordinati per distanza pivotale.

Le previsioni di appartenenza alle due classi sono graficamente rappresentate in Figura 11.

VI. Conclusioni

I metodi di Gaussian Process Regression e Gaussian Process Classification sono facilmente implementabili quando le funzioni da studiare sono definite su $\mathcal{R}^n$ (o equivalentemente esiste una mappatura semplice dallo spazio di definizione ad uno spazio euclideo) e per queste sono disponibili tutti i kernel già noti dallo studio delle SVM, altresì combinazioni lineari di questi come si può verificare usando il teorema di Mercer, quando gli spazi iniziali siano rappresentati da

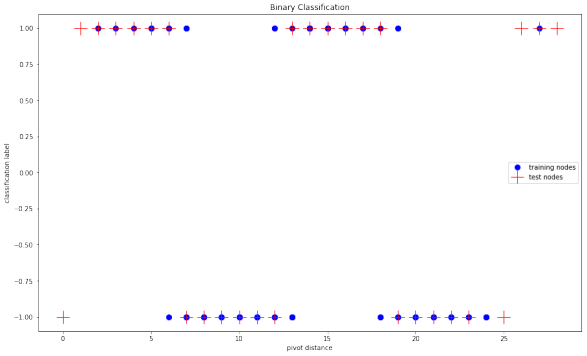

Fig. 11. **Classificazione Binaria**: Previsione di appartenenza binaria, ordinate per distanza pivotale.

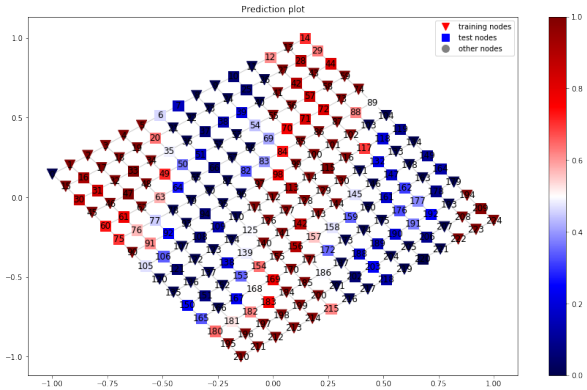

Fig. 12. **Predizioni**: I nodi triangolari sono i punti di training, i nodi quadrati sono i punti di test, il colore di questi ultimi riflette la probabilità di appartenenza alle due classi.

grafi e le funzioni siano definite sui nodi dei grafi stessi risulta necessario trovare altre famiglie di kernel semidefiniti positivi ed in particolare le forme esponenziali del laplaciano del grafo risultano particolarmente adatte a questo scopo, l'introduzione di funzioni di regolarizzazione degli autovalori del laplaciano stesso permettono inoltre di creare kernel che capaci di cogliere caratteristiche diverse dello spazio sul quale agiscono.

VII. Appendice: Diffusion Kernels

E' possibile arrivare alla seconda delle II-C con un modello di tipo diffusivo: si immagini di avere, per ognuno dei V vertici del grafo G una variabile indipendente $Z_i(0)$ a media nulla e varianza σ^2, e che ognuna di queste variabili diffonda una frazione $\alpha \ll 1$ del proprio valore ai vertici adiacenti ad ogni step temporale discreto $t = 1, 2, \ldots$, in modo tale che

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V; j \sim i} Z_j(t) - Z_i(t) \tag{59}$$

Possiamo introdurre un operatore di evoluzione temporale

$$T(t) = (1 + \alpha H)^t \tag{60}$$

che ci permetta di riscrivere la 59 come

$$Z(t) = T(t)Z(0) \tag{61}$$

dove $Z(t) = (Z_1(t), Z_2(t), \ldots, Z_V(t))^\intercal$ è il vettore che rappresenta lo stato delle variabili Z di tutti i vertici al tempo t e $Z(0)$ è lo stato iniziale. Possiamo calcolare la covarianza del set di variabili al tempo t come

$$\begin{aligned} cov_{ij}(t) &= \mathbb{E}[(Z_i(t) - \mathbb{E}Z_i(t))(Z_j(t) - \mathbb{E}Z_j(t))] \\ &= \mathbb{E}[Z_i(t)Z_j(t)] \\ &= \mathbb{E}[(\sum_{i'} T_{ii'}(t)Z_{i'}(0))(\sum_{j'} T_{jj'}(t)Z_{j'}(0))] \end{aligned} \tag{62}$$

considerando che le variabili sono totalmente indipendenti al tempo $t = 0$, $\mathbb{E}[Z_i, Z_j]_{t=0} = \sigma^2\delta(i,j)$ possiamo scrivere

$$\begin{aligned} cov_{ij}(t) &= \sigma^2 \sum_k T_{ik}(t)T_{kj}(t) \\ &= \sigma^2[T(t)^2]_{ij} = \sigma^2 T_{ij}(2t) \end{aligned} \tag{63}$$

Cambiando scala temporale $t = 1, 2, \ldots$ a Δt: $t \to \frac{t}{\Delta t}, \alpha \to \alpha\Delta t$ la definizione dell'operatore di evoluzione temporale diventa

$$T(t) = (1 + \frac{\alpha H}{\frac{1}{\Delta t}})^{\frac{t}{\Delta t}} \tag{64}$$

per la definizione di esponenziale di matrice 40 abbiamo

$$\lim_{\Delta t \to 0} T(t) = e^{\alpha t H} \tag{65}$$

e la 62 diventa

$$cov_{ij}(t) = \sigma^2 e^{2\alpha t H} \tag{66}$$

che ha la stessa forma della seconda delle II-C.

REFERENCES

[1] Chuong B. Do. Gaussian processes. Technical report, Stanford University, 2007.

[2] Nelson. Dunford and Jacob T. Schwartz. *Linear operators*. Interscience Publishers, 1988.

[3] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the ICML*, 2002.

[4] Stephen. Marsland. *Machine learning : an algorithmic perspective*. CRC Press, 2009.

[5] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, 2012.

[6] I Murray. Introduction to Gaussian Processes. *homepages.inf.ed.ac.uk*.

[7] Carl Edward Rasmussen. Gaussian Processes in Machine Learning. pages 63–71. Springer, Berlin, Heidelberg, 2004.

[8] Carl Edward. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[9] Claude. Sammut and Achim Gu¨nther Hoffmann. *Machine learning : proceedings of the Nineteenth International Conference (ICML 2002) : University of New South Wales, Sydney, Australia, July 8-12, 2002*. Morgan Kaufmann Publishers, 2002.

[10] Alex J. Smola, Bernhard Schölkopf, and Klaus Robert Müller. The connection between regularization operators and support vector kernels. *Neural networks : the official journal of the International Neural Network Society*, 11(4):637–649, jun 1998.

[11] Alexander J. Smola and Risi Kondor. Kernels and Regularization on Graphs. pages 144–158. Springer, Berlin, Heidelberg, 2003.

[12] S. V. N. Vishwanathan, Karsten M. Borgwardt, Imre Risi Kondor, and Nicol N. Schraudolph. Graph Kernels. jul 2008.