SOM Bayesian Net

Workshop
“Métodos Bayesianos 2017”
Madrid

7 – 8 Noviembre

Gabriel Antonio Valverde Castilla

Doctorado en Ingeniería Matemática, Estadística e Investigación Operativa.

Directores: Dra. José Manuel Mira McWillians, Dra. Beatriz González
Universidad Complutense de Madrid.

UNIVERSIDAD COMPLUTENSE
MADRID

INDICE

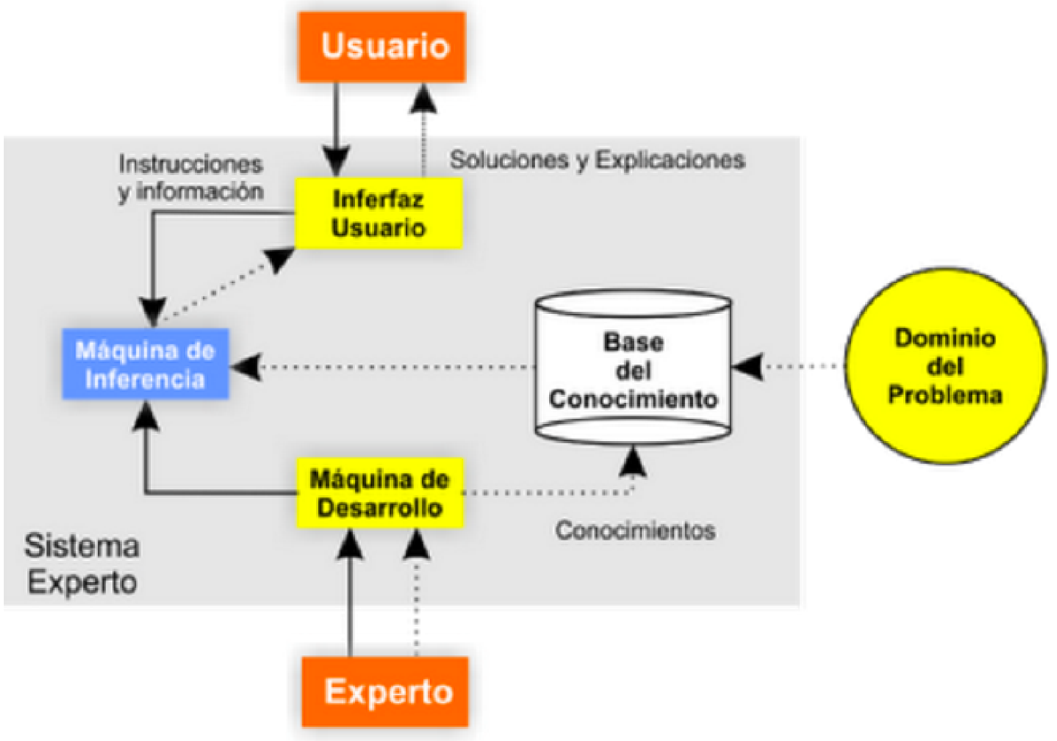
¿Problema inicial?

Comportamiento: Clusterización de clientes por su comportamiento

Marketing: Desarrollo de sistema experto automatizado de recomendación de campañas de marketing

¿Qué es un sistema experto?

Definición: Aplicación informática que en base a un conocimiento declarativo (hechos sobre objetos, acciones y situaciones) y conocimiento de control (seguimientos y resultados de una acción), soluciona un conjunto de problemas sobre un dominio específico.



Ventajas

- Permancencia: No envejece
- Replicación: es fácil de replicar
- Bajo Coste: la posibilidad de replicación compesa el coste inicial.
- Entornos de difícil acceso a ser humano: Big Data
- Fiabilidad: No se ve afectado por su estado.
- Consolidación de varios conocimientos

Limitaciones

- Sentido común: No hay nada obvio (hay que medirlo todo)
- Lenguaje natural: Conversación informal
- Capacidad de aprendizaje: Aprender de errores propios y ajenos
- Flexibilidad: No hay problema en introducir nuevos datos a la hora de resolver un problema en el caso humano
- Conocimiento no estructurado

Reducción dimensional del dominio

Reducción dimensional del Dominio: en la actualidad es común encontrarse con dominios de estudio en los que el volumen de información es elevado, ya sea en términos de registros o de características analizadas.

Nuestro objetivo es simplificar este dominio de forma que se establezca un **entorno general de desarrollo del sistema**, un espacio en el que se identifiquen de forma precisa **perfiles generales** que **reduzcan el espacio de decisiones** para cada caso específico y además pueda ser interpretado rápida y visualmente por el experto.

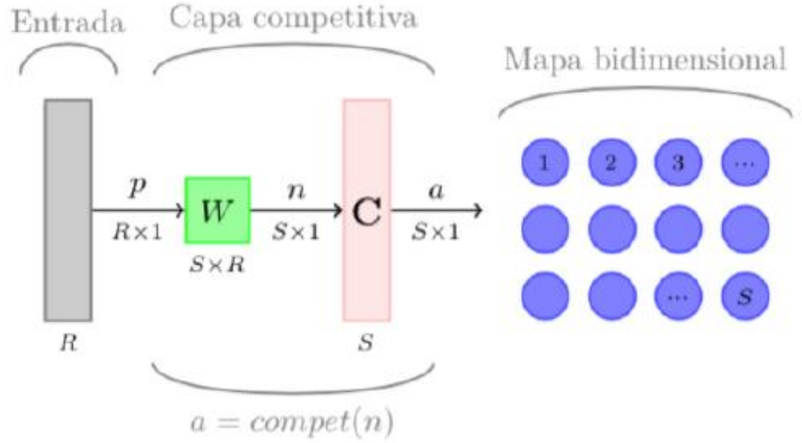
Aportación Big Data y Programación Funcional: Las técnicas Big Data se han desarrollado para permitir el análisis de datos de un gran volumen, variedad o de rápida ingesta.

Es por esto que su aplicación en una Sistema Experto mejora el trabajo del experto que de esa forma puede comprender y estudiar el sistema que conoce desde varias perspectivas y con más información. El Sistema Experto podrá explotar un mayor volumen de información y contrastarla con la información experta que lo configura.

¿ Por qué SOM?

Nuestro objetivo ha evolucionado que planteabamos la tesis sobre el objetivo primero que el desarrollo de sistemas expertos. Encontrado un ámbito de desarrollo que es el de los modelos de SOM las posibilidades de aplicación en Big Data, la constatación de sus ventajas prácticas en este ámbito nos hemos centrado en un trabajo más teórico sobre la validación de las hipótesis mediante argumentos estadísticos.

En la actualidad estamos trabajando para cuantificar de manera exhaustiva las ventajas de la variación del SOM establecida en casos experimentales generados por simulación
Estamos introduciendo rigor científico sobre las conclusiones obtenidas de forma empírica en su aplicación en distintos proyectos y casos de uso.
En esta línea y para poder evaluar las propiedades estadísticas en todo el contexto trabajamos también con la introducción de componente estocástica en cualquiera de sus versiones para poder estimar la incertidumbre de los datos obtenidos como otro parámetro más de la validación del modelo.



1. Definir la topología y en consecuencias el nivel de vecindad
2. Fijar el número de iteraciones $iter$. Normalmente mayor a 500.
3. Seleccionar la medida de distancia a aplicar.
4. Muestrear de forma aleatoria sin repetición sobre el conjunto de entrada, tomamos T_i y los pesos de cada una de las neuronas (inicializados de formar aleatoria o en base a un modelo previo)

$$Por\ ejemplo: d^2 = \sum_{j=1}^{N}(T_{ij} - W_j(k))\text{^}2$$

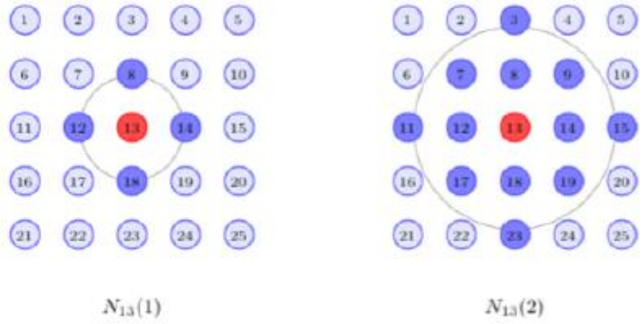
5. La neurona ganadora será: $i^* = argmin(d_i^2)$
6. Se actualizan los pesos de la neurona ganadora y sus vecinos

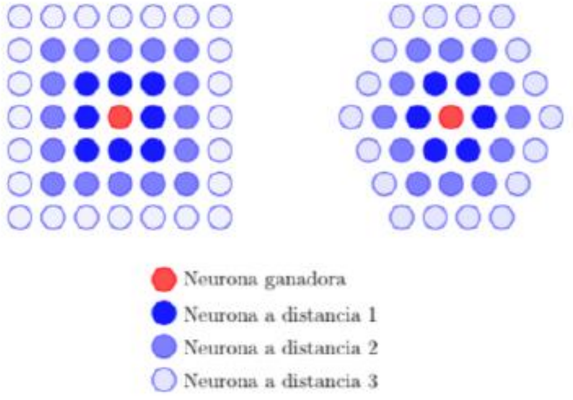
$$W_i(k+1) = W_i(k) + \alpha(k)\big(T_i - W_i(k)\big), para\ i \in N_i^*(d)$$

donde α decrece con número de iteraciones y comienza con valores como $\frac{1}{k}$ o $0.1(1 - \frac{k}{iter})$

7. Se repite hasta completar el número de iteraciones

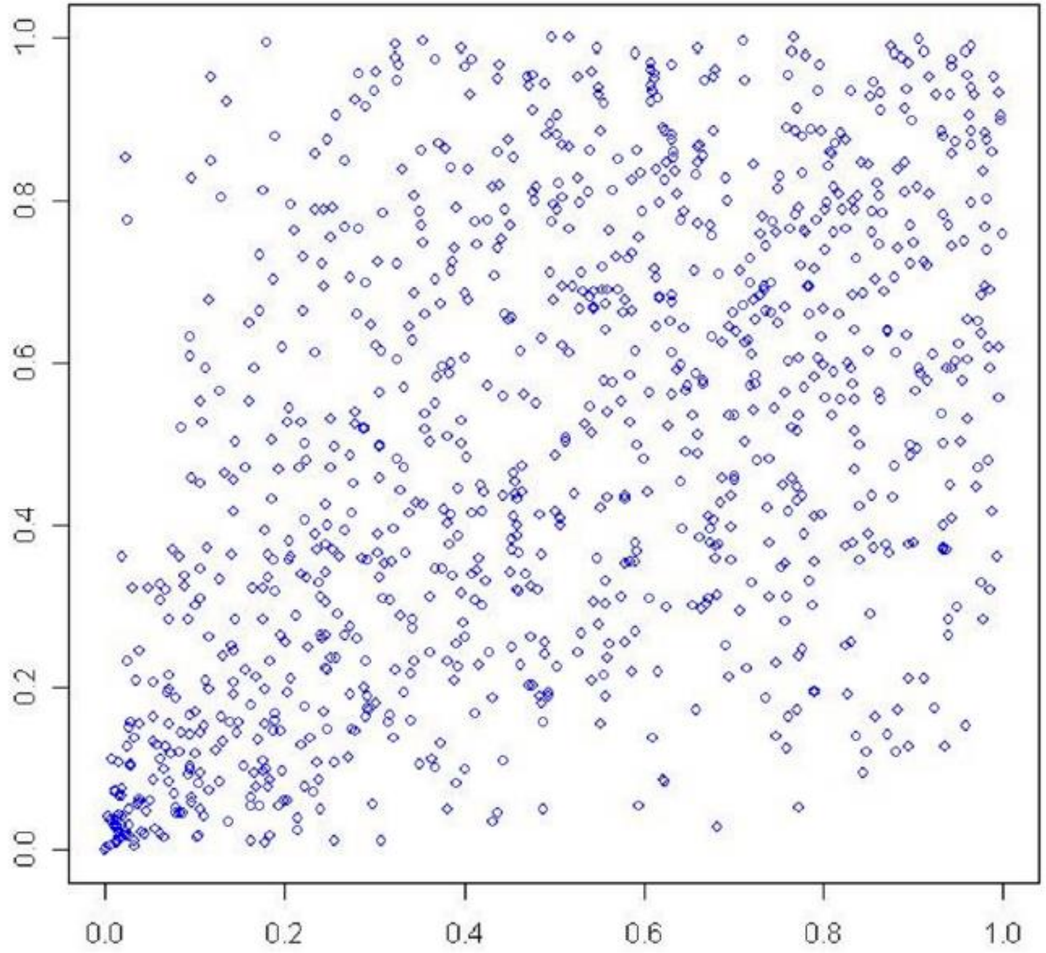
Es la característica que hace que se mantenga la topología, los pesos de neuronas vecinas son también actualizados en cada iteración con un factor de aprendizaje.

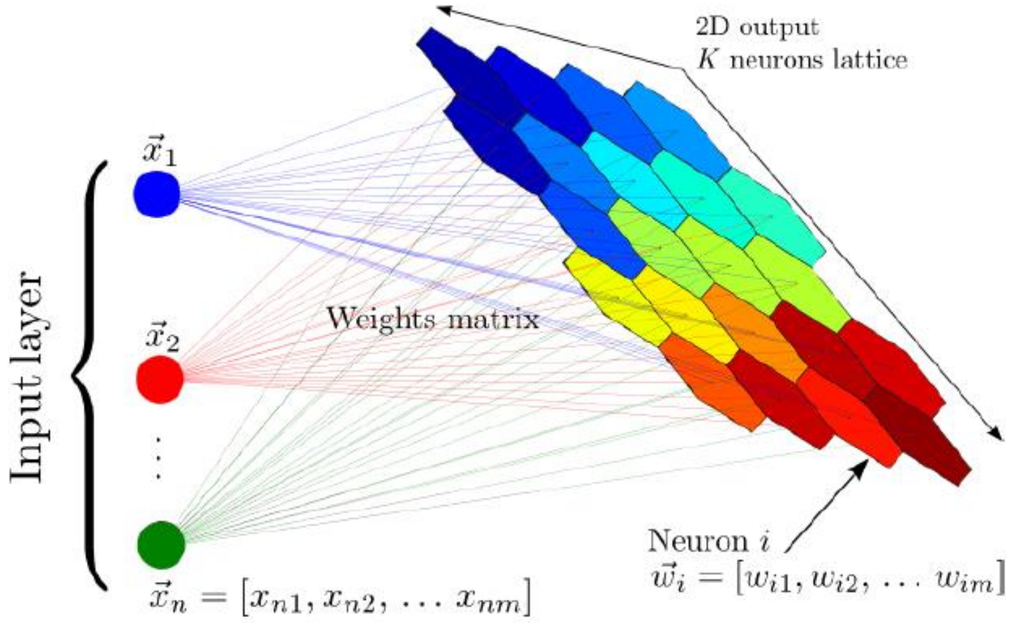




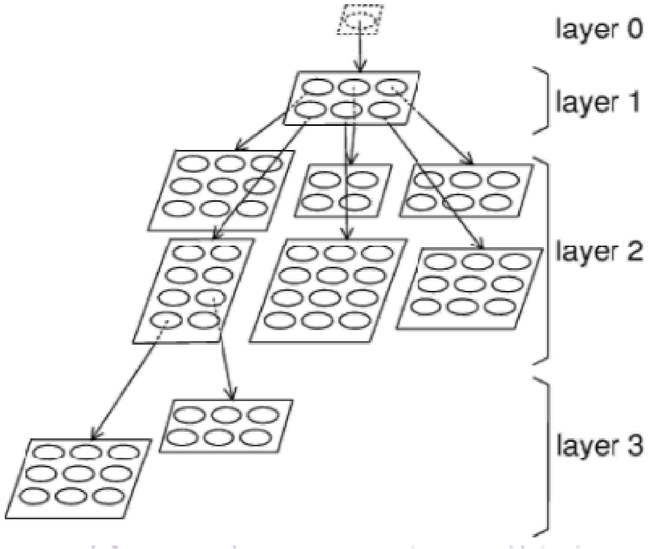
$$N_{i^*}(d) = \{j, d_{i^*j} \leq d\}$$

Introducción a SOM





GHSOM



En este modelo se hace un proceso jerárquico en el que se busca resegmentar grupos de individuos ya clasificados.

Para ello se establece una medida de similaridad entre los elementos asociados a cada neurona, que podría ser:

$$mqe_o = \frac{1}{d} \cdot \|w_o - x\|$$

Siendo w_0 el vector de pesos de la neurona j de la capa 1 y x_{ij} el vector asociado al elemento Xi que ha sido asociado a la neurona j

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)]$$

A continuación se aplica un SOM reducido con todos los individuos clasificados en esa neurona en el paso anterior.

El método de inclusión o no de más capas se decide mediante este criterio:

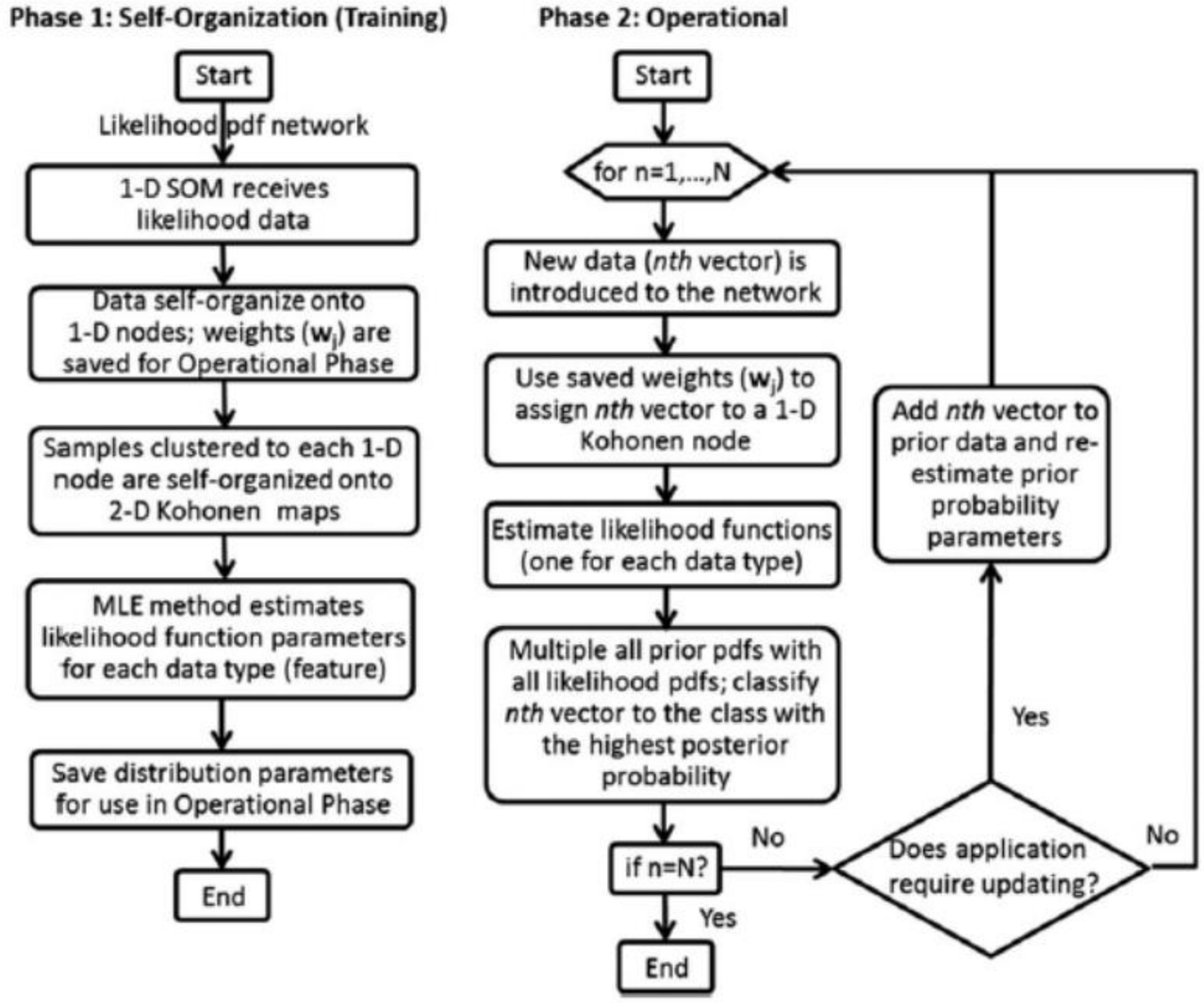
$$\mathbf{mqe}_i > \tau_u \cdot \mathbf{mqe}_0$$

Para evaluar la validez del modelo se estima este valor y sobre el resultado se decide si introducir o no más capas en el mapa m.

$$\mathbf{MQE}_m = \frac{1}{u} \cdot \sum_i \mathbf{mqe}_i$$

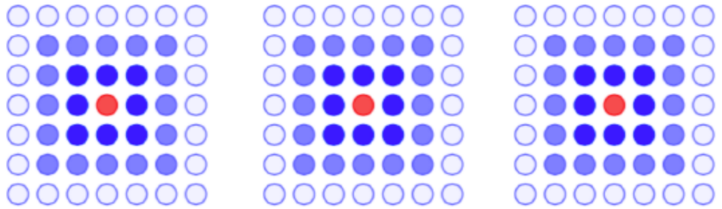
http://www.ifs.tuwien.ac.at/~andi/ghsom/

NBSOM

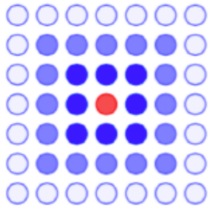

Phase 1: Self-Organization (Training)
Start
Likelihood pdf network
1-D SOM receives likelihood data
Data self-organize onto 1-D nodes; weights (w_j) are saved for Operational Phase
Samples clustered to each 1-D node are self-organized onto 2-D Kohonen maps
MLE method estimates likelihood function parameters for each data type (feature)
Save distribution parameters for use in Operational Phase
End
Phase 2: Operational
Start
for n=1,...,N
New data (nth vector) is introduced to the network
Use saved weights (w_j) to assign nth vector to a 1-D Kohonen node
Estimate likelihood functions (one for each data type)
Multiple all prior pdfs with all likelihood pdfs; classify nth vector to the class with the highest posterior probability
if n=N?
No
Yes
End
Does application require updating?
No
Yes
Add nth vector to prior data and re-estimate prior probability parameters

Parallel SOM

Vamos a introducir paralización para que el modelo pueda aprender del conjunto de datos completo.



Reduce:



1. Segmentar el conjunto de datos de forma balanceada en función del número de máquinas
2. Aplicar la función Map en cada uno de estos subconjuntos:
    a. Definir la topología y en consecuencias el nivel de vecindad
    b. Fijar el número de iteraciones $iter$. Normalmente mayor a 500.
    c. Establecer un método de validación interna [21] y una cota de aceptación con el que aumentar el número de iteraciones.
    d. Seleccionar la medida de distancia a aplicar.
    e. Muestrear de forma aleatoria sin repetición sobre el conjunto de entrada, tomamos T_i y los pesos de cada una de las neuronas (inicializados de formar aleatoria o en base a un modelo previo)

$Por\ ejemplo$: $d^2 = \sum_{j=1}^{N}(T_{ij} - W_j(k))\text{^}2$

Aplicamos las función softmax (muy utilizada en aprendizaje por refuerzo) que se define como:

$$P(j|T) = \frac{e^{d_j^2}}{\sum_{k \in N_{j(k)}} e^{d_i^2}}$$

Esto nos ayudará a definir la estimación del éxito

f. La neurona ganadora será: $i^* = argmax(P(i|T))$

g. Se actualizan los pesos de la neurona ganadora y sus vecinos

$$W_i(k+1) = W_i(k) + \alpha(k)\big(T_i - W_i(k)\big), para\ i \in N_i^*(d)$$

donde α decrece con número de iteraciones y comienza con valores como $\frac{1}{k}$ o $0.1(1 - \frac{k}{iter})$

h. Se repite hasta completar el número de iteraciones

i. Se define uplift la medida del éxito como:

$$uplift_j(ac) = E_j[tasa_t(ac) - tasa_{GC}(ac)] = \sum_{k \in N_j(d)} p(k|j) * (tasa_t(ac) - tasa_{GC}(ac))$$

donde el éxito se define de forma general como alcanzar un objetivo por uno de los elementos de la población como consecuencia de una acción y el uplift modeling [como una técnica de predicción que consiste en medir el impacto directo de un tratamiento trabajando con grupos de control

Donde:

ac: es un intervención sobre uno de los registro buscando un objetivo predefinido

GC: grupo de control

$$tasa_t = \frac{número\ de\ exitos\ en\ target}{número\ de\ acciones\ aplicadas\ por\ target}$$

$$tasa_t = \frac{número\ de\ exitos\ en\ target\ grupo\ de\ control}{número\ de\ ele\square entos\ target\ en\ grupo\ de\ control}$$

3. Aplicamos función Reduce

Dado que el SOM no es un sistema robusto pues depende en gran medida del orden de entrada de la información, no es posible asociar cada una de las neuronas de los distintos modelos realizados por máquina a su homologa en las demás máquinas. La solución es por tanto aplicar el paso 2 con entradas de la capa anterior, con dos posibilidades:

a. Tomar cada una de las neuronas como nuevos registros, esto supone renunciar a la topología establecida en la capa previa y asume esa primera capa como una reducción dimensional, o un modelo generativo donde las muestras son los pesos/centroides calculados en cada neurona.

b. Para mantener de topología en lugar de considerar las neuronas consideramos las siguiente transformada:

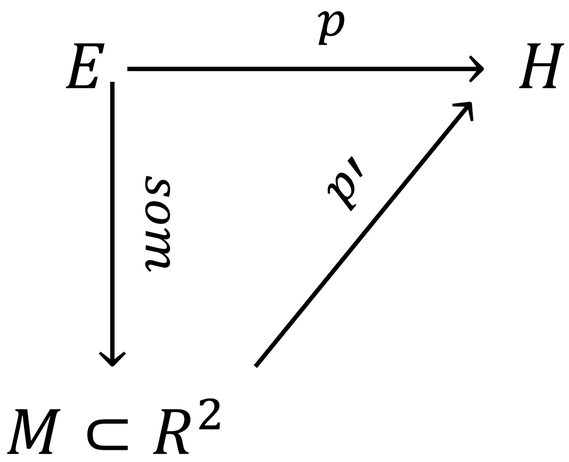
$$W^*_{i_{T_l}} = \sum W_{j_{T_l}} * p\big(j_{T_l}|i_{T_l}\big),\ para\ j_{T_l} \in N^*_{i_{T_l}}(d)$$

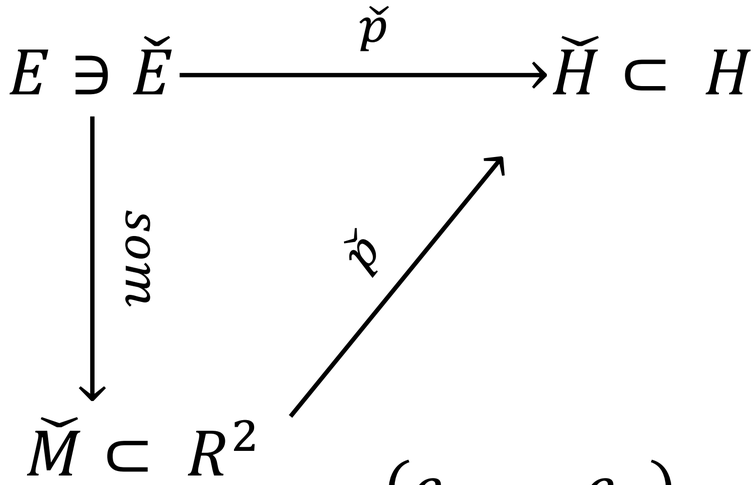
4. Se realiza una validación del modelo comparando la estimación de upfift sobre los datos de entrenamiento y el uplift real obtenido para cada acción.

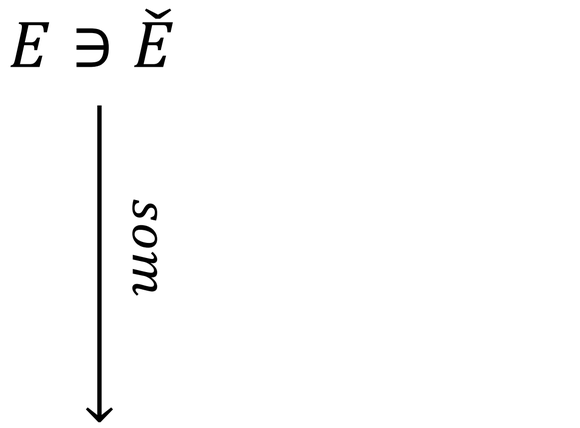
Sea $T_j \in T$ de define

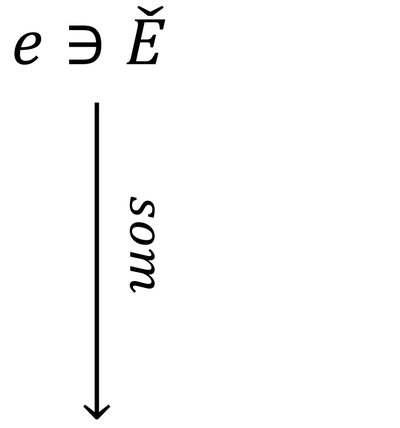
$$uplift_{T_j}(ac) = \sum uplift_{T_j}(ac) * p\big(j\big|T_j\big), j \in Map_{final}$$

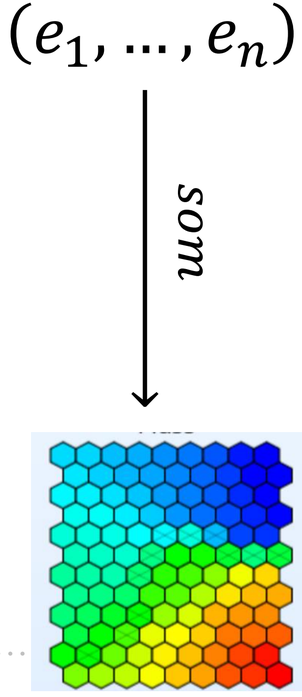
$$uplift_{dif} = uplift_{training}(ac) - uplift_{real}(ac) = \frac{\sum uplift_{T_j}(ac)}{M} - uplift_{real}(ac)$$


E
p
H
uos
pı
M ⊂ R²


E ∋ Ě
p̌
Ȟ ⊂ H
uos
p̌
M̌ ⊂ R²


E ∋ Ě
uos
(M, F) ⊂ (R² , f: E x M → [0,1])


e ∋ Ě
uos
(ci, pi = P(e|cᵢ)) ∀i


(e₁, … , eₙ)
uos

Algorithm 3.1: Growing Bayesian self-organizing map

initialization: start with 2 neurons;

m_i are set as data samples randomly selected from the dataset;

Σ_i are set large compared to the variance of the dataset;

$P(c_i)$ are set equally, $1/2$;

$\alpha(0) \gg \beta(0)$, (e.g. $\alpha(0) = 0.5, \beta(0) = 0.1$);

while *a stopping criterion is not reached;*

do

repeat

for $n = 0;\ n < numberOfDataSamples;\ n{+}{+}$ **do**

randomly select an input, x, from the dataset;

select the winning neuron,

$$\nu = \underset{i}{\operatorname{argmax}} \left\{ \hat{P}[c_i|x(t), \hat{\theta}_i] \equiv \frac{\hat{P}(c_i)p(x|c_i, \hat{\theta}_i)}{\Sigma_{j=1}^{K} \hat{P}(c_j)p(x|c_j, \hat{\theta}_j)} \right\};$$

update mean, covariance and probability weights as follows:

$\hat{m}_i(t+1) = \hat{m}_i(t) + \alpha(s)\hat{P}[c_i|x(t), \hat{\theta}_i](t)[x(t) - \hat{m}_i(t)], i \in N_v;$

$\hat{\Sigma}_i(t+1) = \hat{\Sigma}_i(t) + \beta(s)\hat{P}[c_i|x(t), \hat{\theta}_i](t)\{[x(t) - \hat{m}_i(t)][x(t) - \hat{m}_i(t)]^T - \hat{\Sigma}_i(t)\}, i \in N_v;$

$\hat{P}(c_i|t+1) = \hat{P}(c_i|t) + \beta(s)\{\hat{P}(c_i|x(t), \hat{\theta}_i) - \hat{P}(c_i|t)\}, i \in Y;$

end

adjust the learning rates, $\alpha(s)$ and $\beta(s)$:

$\alpha(s+1) = \alpha(0)/(1 + s/100);$

$\beta(s+1) = \beta(0)/(1 + s/100);$

until *a growing criterion is reached*;

grow a new neuron as in Section 3.2;

end

Modelado estocástico som

- Los parámetros sobre los que aplicamos la **metodología estocástica son** los parámetros del mapa, **sus pesos**.
- Como consecuencia sobre cada uno de los elementos tenemos una **probabilidad de pertenencia condicionada al nodo**, lo que nos permite representar cada elemento como una **imagen pixelada** en el espacio de los mapas
- Esto nos permite no sólo trabajar en un espacio restringido a cada uno de los nodos que forman el mapa sino aplicar a **interacciones intermedias** de los nodos que pueden no haberse detectado en el conjunto de entrenamiento
- Este sistema además permite un **modelado dinámico** de la representación del espacio pues los pesos se van modificando de forma estocástica a medida que la población varia, definiendo como a priori la salida de las iteraciones prevista.

- Algunos trabajos relacionados con BSOM son:
  - Guo, Bayesian SOM for data classification and clustering
  - Estimación de **Deep SOM**
- Aunque la pretensión es ser bayesianos, se quedan en una perspectiva probabilística del proceso de selección pero no la lleva hasta último término en cuanto a la actualización de pesos.
- ¿ Qué pasa si la iteración inicial consiste en distribuciones de probabilidad a priori sobre el conjunto de pesos teniendo encuentra correlaciones por la vecindad?

- Esto sería muy interesante para generalizar el clustering y que la dependencia con respecto al momento de obtención de la muestra no fuera tan importante en el proceso

Modelado estocástico som

- Hemos de tener en cuenta que en el espacio original podemos tener una función de distribución que nos represente la probabilidad de pertenencia a cada una de la componentes
- Esto no queda representado de manera eficiente por un SOM hard
- La función objetivo aunque en la bibliografía podemos encontrar frases como :

" …Specially showed that there was no cost function that the SOM will follow exactly,.."

En otros se da una función a optimizar.

- La topología que se pretende conservar es la de los vecinos:

  Sea e $\in E,\ \pi(e) = \pi_{som}(e), siendo\ \pi\ la\ función\ que\ define\ el\ vecindaio\ en\ cada\ paso$

Table I: Variational Principle and Unsupervised Networks.

Term	Variational principle	Unsupervised network
$\boldsymbol{x}$	Input vector	Training/test vector
$\boldsymbol{y}(\boldsymbol{x})$	Input to output transformation	Encoding prescription
$\boldsymbol{x}'(\boldsymbol{y})$	Output to input transformation	Reference vector
$d[\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x}), \boldsymbol{x}'(\boldsymbol{y})]$	Function	Error function
$P(\boldsymbol{x})$	Integration measure	Probability density of training/test vectors
D	Functional	Average error over the training/test set

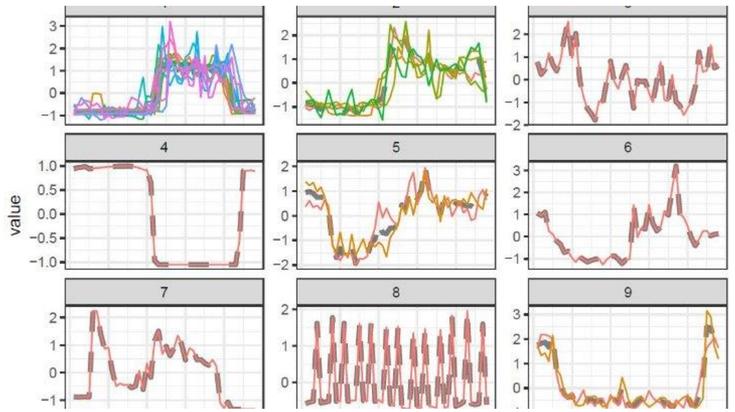
$$D = \int d\boldsymbol{x} P(\boldsymbol{x}) \int d\boldsymbol{n} \pi(\boldsymbol{n}) d\left\{\boldsymbol{x}, \boldsymbol{x}'[\boldsymbol{y}(\boldsymbol{x}) + \boldsymbol{n}]\right\}$$

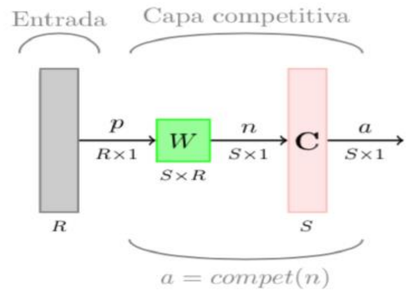
$$D = \int d\boldsymbol{x} P(\boldsymbol{x}) \int d\boldsymbol{n} \pi(\boldsymbol{n}) \int d\boldsymbol{y} d\boldsymbol{x}' P_2(\boldsymbol{x}'|\boldsymbol{y} + \boldsymbol{n}) P_1(\boldsymbol{y}|\boldsymbol{x}) d(\boldsymbol{x}, \boldsymbol{x}')$$

1: Yin(2008) pag 726-727

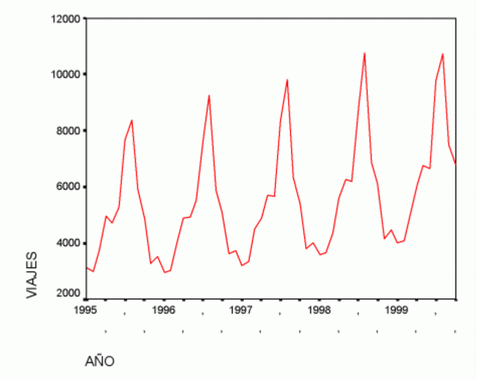
2: A Bayesian Analysis of SOM, Stephen P Luttrell 1994

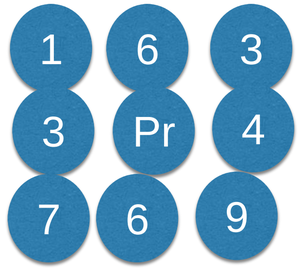
Modelado estocástico som





Prior





Conclusión: Las ventajas del BSOM son:

- En el caso de BSOM permite establecer a priori informativas sobre los pesos de los nodos
- Trabajar con asignaciones probabilísticas a los nodos
- Definir los nodos de forma más informativa, con más párametros:
  - **Peso**
  - **Incertidumbre**
  - **Frecuencia relativa**
- Deep SOM permite generar además redes probabilisticas sobre espacios basados en datos

Contribuciones Actuales

Evolución de Investigación: El proceso de desarrollo de la tesis ha evolucionado hacía una visión más teórica de la misma localizada en el estudio y evaluación de las ventajas e inconvenientes de la alternativa Deep del SOM que se derivó de los distintos proyectos y casos de uso en los que se trabajó en los primeros años.

- Experimento computacional para estudiar ventajas **Deep SOM vs shallow**
- Modelado estocástico **Deep SOM y Shallow SOM**
  - **Frecuentista BoostrapSOM**
  - **Bayesiano BSOM**

Experimentos computacionales

Espacio de Investigación: Como modelo de reducción dimensional el SOM pretende proyectar el conjunto de datos a un espacio de menor dimensión con la mayor representatividad posible.

Esta representatividad, tiene como mayor ventaja su gran interpretabilidad ya que mantiene la topología original del conjunto de datos.

A la hora de comparar la alternativa Deep vs a la Shallow buscamos que manteniendo en la medida de lo posible las virtudes que el modelo plano aporta frente a otros sistemas de reducción dimensional se solventen alguno de los problemas que se detectan el aplicar este algoritmo de aprendizaje por competición.

Experimentos computacionales

Objetivo: Detectar a partir de experimentos computacionales los distintos escenarios en los que el algoritmo planteado tiene ventajas sobre el clásico tanto en propiedades puramente estadísticas como a nivel computacional, permitiendo paralelización.

En ambos casos el experimento se plantea de la siguiente forma:

- Generación de muestras de distribuciones normales multivariantes
- Distribución no uniforme de las muestras obtenidas de cada una de las distribuciones
- Distribución de las muestras en los distintos nodos de ejecución
  - Estimación de **SOM Shallow**
  - Estimación de **Deep SOM**
- Variar parámetros de la mistura para definir condiciones Deep SOM mejora
- Definimos mejora por medio de distintas medidas de comparación de modelos de clustering que evaluar propiedades estadísticas de estos modelos: BIC, DB-Index, SV-Index.

Mixturas de Gaussianas: EM vs BSOM

- Yin y Allison proponen BSOM como modelo de detección de misturas
- Lo derivan de utilizar minimizar la distancia entre la distribución original y la propuesta por el mapa de neuronas, usando Kullback-Leibler como medida de distancia.

$x \in \Omega \subset R^d$ generados por fuentes $\omega_1, \omega_2, \ldots, \omega_K$

$P(\omega_i)$ Probabilidad a priori

Probabilidad final de los datos:

Donde tenemos los parámetros asociados a cada distribución:

$$p(\boldsymbol{x}\,|\,\Theta) = \sum_{i=1}^{K} p(\boldsymbol{x}\,|\,\omega_i, \theta_i) P(\omega_i)$$

$$p(\boldsymbol{x}\,|\,\omega_i, \theta_i) = \frac{1}{(2\pi)^{d/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T \Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right\} \qquad \theta_i = \{\boldsymbol{m}_i,\ \Sigma_i\}$$

1: H. Yin and N.M. Allison “Bayesian self-organising map for Gaussian Mixtures

$$\hat{\boldsymbol{m}}_i = \frac{\sum_{n=1}^{N} \hat{P}[\omega_i \,|\, \boldsymbol{x}(n), \hat{\theta}_i]\boldsymbol{x}(n)}{\sum_{n=1}^{N} \hat{P}[\omega_i \,|\, \boldsymbol{x}(n), \hat{\theta}_i]}$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{n=1}^{N} \hat{P}[\omega_i \,|\, \boldsymbol{x}(n), \hat{\theta}_i][\boldsymbol{x}(n) - \hat{\boldsymbol{m}}_i][\boldsymbol{x}(n) - \hat{\boldsymbol{m}}_i]^T}{\sum_{n=1}^{N} \hat{P}[\omega_i \,|\, \boldsymbol{x}(n), \hat{\theta}_i]}$$

$$\hat{P}(\omega_i) = \frac{1}{N} \sum_{n=1}^{N} \hat{P}[\omega_i \,|\, \boldsymbol{x}(n), \hat{\theta}_i]$$

$$\hat{P}[\omega_i \,|\, \boldsymbol{x}(n), \hat{\theta}_i] = \frac{p[\boldsymbol{x}(n) \,|\, \omega_i, \hat{\theta}_i]\hat{P}(\omega_i)}{\sum_{j=1}^{K} p[\boldsymbol{x}(n) \,|\, \omega_j, \hat{\theta}_j]\hat{P}(\omega_j)}$$

1: H. Yin and N.M. Allison “Bayesian self-organising map for Gaussian Mixtures

Mixturas de Gaussianas: EM vs BSOM

- Utilizando la medida de distancia entre distribuciones Kullback-Leibler:

$$\boldsymbol{I} = -\int \log\left[\frac{\hat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}\right] p(\boldsymbol{x}) d\boldsymbol{x}$$

- Para misturas de Gaussianas tenemos:

$$\frac{\partial \boldsymbol{I}}{\partial \hat{\boldsymbol{m}}_i} = -\int \frac{1}{p(\boldsymbol{x}\,|\,\hat{\Theta})} \frac{\partial p(\boldsymbol{x}\,|\,\hat{\Theta})}{\partial \hat{\boldsymbol{m}}_i} p(\boldsymbol{x}) d\boldsymbol{x}$$

$$= -\frac{1}{\hat{\Sigma}_i} \int \frac{p(\boldsymbol{x}\,|\,\omega_i, \hat{\theta}_i)\hat{P}(\omega_i)}{p(\boldsymbol{x}\,|\,\hat{\Theta})} (\boldsymbol{x} - \hat{\boldsymbol{m}}_i) p(\boldsymbol{x}) d\boldsymbol{x}$$

$$\frac{\partial \boldsymbol{I}}{\partial \hat{\Sigma}_i} = -\int \frac{1}{p(\boldsymbol{x}\,|\,\hat{\Theta})} \frac{\partial p(\boldsymbol{x}\,|\,\hat{\Theta})}{\partial \hat{\Sigma}_i} p(\boldsymbol{x}) d\boldsymbol{x}$$

$$= -\frac{1}{2\hat{\Sigma}_i} \left\{ \int \frac{p(\boldsymbol{x}\,|\,\omega_i, \hat{\theta}_i)\hat{P}(\omega_i)}{p(\boldsymbol{x}\,|\,\hat{\Theta})} \times \left[(\boldsymbol{x} - \hat{\boldsymbol{m}}_i)(\boldsymbol{x} - \hat{\boldsymbol{m}}_i)^T - \hat{\Sigma}_i \right] p(\boldsymbol{x}) d\boldsymbol{x} \right\} \hat{\Sigma}_i^{-1}$$

$$\frac{\partial \boldsymbol{I}}{\partial \hat{P}(\omega_i)} = -\int \frac{1}{p(\boldsymbol{x}\,|\,\hat{\Theta})} \frac{\partial p(\boldsymbol{x}\,|\,\hat{\Theta})}{\partial \hat{P}(\omega_i)} p(\boldsymbol{x}) d\boldsymbol{x} + \lambda \frac{\partial}{\partial \hat{P}(\omega_i)} \left[\sum_{j=1}^{K} \hat{P}(\omega_j) - 1 \right]$$

$$= -\frac{1}{\hat{P}(\omega_i)} \int \left[\frac{p(\boldsymbol{x}\,|\,\omega_i, \hat{\theta}_i)\hat{P}(\omega_i)}{p(\boldsymbol{x}\,|\,\hat{\Theta})} - \lambda \hat{P}(\omega_i) \right] p(\boldsymbol{x}) d\boldsymbol{x}$$

1: H. Yin and N.M. Allison “Bayesian self-organising map for Gaussian Mixtures

Mixturas de Gaussianas: EM vs BSOM

- Tenemos un sistemas de ecuaciones que podríamos resolver si conocieramos:

- Al no conocerlos esto nos lleva a aplicar un método de aproximación estocástico, por ejemplo, Robbins - Monro:

$$M(x) \equiv E\{Y(x)\} = \int yp(y|x)dy = \rho$$

Función de densidad de y respecto a x

Una constante

$$x = \theta \qquad \theta_{n+1} = \theta_n + \alpha_n(\rho - y_n) \qquad \text{(i) } 0 < \alpha_n < 1;\ \text{(ii) } \Sigma\alpha_n \to \infty;\ \text{(iii) } \Sigma\alpha_n^2 < \infty$$

1: H. Yin and N.M. Allison “Bayesian self-organising map for Gaussian Mixtures

Mixturas de Gaussianas: EM vs BSOM

$$\hat{\boldsymbol{m}}_i(n+1) = \hat{\boldsymbol{m}}_i(n) + \alpha(n)\hat{P}[\omega_i \,|\boldsymbol{x}(n), \hat{\theta}_i][\boldsymbol{x}(n) - \hat{\boldsymbol{m}}_i(n)]$$

$$\hat{\Sigma}_i(n+1) = \hat{\Sigma}_i(n) + \alpha(n)\hat{P}[\omega_i \,|\boldsymbol{x}(n), \hat{\theta}_i]\{[\boldsymbol{x}(n) - \hat{\boldsymbol{m}}_i(n)] \times [\boldsymbol{x}(n) - \hat{\boldsymbol{m}}_i(n)]^T - \hat{\Sigma}_i(n)\}, \quad i \in \eta_v$$

$$\hat{P}_i(n+1) = \hat{P}_i(n) + \alpha(n)\{\hat{P}[\omega_i \,|\boldsymbol{x}(n), \hat{\theta}_i] - \hat{P}_i(n)\}$$

η_v is a neighbourhood of the winner v.

- El vecindario definido en este caso de forma general, se estima con la distribución de probabilidad a posteriori y en el caso de gaussianas coincide con la definición de vecindad de un modelo SOM

1: H. Yin and N.M. Allison “Bayesian self-organising map for Gaussian Mixtures

Mixturas de Gaussianas: EM vs BSOM

	P_1	P_2	P_3	$\boldsymbol{m}_1$	$\boldsymbol{m}_2$	$\boldsymbol{m}_3$	Σ_1		Σ_2		Σ_3	
Preset	0.345	0.32	0.335	2.5	−1.8	−0.5	4.0	−0.9	3.5	0.75	2.0	0.2
				1.0	2.2	−0.5	−0.9	0.3	0.75	0.3	0.2	0.3
BSOM	0.368	0.320	0.312	2.52	−1.78	−0.40	4.37	−0.98	3.46	0.75	2.13	0.24
				1.04	2.22	−0.48	−0.98	0.32	0.75	0.29	0.24	0.33
EM	0.361	0.300	0.339	2.23	−1.70	−0.41	5.47	−1.32	3.69	0.74	2.14	0.26
				1.12	2.19	−0.49	−1.32	0.42	0.74	0.34	0.26	0.31

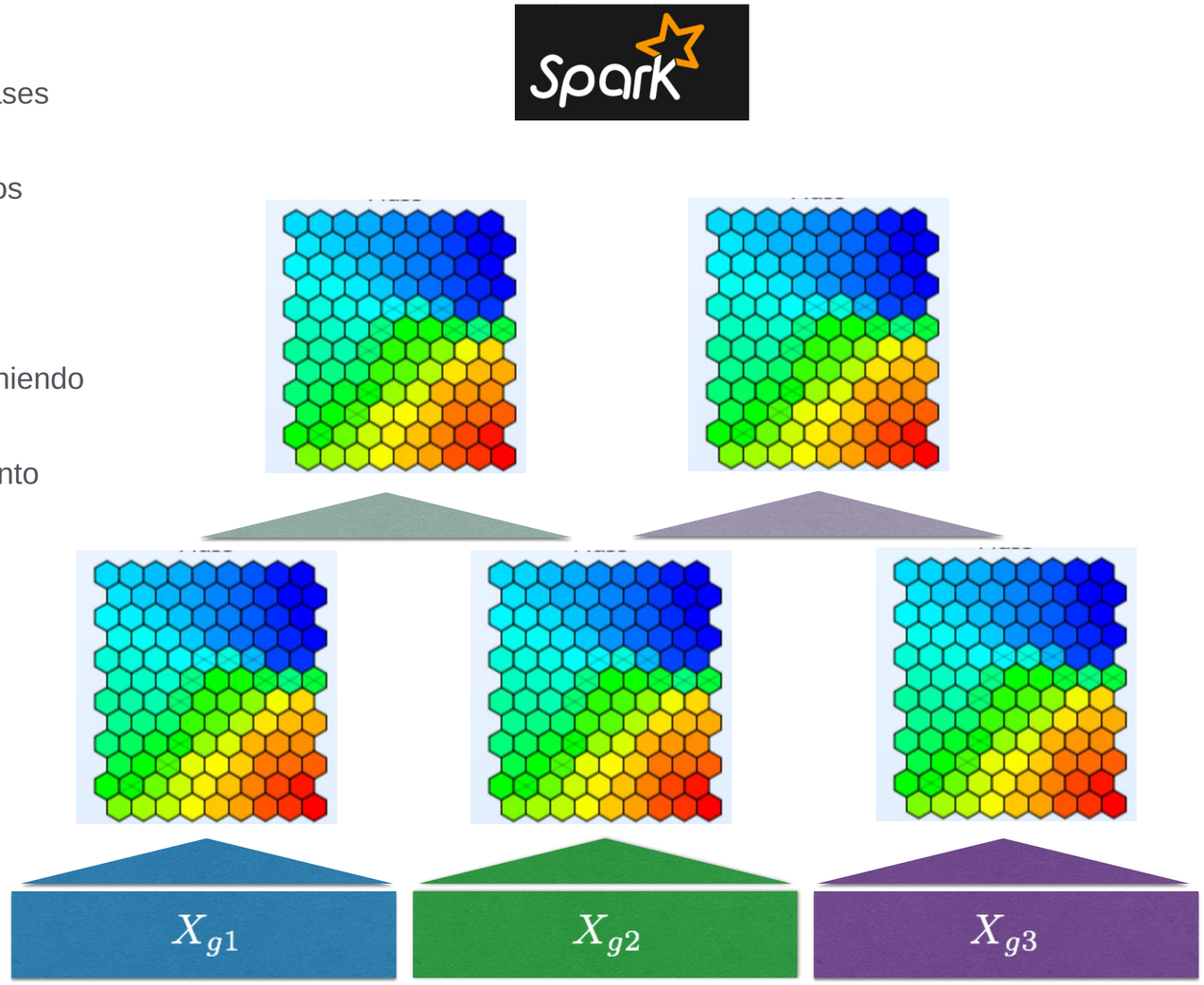
1: H. Yin and N.M. Allison “Bayesian self-organising map for Gaussian Mixtures

Simulaciones para su valoración en proceso de ejecución

- En este caso el conjunto de entrenamiento es mayor, tenemos más clases generadas
- El algoritmo Deep SOM permite una paralización sencilla que mejora los tiempos de ejecución y no sesga por la convergencia
- Pues de forma paralelizada se evalúan distintos bloques de datos
- El resultado mejora al de una sola capa también en propiedades estadísticas. Pues permite trabajar con mayor número de datos manteniendo la incertidumbre sobre los resultados.
- Nos permite establecer variabilidades en los distintos nodos y con distinto nivel de granularidad

$$N = 100000000$$

$$x_i \in X_k \sim N_{150}(\mu_k, \Sigma_k) \qquad X = (x_1, \ldots, x_n)$$



Primeras simulaciones

- Generamos 6 mixturas de distribuciones normales multivariantes de dimensión 3 con correlaciones independientes (dibujar con la distribución y muestra)
- Entrenamos un SOM de una sola capa
- Vemos que ajusta muy bien cada una de las mixturas con un mapa formado por los mismos nodos que componentes

$$M_1 \in X_1 \sim N_3(\mu_1, \Sigma_1)$$
$$M_2 \in X_2 \sim N_3(\mu_2, \Sigma_2)$$
$$M_3 \in X_3 \sim N_3(\mu_3, \Sigma_3)$$
$$M_4 \in X_4 \sim N_3(\mu_4, \Sigma_4)$$
$$M_5 \in X_5 \sim N_3(\mu_5, \Sigma_5)$$
$$M_6 \in X_6 \sim N_3(\mu_6, \Sigma_6)$$

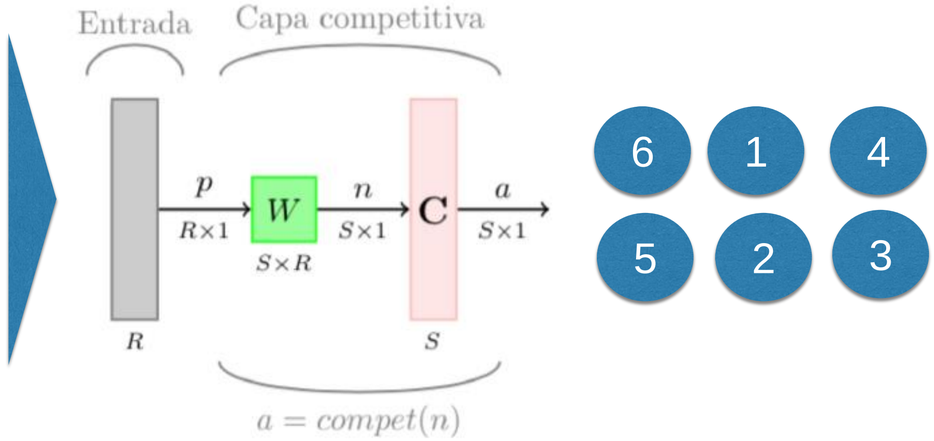
$$N = 100000$$

$$\mu_1 = (2,10,16)$$
$$\mu_2 = (3,11,17)$$
$$\mu_3 = (7,12,18)$$
$$\mu_4 = (4,12,19)$$
$$\mu_5 = (2,10,21)$$
$$\mu_6 = (1,9,20)$$

$$f_1 = 1/6$$
$$f_2 = 1/6$$
$$f_3 = 1/6$$
$$f_4 = 1/6$$
$$f_5 = 1/6$$
$$f_6 = 1/6$$

$$\Sigma_1 = \begin{matrix} \sigma_{11}^1 & 0 & 0 \\ 0 & \sigma_{22}^1 & 0 \\ 0 & 0 & \sigma_{33}^1 \end{matrix} \quad \Sigma_2 = \begin{matrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{matrix} \quad \Sigma_3 = \begin{matrix} \sigma_{11}^3 & 0 & 0 \\ 0 & \sigma_{22}^3 & 0 \\ 0 & 0 & \sigma_{33}^3 \end{matrix}$$

$$\Sigma_4 = \begin{matrix} \sigma_{11}^4 & 0 & 0 \\ 0 & \sigma_{22}^4 & 0 \\ 0 & 0 & \sigma_{33}^4 \end{matrix} \quad \Sigma_5 = \begin{matrix} \sigma_{11}^5 & 0 & 0 \\ 0 & \sigma_{22}^5 & 0 \\ 0 & 0 & \sigma_{33}^5 \end{matrix} \quad \Sigma_6 = \begin{matrix} \sigma_{11}^6 & 0 & 0 \\ 0 & \sigma_{22}^6 & 0 \\ 0 & 0 & \sigma_{33}^6 \end{matrix}$$



$$\omega_{12} = (1.99,10.22,16.5)$$
$$\omega_{22} = (3,10.6,17.7)$$
$$\omega_{23} = (7.2,11.5,13.2)$$
$$\omega_{13} = (4.1,11.5,16.9)$$
$$\omega_{21} = (2.5,11.3,20.9)$$
$$\omega_{11} = (1,9.5,20.6)$$

- Cuando alguno de los parámetros son muy dispares y aparecen con poca frecuencia estos no aparecen

$$M_1 \in X_1 \sim N_3(\mu_1, \Sigma_1)$$
$$M_2 \in X_2 \sim N_3(\mu_2, \Sigma_2)$$
$$M_3 \in X_3 \sim N_3(\mu_3, \Sigma_3)$$
$$M_4 \in X_4 \sim N_3(\mu_4, \Sigma_4)$$
$$M_5 \in X_5 \sim N_3(\mu_5, \Sigma_5)$$
$$M_6 \in X_6 \sim N_3(\mu_6, \Sigma_6)$$

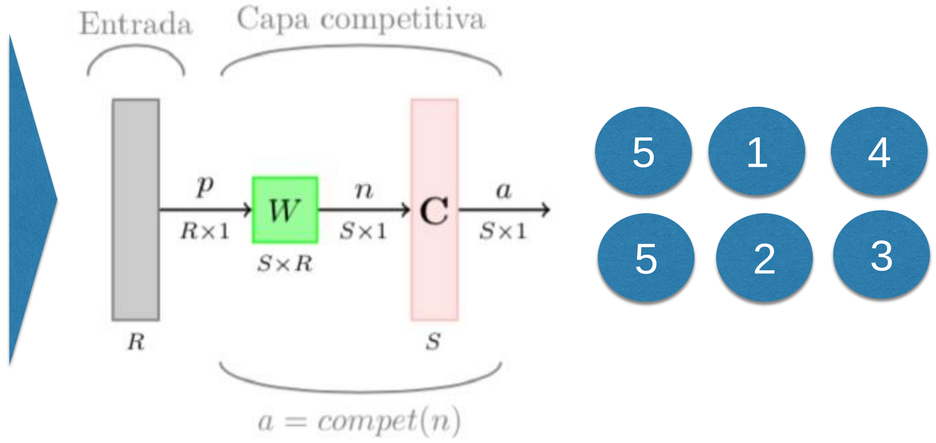
$$\mu_1 = (2,10,16)$$
$$\mu_2 = (3,11,17)$$
$$\mu_3 = (7,12,18)$$
$$\mu_4 = (4,12,19)$$
$$\mu_5 = (2,10,21)$$
$$\mu_6 = (1,9,20)$$

$$f_1 = \frac{1}{6} + \frac{1}{16}$$
$$f_2 = 1/6$$
$$f_3 = 1/6$$
$$f_4 = 1/6$$
$$f_5 = 1/6$$
$$f_6 = 1/16$$

$$\Sigma_1 = \begin{matrix} \sigma_{11}^1 & 0 & 0 \\ 0 & \sigma_{22}^1 & 0 \\ 0 & 0 & \sigma_{33}^1 \end{matrix} \qquad \Sigma_4 = \begin{matrix} \sigma_{11}^4 & 0 & 0 \\ 0 & \sigma_{22}^4 & 0 \\ 0 & 0 & \sigma_{33}^4 \end{matrix}$$

$$\Sigma_2 = \begin{matrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{matrix} \qquad \Sigma_5 = \begin{matrix} \sigma_{11}^5 & 0 & 0 \\ 0 & \sigma_{22}^5 & 0 \\ 0 & 0 & \sigma_{33}^5 \end{matrix}$$

$$\Sigma_3 = \begin{matrix} \sigma_{11}^3 & 0 & 0 \\ 0 & \sigma_{22}^3 & 0 \\ 0 & 0 & \sigma_{33}^3 \end{matrix} \qquad \Sigma_6 = \begin{matrix} \sigma_{11}^6 & 0 & 0 \\ 0 & \sigma_{22}^6 & 0 \\ 0 & 0 & \sigma_{33}^6 \end{matrix}$$



$$\omega_{12} = (1.99,10.22,16.5)$$
$$\omega_{22} = (3,10.6,17.7)$$
$$\omega_{23} = (7.2,11.5,13.2)$$
$$\omega_{13} = (4.1,11.5,16.9)$$
$$\omega_{21} = (2.5,11.3,20.9)$$
$$\omega_{11} = (2.4,11.5,20.6)$$

Primeras simulaciones

$$M_1 \in X_1 \sim N_3(\mu_1, \Sigma_1)$$
$$M_2 \in X_2 \sim N_3(\mu_2, \Sigma_2)$$
$$M_3 \in X_3 \sim N_3(\mu_3, \Sigma_3)$$
$$M_4 \in X_4 \sim N_3(\mu_4, \Sigma_4)$$
$$M_5 \in X_5 \sim N_3(\mu_5, \Sigma_5)$$
$$M_6 \in X_6 \sim N_3(\mu_6, \Sigma_6)$$

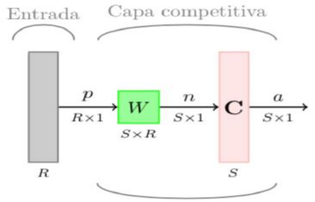
$$\mu_1 = (2,10,16)$$
$$\mu_2 = (3,11,17)$$
$$\mu_3 = (7,12,18)$$
$$\mu_4 = (4,12,19)$$
$$\mu_5 = (2,10,21)$$
$$\mu_6 = (1,9,20)$$

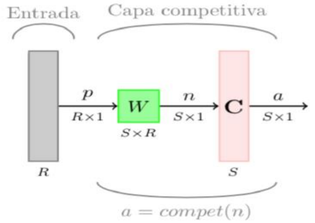
$$f_1 = \frac{1}{6} + \frac{1}{50}$$
$$f_2 = 1/6$$
$$f_3 = 1/6$$
$$f_4 = 1/6$$
$$f_5 = 1/6$$
$$f_6 = 1/50$$

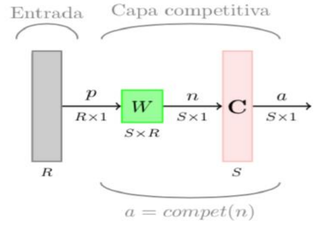
$$\Sigma_1 = \begin{matrix} \sigma_{11}^1 & 0 & 0 \\ 0 & \sigma_{22}^1 & 0 \\ 0 & 0 & \sigma_{33}^1 \end{matrix} \quad \Sigma_4 = \begin{matrix} \sigma_{11}^4 & 0 & 0 \\ 0 & \sigma_{22}^4 & 0 \\ 0 & 0 & \sigma_{33}^4 \end{matrix}$$

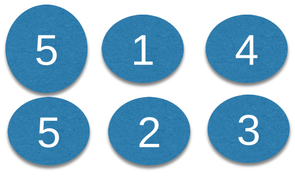
$$\Sigma_2 = \begin{matrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{matrix} \quad \Sigma_5 = \begin{matrix} \sigma_{11}^5 & 0 & 0 \\ 0 & \sigma_{22}^5 & 0 \\ 0 & 0 & \sigma_{33}^5 \end{matrix}$$

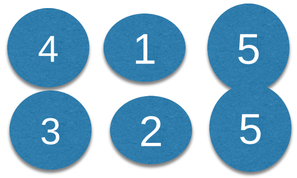
$$\Sigma_3 = \begin{matrix} \sigma_{11}^3 & 0 & 0 \\ 0 & \sigma_{22}^3 & 0 \\ 0 & 0 & \sigma_{33}^3 \end{matrix} \quad \Sigma_6 = \begin{matrix} \sigma_{11}^6 & 0 & 0 \\ 0 & \sigma_{22}^6 & 0 \\ 0 & 0 & \sigma_{33}^6 \end{matrix}$$

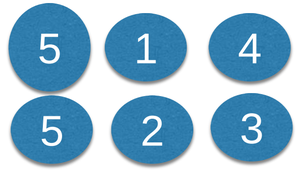


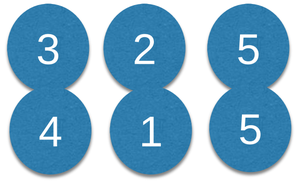












$$\omega_{12} = (2.3,10.2,17.5)$$
$$\omega_{22} = (3,11,17)$$
$$\omega_{23} = (7,12,18)$$
$$\omega_{13} = (4,12,19)$$
$$\omega_{21} = (2,10,21)$$
$$\omega_{11} = (2,10,21)$$

Primeras simulaciones

$$M_1 \in X_1 \sim N_3(\mu_1, \Sigma_1)$$
$$M_2 \in X_2 \sim N_3(\mu_2, \Sigma_2)$$
$$M_3 \in X_3 \sim N_3(\mu_3, \Sigma_3)$$
$$M_4 \in X_4 \sim N_3(\mu_4, \Sigma_4)$$
$$M_5 \in X_5 \sim N_3(\mu_5, \Sigma_5)$$
$$M_6 \in X_6 \sim N_3(\mu_6, \Sigma_6)$$

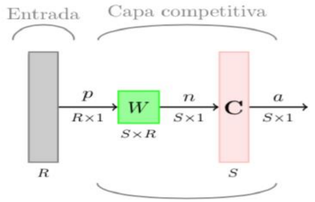
$$\mu_1 = (2,10,16)$$
$$\mu_2 = (3,11,17)$$
$$\mu_3 = (7,12,18)$$
$$\mu_4 = (4,12,19)$$
$$\mu_5 = (2,10,21)$$
$$\mu_6 = (1,9,20)$$

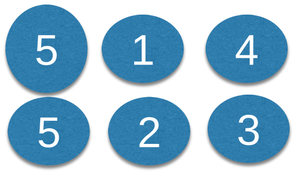
$$f_1 = \frac{1}{6} + \frac{1}{50}$$
$$f_2 = 1/6$$
$$f_3 = 1/6$$
$$f_4 = 1/6$$
$$f_5 = 1/6$$
$$f_6 = 1/50$$

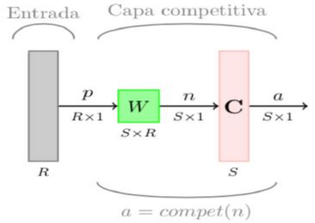
$$\Sigma_1 = \begin{matrix} \sigma_{11}^1 & 0 & 0 \\ 0 & \sigma_{22}^1 & 0 \\ 0 & 0 & \sigma_{33}^1 \end{matrix} \quad \Sigma_4 = \begin{matrix} \sigma_{11}^4 & 0 & 0 \\ 0 & \sigma_{22}^4 & 0 \\ 0 & 0 & \sigma_{33}^4 \end{matrix}$$

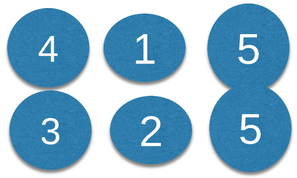
$$\Sigma_2 = \begin{matrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{matrix} \quad \Sigma_5 = \begin{matrix} \sigma_{11}^5 & 0 & 0 \\ 0 & \sigma_{22}^5 & 0 \\ 0 & 0 & \sigma_{33}^5 \end{matrix}$$

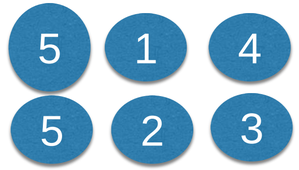
$$\Sigma_3 = \begin{matrix} \sigma_{11}^3 & 0 & 0 \\ 0 & \sigma_{22}^3 & 0 \\ 0 & 0 & \sigma_{33}^3 \end{matrix} \quad \Sigma_6 = \begin{matrix} \sigma_{11}^6 & 0 & 0 \\ 0 & \sigma_{22}^6 & 0 \\ 0 & 0 & \sigma_{33}^6 \end{matrix}$$

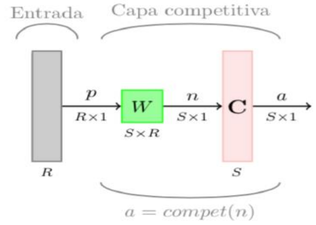


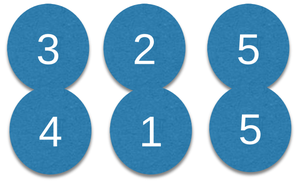












$$\omega_{12} = (2.2,10.1,16.5)$$
$$\omega_{22} = (3.9,9.6,17.7)$$
$$\omega_{23} = (7.3,10,14)$$
$$\omega_{13} = (4,11.7,17)$$
$$\omega_{21} = (2.1,10,21)$$
$$\omega_{11} = (2,9.5,20.6)$$

Primeras simulaciones

$$M_1 \in X_1 \sim N_3(\mu_1, \Sigma_1)$$
$$M_2 \in X_2 \sim N_3(\mu_2, \Sigma_2)$$
$$M_3 \in X_3 \sim N_3(\mu_3, \Sigma_3)$$
$$M_4 \in X_4 \sim N_3(\mu_4, \Sigma_4)$$
$$M_5 \in X_5 \sim N_3(\mu_5, \Sigma_5)$$
$$M_6 \in X_6 \sim N_3(\mu_6, \Sigma_6)$$

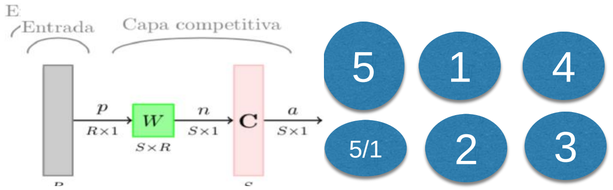
$$\mu_1 = (2,10,16)$$
$$\mu_2 = (3,11,17)$$
$$\mu_3 = (7,12,18)$$
$$\mu_4 = (4,12,19)$$
$$\mu_5 = (2,10,21)$$
$$\mu_6 = (1,9,20)$$

$$f_1 = \frac{1}{6} + \frac{1}{50}$$
$$f_2 = 1/6$$
$$f_3 = 1/6$$
$$f_4 = 1/6$$
$$f_5 = 1/6$$
$$f_6 = 1/50$$

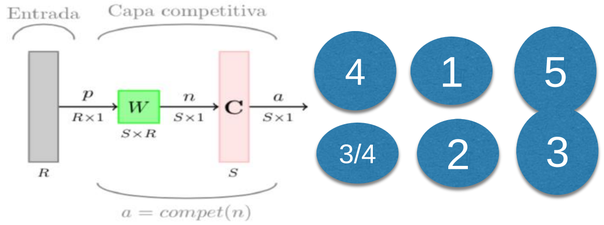
$$\Sigma_1 = \begin{matrix} \sigma_{11}^1 & 0 & 0 \\ 0 & \sigma_{22}^1 & 0 \\ 0 & 0 & \sigma_{33}^1 \end{matrix} \quad \Sigma_2 = \begin{matrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{matrix} \quad \Sigma_3 = \begin{matrix} \sigma_{11}^3 & 0 & 0 \\ 0 & \sigma_{22}^3 & 0 \\ 0 & 0 & \sigma_{33}^3 \end{matrix}$$

$$\Sigma_4 = \begin{matrix} \sigma_{11}^4 & 0 & 0 \\ 0 & \sigma_{22}^4 & 0 \\ 0 & 0 & \sigma_{33}^4 \end{matrix} \quad \Sigma_5 = \begin{matrix} \sigma_{11}^5 & 0 & 0 \\ 0 & \sigma_{22}^5 & 0 \\ 0 & 0 & \sigma_{33}^5 \end{matrix} \quad \Sigma_6 = \begin{matrix} \sigma_{11}^6 & 0 & 0 \\ 0 & \sigma_{22}^6 & 0 \\ 0 & 0 & \sigma_{33}^6 \end{matrix}$$

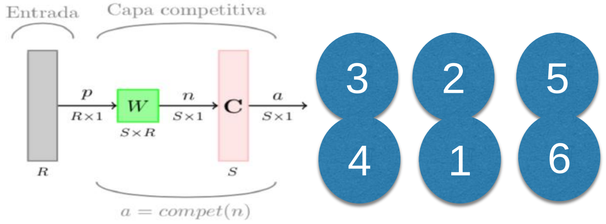
$f_1 = 1/5$
$f_2 = 1/5$
$f_3 = 1/5$
$f_4 = 1/5$
$f_5 = 1/5$

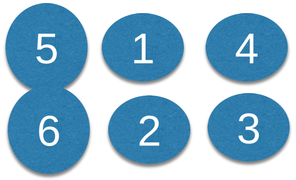


$f_1 = 1/5$
$f_2 = 1/5$
$f_3 = 1/5$
$f_4 = 1/5$
$f_5 = 1/5$



$f_1 = 1/6$
$f_2 = \frac{1}{6}$
$f_3 = \frac{1}{6}$
$f_4 = \frac{1}{6}$
$f_5 = \frac{1}{6}$
$f_6 = \frac{1}{6}$





$$\omega_{12} = (2.3,10.2,17.5)$$
$$\omega_{22} = (4.2,10.6,16.7)$$
$$\omega_{23} = (7.4,10,14)$$
$$\omega_{13} = (4,11.7,17)$$
$$\omega_{21} = (2.1,10,21)$$
$$\omega_{11} = (1,9.5,19.6)$$

- Al distribuir los datos de forma aleatoria el modelo de deep-som no mejora en todos lo casos, en el sentido de que no detectamos una de las poblaciones componentes
- Aunque con suficientes iteraciones por bloque se detecta
- Si la concentramos en uno de los nodos esto si ocurre mejora y en la capa superior se detecta

Trabajo Futuro

Análisis modelo tsne vs SOM

Estado del arte de sistemas de medición de incertidumbre en Deep Learning

Definir sistemas de decisión multicriterio en sistema experto

Estudiar de forma exhaustiva las medidas de incertidumbre en el modelo

Desarrollar Parallel Deep SOM bayesian que permita trabajar con multi-armed sobre distribuciones de probabilidad

Desarrollar Parallel Deep SOM Posibility Functions que permita trabajar con multi-armed sobre distribuciones de probabilidad

Sistemas expertos: sistemas de recomendación

Desarrollo de modelo de evaluación de importancia de variables

Redacción y publicación de artículos

BIBLIOGRAFÍA

[1] C. B. Bhattacharya. (1998) When customers are members: customer retention in paid membership contexts. Journal of the Academy of Marketing Science, 26 (1) 31-44.
[2] E. Rasmusson. (1999) Complaints Can Build Relationships. Sales and Marketing Management, 151 (9) 89-90.
[3] M. Colgate, K. Stewart, R. Kinsella. (1996) Customer defection: a study of the student market in Ireland. International Journal of Bank Marketing, 14 (3) 23-29.
[4] A. D. Athanassopoulos. (2000) Customer Satisfaction Cues To Support Market Segmentation and Explain Switching Behavior. Journal of Business Research, 47 (3), 191-207.
[5] P. Kisioglu & Y. I. Topcu. (2011) Applying Bayesian Belief Network approach to customer churn analysis: A case study on the telecom industry of Turkey. Expert Systems with Applications, 38 (6), 7151-7157.
[6] S. Yoon, J. Koehler & A. Ghobarah. (2010) Prediction of Advertiser Churn for Google AdWords. JSM Proceedings, American Statistical Association.
[7] B. Huanh, M.T. Kechadi & B. Buckley. (2013) Customer Churn Prediction in Telecommunications. Expert Systems with Applications, 39, 1414-1425.
[8] D. Chakraborty et. al. (2012) Method for Predicting Churners in a Telecommunications Network. US Patent 8194830 B2.
[9] B. Eilam, Y. Lubowich & H. Lam. (2013) Method and Apparatus for Predicting Customer Churn.US Patent 8615419 B2.
[10] L. Breiman. (2001) Random Forests.Machine Learning, 45 (1), 5-32. [11] L. Deng et. al. (2013) Recent Advances in Deep Learning for Speech Research at Microsoft. ICASSP, 8604-8608.
[11] G. Dahl et. al. (2013) Improving Deep Neural Networks for LVCSR using Rectified Linear Units and Dropout. ICASSP, 8609-8613.

[12] Hernández-Lobato, J.M y Adams, R, (2015)“Probabilistic backpropagation for Scalable learning of bayesian Neural Networks

[12+1] Gal, Y., y Gharamani , Z., (2016), “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learnung”, Proeedings of the 33 int conference on Machine Learning, Nueva York.

[14] Gal, Y y Gharamani, Z., (2016), “nayesian Convolutional neural networks with Bernoiuii Approximate variaonal Inference

[15] Cheng, S., cao, Y., Sun, J., y Liu, G., (2015), “Visual tracking with Online Incremental Deep learning and particle filter, International Journal of Signal porcessing , image processing and pattern recognition, vol 8, num 2, pp 107-120

[16] Gnawan, A., fanany, M.I., y Jatmiko, W., (2014), Deep extreme tracker based on bootstrap particle filter, Journal of Theoretical and Appllied information technology, vol 66, no 3

[17] Liu, N., wang, J. y Gong, Y., (2015), “Deep self-organizaing Map for Visual Classification, IEEE ?

[18] Sokolovskam N., than Hai, N., Clement, K., y zucker, J-D., (2013?), “Dee Self-Organizing maps for efficient Heterogeneous Biomedical Signatures extraction,

[19] Yin, H, y allison, N.M., (2001), “bayesian Sel-organizing maps for gaussian mixtires, IEEE

[20] Guo, X., wang, H y Glass, D., (2013), “bayesian Self-Oganizing Map for data classification and clustering, International Journal of wavelets, muktiresilution and information processing, vol 11, num 5,

[21] Lutrell, S., (1994), “A bayesian analysis of self-organizing Maps”, neural computation, vol 6, pp 767-794

[22] Yin, H., y allison, N.M., (1997), Bayesian learning for selg-organizing maps, Electronics Ltters, vol 33, num 4.

[23] Ruz, G.A., y truong Pham, Duc, 82012), NBSOM,: naive bayes self-organizing map, Neural Compt & applic, 21, 1319-1330.