



DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL SIMULADOR DE EVENTOS DISCRETOS GALATEA

Erik Velasquez

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas
Trabajo de Grado

Mérida, 09 de diciembre, 2016



- ① Introducción
- ② Desarrollo del Sistema Web
- ③ Integración de GALATEA
- ④ Pruebas
- ⑤ Conclusiones y Recomendaciones
- ⑥ Target Detection
- ⑦ Classification
- ⑧ SVD
- ⑨ K-SVD
- ⑩ Bibliography



- No se dispone de una plataforma web acondicionada para que, de manera fácil y rápida se pueda hacer uso del simulador GALATEA.
- Los usuarios y usuarias del Centro de Simulación y Modelado (CESIMO) de la Universidad de los Andes, suelen tener muchas dificultades para ejecutar los modelos de simulación en el simulador GALATEA.



- Desarrollar un sistema web que sirva como base para la simulación de eventos discretos.
- Sistema web amplíe la base de usuarios y usuarias del simulador.
- Ponerlos en contacto con diferentes expertos que trabajan en el CESIMO.
- Hacer que el simulador sea amigable a la web, para así, aprovechar todas las ventajas que nos proporciona.



Diseñar e implementar un sistema web para los usuarios y usuarias, modelistas y simulistas del simulador de eventos discretos GALATEA, que les permita realizar todas las tareas habituales de modelado, codificación y análisis en sus computadores y en la forma que prefieran, pero permitiéndoles realizar las tareas automáticas de compilación, gestión de archivos, simulación y gestión de salidas, en el espacio virtual y con los recursos compartidos de un servidor Web.



- Desarrollar un sistema web que permita el control de usuarios junto con los roles a ser utilizados en el sistema.
- Diseñar e implementar una arquitectura de software que permita la comunicación entre el software de simulación y el sistema web.
- Instalar y configurar en un servidor la arquitectura de software para el sistema de simulación.
- Incorporar el simulador GALATEA como servicio para el sistema web.
- Diseñar y desarrollar un cliente GUI/controlador para un modelo que se pueda gestionar archivos y simular con GALATEA a través del sistema web desarrollado.
- Sistematizar la experiencia de uso del sistema web para simulación.
- Analizar el sistema web desarrollado y establecer las conclusiones.



- GALATEA.
- Control de Usuarios.
- Desarrollo de un Sistema Web.
- Integración del Sistema Web y GALATEA.

- La aplicación Web que funciona como coordinador del sistema, control de usuarios y roles.
- Administración de archivos.
- El Sistema de Integración con GALATEA.

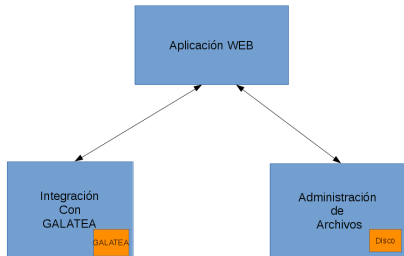
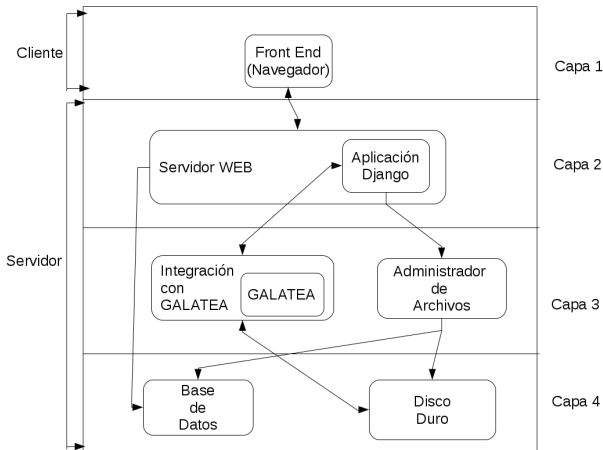


Figura: Sistema de Simulación Web GALATEA.



- Presentación.
- Servidor Web.
- Integración de Procesos.
- Datos.

Figura: Arquitectura de la aplicación web.



- Diseño modular.
- Validación y verificación de cada módulo.
- Arquitectura por capas.

Target Detection and Classification in Multispectral Imagery via Sparse Representation



Se piensa en reportar antecedentes sobre Detección y Clasificación para recordar de donde surge la idea de entrenar un Diccionario (esto de forma muy breve).

Target Detection and Classification in Multispectral Imagery via Sparse Representation



Se piensa en reportar antecedentes sobre Detección y Clasificación para recordar de donde surge la idea de entrenar un Diccionario (esto de forma muy breve).

Target Detection and Classification in Multispectral Imagery via Sparse Representation



Se piensa en reportar antecedentes sobre Detección y Clasificación para recordar de donde surge la idea de entrenar un Diccionario (esto de forma muy breve).



Sea $\mathbf{y} \in \mathbb{R}^n$ un pixel de observacion multiespectral (pixel de una imagen multiespectral). Entonces

- \mathbf{y} se representa como un vector $n - dimensional$ cuyos elementos corresponden a la respuesta a varias bandas espectrales

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^T$$

- En nuestro caso, se tienen 4 bandas espectrales ($n = 4$)



Si \mathbf{y} es un **pixel de fondo**, éste puede ser representado aproximadamente como una combinación lineal de átomos de fondo $\{\mathbf{d}_i\}_{i=1,2,\dots,K_b}$

$$\mathbf{y} \approx \alpha_1 \mathbf{d}_1^b + \alpha_2 \mathbf{d}_2^b + \dots + \alpha_{K_b} \mathbf{d}_{K_b}^b$$

$$\mathbf{y} \approx \underbrace{\begin{bmatrix} \mathbf{d}_1^b & \mathbf{d}_2^b & \dots & \mathbf{d}_{K_b}^b \end{bmatrix}}_{\mathbf{D}_b} \underbrace{\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{K_b} \end{bmatrix}}_{\mathbf{x}_b}^T$$

$$\mathbf{y} = \mathbf{D}_b \mathbf{x}_b$$

Donde:

- $\mathbf{D}_b \in \mathbb{R}^{n \times K_b}$: Es el diccionario de fondo.
- $\mathbf{x}_b \in \mathbb{R}^{K_b \times 1}$: Es el vector de coeficientes de fondo.



Si \mathbf{y} es un **pixel objetivo**, éste puede ser representado aproximadamente como una combinación lineal de átomos de objetivo $\{\mathbf{d}_i\}_{i=1,2,\dots,K_t}$

$$\mathbf{y} \approx \beta_1 \mathbf{d}_1^t + \beta_2 \mathbf{d}_2^t + \dots + \beta_{K_t} \mathbf{d}_{K_t}^t$$

$$\mathbf{y} \approx \underbrace{\begin{bmatrix} \mathbf{d}_1^t & \mathbf{d}_2^t & \dots & \mathbf{d}_{K_t}^t \end{bmatrix}}_{\mathbf{D}_t} \underbrace{\begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_{K_t} \end{bmatrix}}_{\mathbf{x}_t}^T$$

$$\mathbf{y} = \mathbf{D}_t \mathbf{x}_t$$

Donde:

- $\mathbf{D}_t \in \mathbb{R}^{n \times K_t}$: Es el diccionario de objetivo.
- $\mathbf{x}_t \in \mathbb{R}^{K_t \times 1}$: Es el vector de coeficientes de objetivo.



Para modelar cualquier pixel de observación (**pixel global**), se combinan los diccionarios \mathbf{D}_b y \mathbf{D}_t formando así un **diccionario global** \mathbf{D} . Por tanto, un **pixel global** de prueba se puede escribir como una combinación lineal de unos pocos **átomos globales** $\{\mathbf{d}_i\}_{i=1,2,\dots,K}$. Así

$$\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{D}_b & \mathbf{D}_t \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} \mathbf{x}_b \\ \mathbf{x}_t \end{bmatrix}}_{\mathbf{x}}$$

$$\mathbf{y} = \mathbf{D}\mathbf{x}$$

Donde:

- $\mathbf{D} \in \mathbb{R}^{n \times K}$: Es el diccionario global (con $K = K_b + K_t$).
- $\mathbf{x} \in \mathbb{R}^{K \times 1}$: Es el vector de coeficientes global.



Asumiendo que se conoce el diccionario global \mathbf{D} , \mathbf{x} se puede hallar resolviendo el problema de minimización (ampliamente conocido)

$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq T$$

Una vez conocido (reconstruido) el vector de coeficientes \mathbf{x} , se pueden determinar los respectivos \mathbf{x}_b y \mathbf{x}_t

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_b \\ \mathbf{x}_t \end{bmatrix}$$



Con los correspondientes vectores de coeficientes \mathbf{x}_b y \mathbf{x}_t , se pueden calcular los respectivos errores de reconstrucción

$$\mathbf{r}_b(\mathbf{y}) = \|\mathbf{y} - \mathbf{D}_b\mathbf{x}_b\|_2$$

$$\mathbf{r}_t(\mathbf{y}) = \|\mathbf{y} - \mathbf{D}_t\mathbf{x}_t\|_2$$

Luego, el detector podría plantearse de la sig. manera

$$\mathbf{TD}(\mathbf{y}) = \mathbf{r}_b(\mathbf{y}) - \mathbf{r}_t(\mathbf{y})$$

Donde:

- Si $\mathbf{TD}(\mathbf{y}) > \delta$: Entonces \mathbf{x} es determinado como un pixel objetivo.
- Si $\mathbf{TD}(\mathbf{y}) < \delta$: Entonces \mathbf{x} es determinado como un pixel de fondo.



Para \mathbf{D} se trabajó inicialmente con átomos tomados directamente de muestras sobre la imagen.

- \mathbf{D}_b : átomos conformados de pixeles de fondo, directamente.
- \mathbf{D}_t : átomos conformados de pixeles de objetivo, directamente.

Lo cual, probablemente, conduce a una reconstrucción relativamente pobre (menos discriminativa).

- ▶ Por esta razón, se pensó en el entrenamiento de un diccionario. Apareciendo el estudio del algoritmo K-SVD como mejor opción.
- ▶ Con lo que se espera una reconstrucción más discriminativa, que potencie el rendimiento del detector.



Con sólo entrenar un diccionario \mathbf{D} que se adapte a la data (mediante K-SVD) se procedería con las tareas de reconstrucción y luego detección. Con ello:

- Se espera un mejor rendimiento en la detección debido a una reconstrucción de más fiable.
- No obstante, los procesos de **entrenamiento** y **reconstrucción** se trabajan aisladamente.

Lo anterior, conduce a pensar en algo más ambisioso.

- ▶ Durante el mismo entrenamiento, se trabajen ambas operaciones (entrenamiento/detección) de manera simultánea.



Suponiendo que se presentan ***m*** posibles clases. El diccionario global estaría determinado por la combinación de ***m*** subdiccionarios. (uno para cada clase)

$$\mathbf{D} = \left[\mathbf{D}_1 \quad \mathbf{D}_2 \quad \dots \quad \mathbf{D}_m \right]$$

Para lo cual, resolviendo la tarea de reconstrucción, se obtendría un vector de coeficientes global ***x*** formado por la combinación de ***m*** subvectores de coeficientes (uno por clase).

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{bmatrix}$$



Con los **m** subdiccionrios y vectores de coeficientes, se pueden calcular **m** errores de reconstrucción (uno para cada clase).

$$\mathbf{r}_i(\mathbf{y}) = \|\mathbf{y} - \mathbf{D}_i \mathbf{x}_i\|_2 \quad i=1,2,\dots,m$$

Obteniéndose

$$\mathbf{r}(i) = \begin{bmatrix} \mathbf{r}_1(\mathbf{y}) \\ \mathbf{r}_2(\mathbf{y}) \\ \vdots \\ \mathbf{r}_m(\mathbf{y}) \end{bmatrix}$$

Luego, la clase queda determinada por

$$\text{Class}(\mathbf{y}) = i = \arg \min_i \mathbf{r}(i) \quad i=1,2,\dots,m$$

Singular Value Descomposition

Definition



Cualquier matriz rectangular $\mathbf{A} \in \mathbb{R}^{m \times n}$ se puede factorizar de la forma

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

Donde:

- $\mathbf{U} \in \mathbb{R}^{m \times m}$: Matriz cuyas columnas son los autovectores de $\mathbf{A}\mathbf{A}^T$
- $\mathbf{V} \in \mathbb{R}^{n \times n}$: Matriz cuyas columnas son los autovectores de $\mathbf{A}^T\mathbf{A}$
- $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$: Matriz diagonal (pero rectangular) que contiene la raiz cuadrada de los autovalores no-negativos de \mathbf{V} .
Ellos son los valores singulares de \mathbf{A}



Segun su definición, SVD es una multiplicación de matrices.

- Se puede manejar esta multiplicacion como *columns times rows*

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{u}_1\sigma_1\mathbf{v}_1^T + \mathbf{u}_2\sigma_2\mathbf{v}_2^T + \dots + \mathbf{u}_r\sigma_r\mathbf{v}_r^T$$

- ▶ Cualquier matriz \mathbf{A} (rectangular) se puede **descomponer** como la suma de r submatrices de *rango*–1.
- ▶ Cada matriz está formada por la multiplicacion entre la i –th columna de \mathbf{U} y la i –th fila de \mathbf{V}^T , además de estar ponderadas/escaladas/pesadas por el correspondiente autovalor (**valor singular**).



Dadas un conjunto de N señales de entrenamineto \mathbf{Y} , el algoritmo busca el diccionario \mathbf{D} que conduce a la mejor representacion \mathbf{X} de cada uno de los N miembros en ese conjunto, bajo una estricta *sparsity constraint*.

$$\langle \mathbf{D}, \mathbf{X} \rangle = \arg \min_{\mathbf{D}, \mathbf{X}} \{ \|\mathbf{Y} - \mathbf{DX}\|_F^2 \} \text{ s.t. } \forall i, \|\mathbf{x}_i\|_0 \leq T \quad (1)$$

Para tal fin, trabaja en base a dos etapas

- 1 *Sparse Coding Stage.*
- 2 *Process of Updating the dictionary.*



- Se asume un \mathbf{D} fijo y se considera el problema de optimización (1).
- Se halla la matriz de de representación poco densa \mathbf{X} (*Sparse Coding*)

Esto se logra, resolviendo N problemas

$$\mathbf{x}_i = \arg \min_{\mathbf{x}_i} \|\mathbf{y} - \mathbf{D}\mathbf{x}_i\|_2^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq T$$

Donde la i -th columna en \mathbf{X} , corresponde al vector de codificación \mathbf{x}_i de la i -th señal de entrenamineto \mathbf{y}_i en \mathbf{Y} para el \mathbf{D} fijo.



- Se asume que tanto \mathbf{D} como \mathbf{X} están fijos y se considera el problema de optimización (1).

Recordando la multiplicación matricial vista como *columns times rows*. Entonces

$$\mathbf{DX} = \sum_{j=1}^K \mathbf{d}_j \mathbf{x}_T^j$$

Con lo cual, el término $\|\mathbf{Y} - \mathbf{DX}\|_F^2$ en (1) se puede reescribir como

$$\|\mathbf{Y} - \mathbf{DX}\|_F^2 = \left\| \mathbf{Y} - \sum_{j=1}^K \mathbf{d}_j \mathbf{x}_T^j \right\|_F^2 \quad (2)$$



- Para proponer la actualización de \mathbf{D} , se pone en cuestion un sólo átomo \mathbf{d}_k y la fila de coeficientes \mathbf{x}_T^k correspondientes

Ello se logra reescribiendo (2) de la sig. manera

$$\begin{aligned}\|\mathbf{Y} - \mathbf{DX}\|_F^2 &= \left\| \left(\mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j \right) - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 \\ &= \left\| \mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2.\end{aligned}\tag{3}$$

- Se desea minimizar la diferencia en (3) sabiendo que $\mathbf{d}_k \mathbf{x}_T^k$ es una matriz de *rango*–1. ¿Qué se sugiere?



- ▶ Aplicar SVD a \mathbf{E}_k para determinar la matriz de *rango*—1 más semejante a ella.
- ▶ Esto se realiza K veces, a favor de actualizar cada uno de los K átomos en \mathbf{D} .
- ▶ Lo anterior justifica el nombre del algoritmo, pues se aplica "SVD" K veces (K-SVD)



Muy importante, realmente (3) se redefine antes de aplicar SVD a favor de mantener una representación poco densa. Así

$$\left\| \mathbf{E}_k \mathbf{\Omega}_k - \mathbf{d}_k \mathbf{x}_T^k \mathbf{\Omega}_k \right\|_F^2 = \left\| \mathbf{E}_k^R - \mathbf{d}_k \mathbf{x}_R^k \right\|_F^2.$$

Donde:

- $\mathbf{\Omega}_k$ es la matriz que permite preservar el soporte del vector fila \mathbf{x}_T^k
- \mathbf{x}_R^k son los coeficientes no nulos de \mathbf{x}_T^k
- \mathbf{E}_k^R son las columnas de errores que corresponden a los ejemplos que usan el átomo \mathbf{d}_k

Luego, Se aplica SVD a \mathbf{E}_k^R , con lo cual

$$\mathbf{d}_k \mathbf{x}_R^k = \mathbf{u}_i \sigma_i \mathbf{v}_i^T$$



- [1] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *IEEE J. Sel. Topics Signal Process*, vol. 5, no. 3, pp. 629-640, June 2011.
- [2] Y. Chen, N. M. Nasrabadi, y T. D. Tran, "Hyperspectral Image Classifications using Dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973-3985., Oct. 2011.
- [3] G. Strang, *Linear Algebra and Its Applications*. Fourth Edition.
- [4] M. Aharon, M. Elad, y A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.