Minería de Datos con Agentes (*JaCa-DDM*)

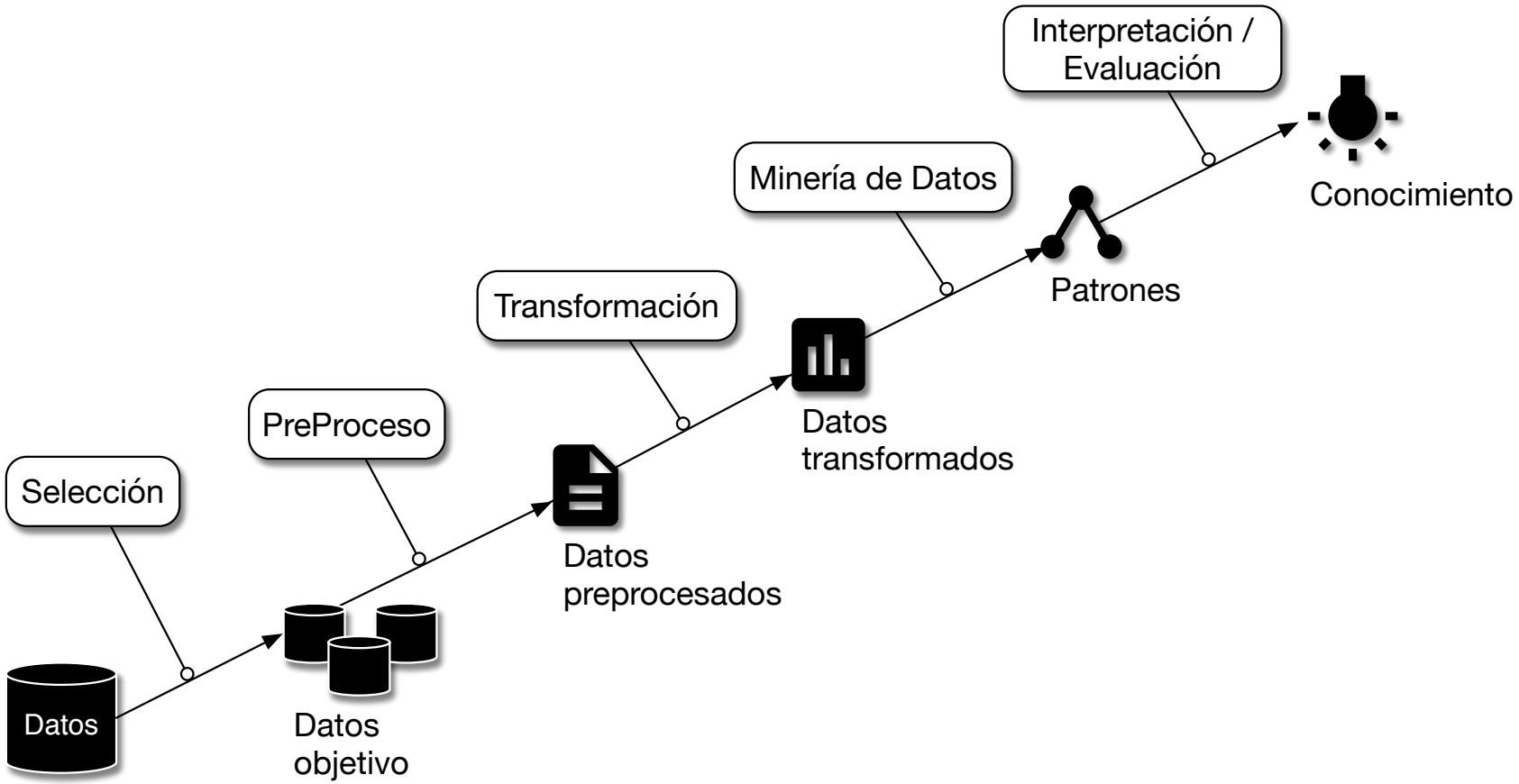
Dr. Alejandro Guerra-Hernández

Universidad Veracruzana
Centro de Investigación en Inteligencia Artificial
Sebastián Camacho No. 5, Centro
Xalapa, Ver., México 91000
mailto:aguerra@uv.mx
http://www.uv.mx/personal/aguerra

CITII 2016, Apizaco, Tlax., noviembre 17, 2016

Universidad Veracruzana

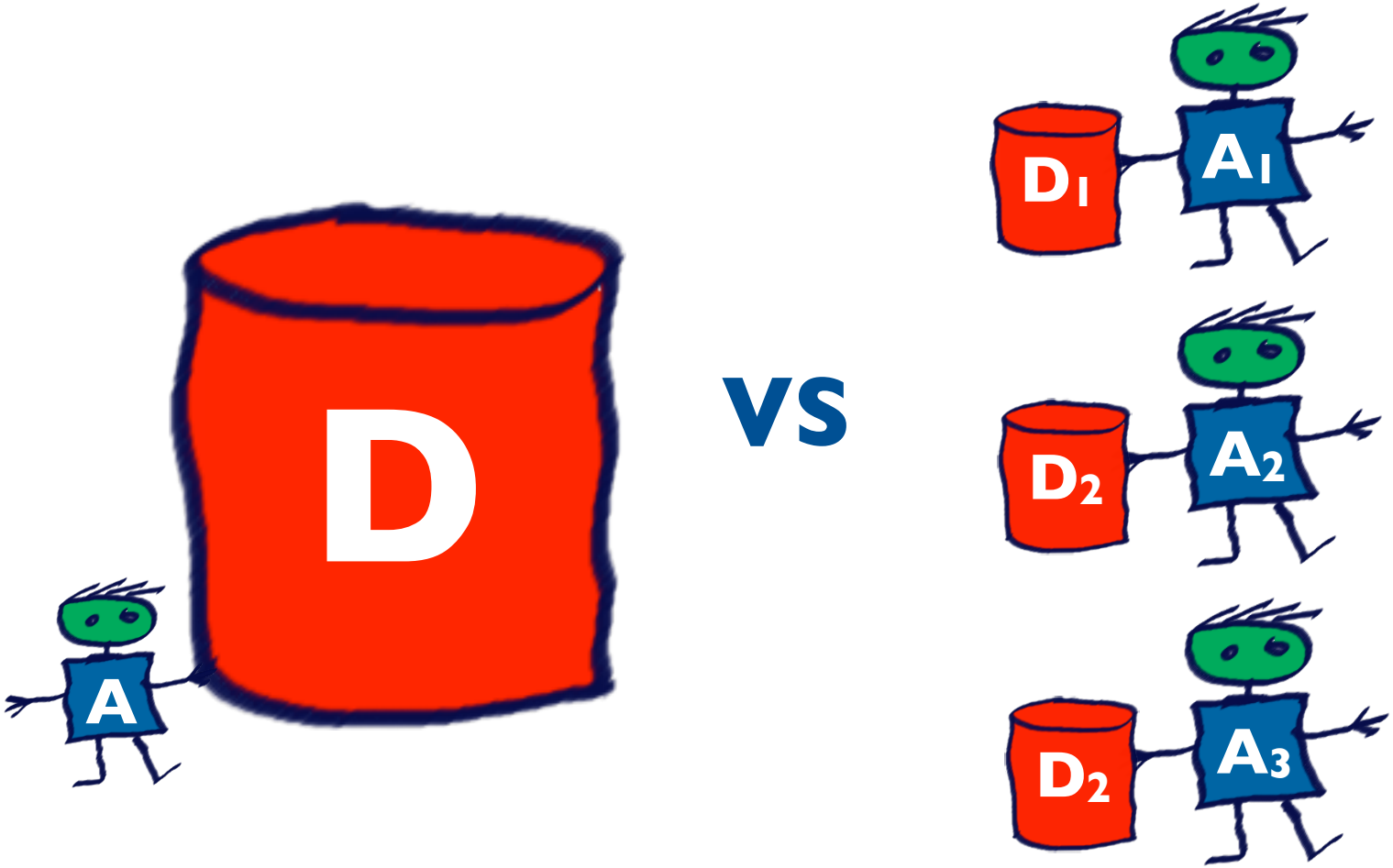
De los datos al conocimiento[1]



[1] U Fayyad, G Piatetsky-Shapiro y P Smyth, "From data mining to knowledge discovery in databases", *AI Magazine*, vol. 17, n.º 3, págs. 37-54, 1996.

Universidad Veracruzana

¿Y si los datos son demasiados?



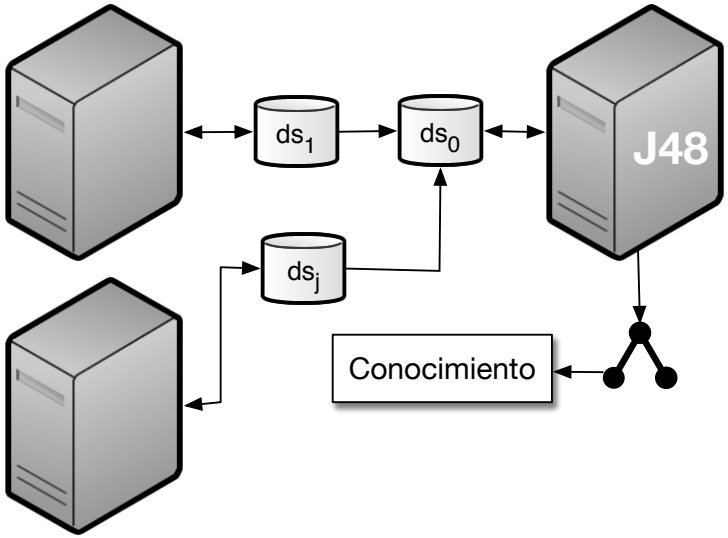
Universidad Veracruzana

¿Y si los datos están distribuidos?

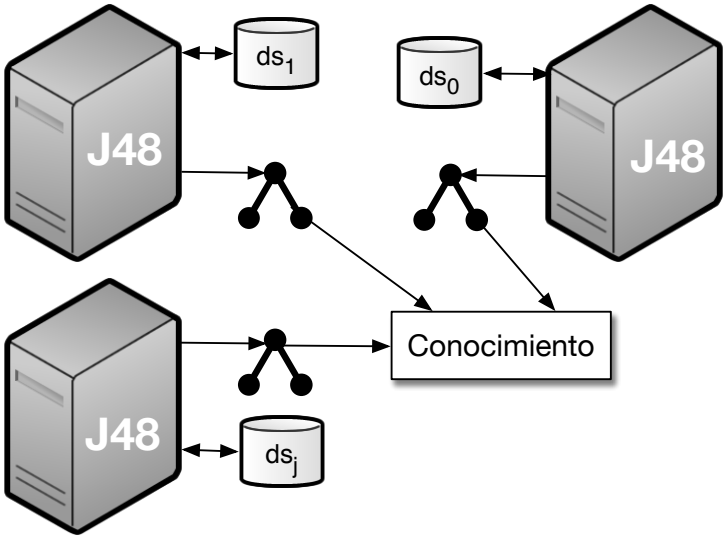
Universidad Veracruzana

Posibles soluciones

Centralización de Datos



Distribución del Aprendizaje



Universidad Veracruzana

Herramientas: Weka[2] + MOA

- Representación uniforme de ejemplos, atributos, modelos, etc.
- Herramientas de evaluación de modelos.
- Soporte: Libro, documentación, comunidad activa, etc.
- Diversos algoritmos, código abierto, para:

Minería de Datos	Transformación	Meta-Aprendizaje
Árboles de decisión	Selección de atributos	Bagging
Reglas	Discretización	Boosting
Modelos lineales	Proyección	Combinación de modelos
Modelos basados en casos	Muestreo	Regresión aditiva
Predicción numérica	Calibración	Stacking
Modelos bayesianos	etc.	etc.
Redes neuronales		
Clustering		
Flujos		

[2] IH Witten, E Frank y MA Hall, *Data mining: Practical machine learning toools and techniques*. Burlington, MA., USA: Morgan Kaufmann Publishers, 2011.

Posibles problemas

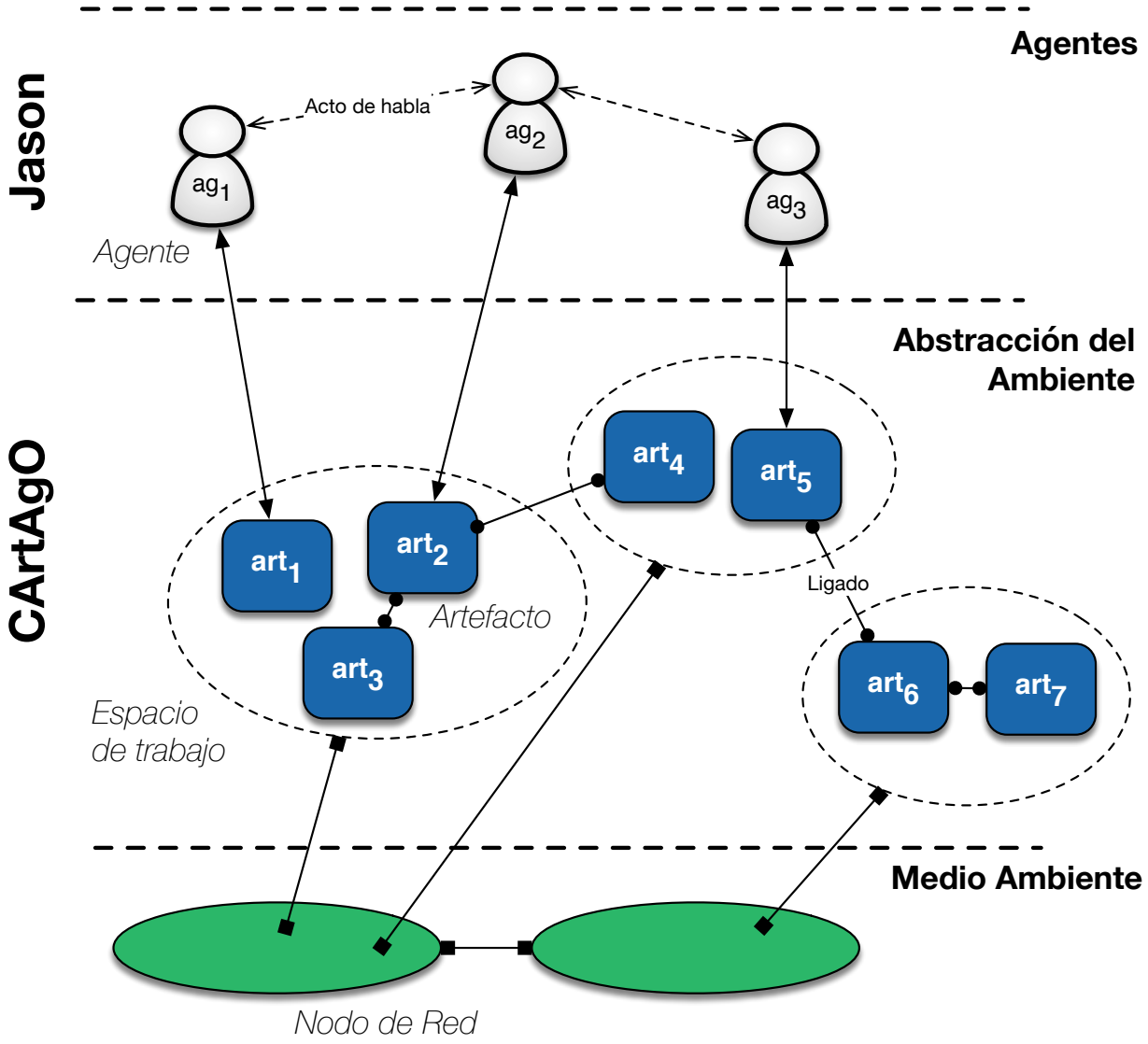
- Distribución de los datos.
- Algoritmos centralizados.
- Datos heterogéneos.
- Comunicación.
- Privacidad.
- Cómputo concurrente y/o distribuido.
- Datos cambiantes y/o crecientes.
- Escalabilidad.

Soluciones basadas en agentes

- Frameworks y Sistemas:
  - JAM [15], BODHI [8], Papyrus [3], i-Analyst [11], EMADS [2], SMAJL [17], GLS [18], etc.
- Centralizantes.
- Meta-aprendizaje.
  - Votación.
  - Arbitraje.
  - Combinación.
- Problemas existentes:
  - Modelos de agencia débil.
  - Extensibilidad.
  - Generalidad.
- Solución: JaCa-DDM[3]

[3] X Limón, A Guerra-Hernández, N Cruz-Ramírez y col., "An agents & artifacts approach to distributed data mining", en *MICAI 2013: Eleventh Mexican International Conference on Artificial Intelligence*, F Castro, A Gelbukh y MG Mendoza, eds., ép. Lecture Notes in Artificial Intelligence, vol. 8266, Berlin Heidelberg: Springer Verlag, 2013, págs. 338-349.

Agentes & Artefactos (JaCa)



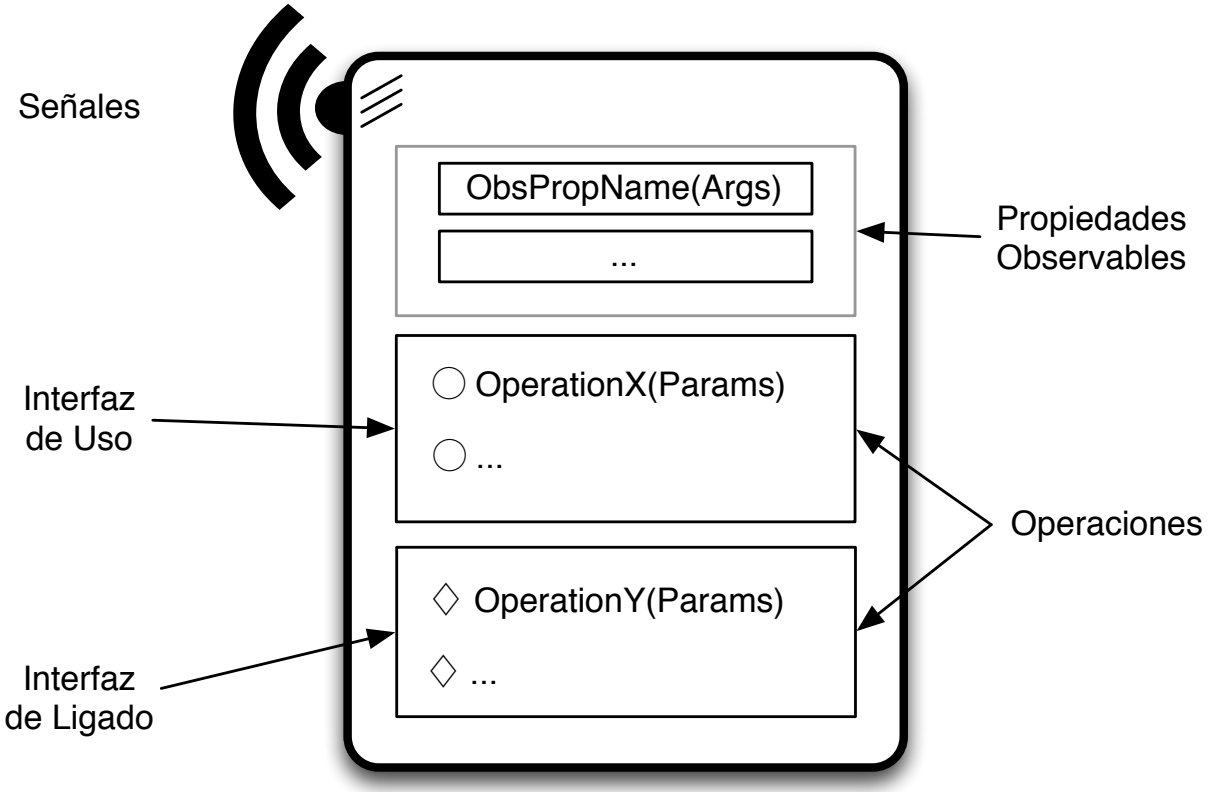
Universidad Veracruzana

Jason[4]

- Lenguaje de Programación Orientado a Agentes.
- *AgentSpeak*(L) + Anotaciones.
- BDI: Creencias, Metas, Intenciones, Planes, Eventos.
- Semántica operacional bien definida.
- Comunicación basada en KQML: Actos de habla.
- Concurrente.
- *Modular*.
- Facilidades para programar ambientes.
- Abierto, documentado, implementado en Java.

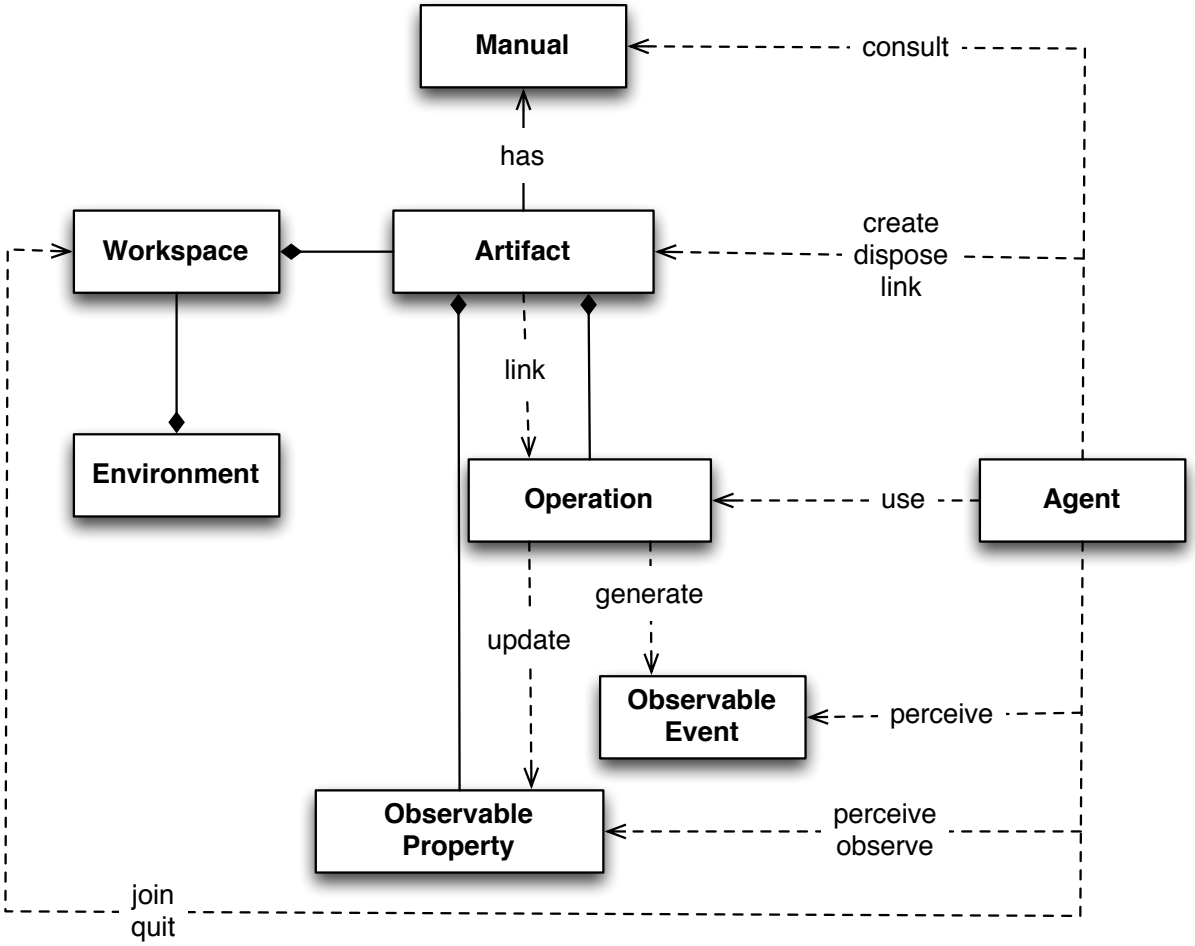
[4] RH Bordini, JF Hübner y M Wooldridge, *Programming multi-agent systems in agent-speak using jason*. John Wiley & Sons Ltd, 2007.

Artefacto



Universidad Veracruzana

Meta-modelo de Agentes y Artefactos[5]



[5] A Ricci, M Piunti y M Viroli, "Environment programming in multi-agent systems: An artifact-based perspective", *Autonomous Agents and Multi-Agent Systems*, vol. 23, n.º 2, págs. 158-192, 2011.

El Modelo JaCa-DDM

- Se ha definido un modelo abstracto JaCa-DMM, a partir de dos constructores básicos:

  Estrategia JaCa-DDM. Flujo de trabajo basado en agentes Jason y artefactos CArtAgO, basados en Weka [16] y MOA [4].

  Sistema de Despliegue JaCa-DDM. La manera en que los componentes del sistema se ubican en la arquitectura de cómputo distribuida donde se ejecutará la estrategia. Basada en nodos CArtAgO.

Estrategia JaCa-DDM

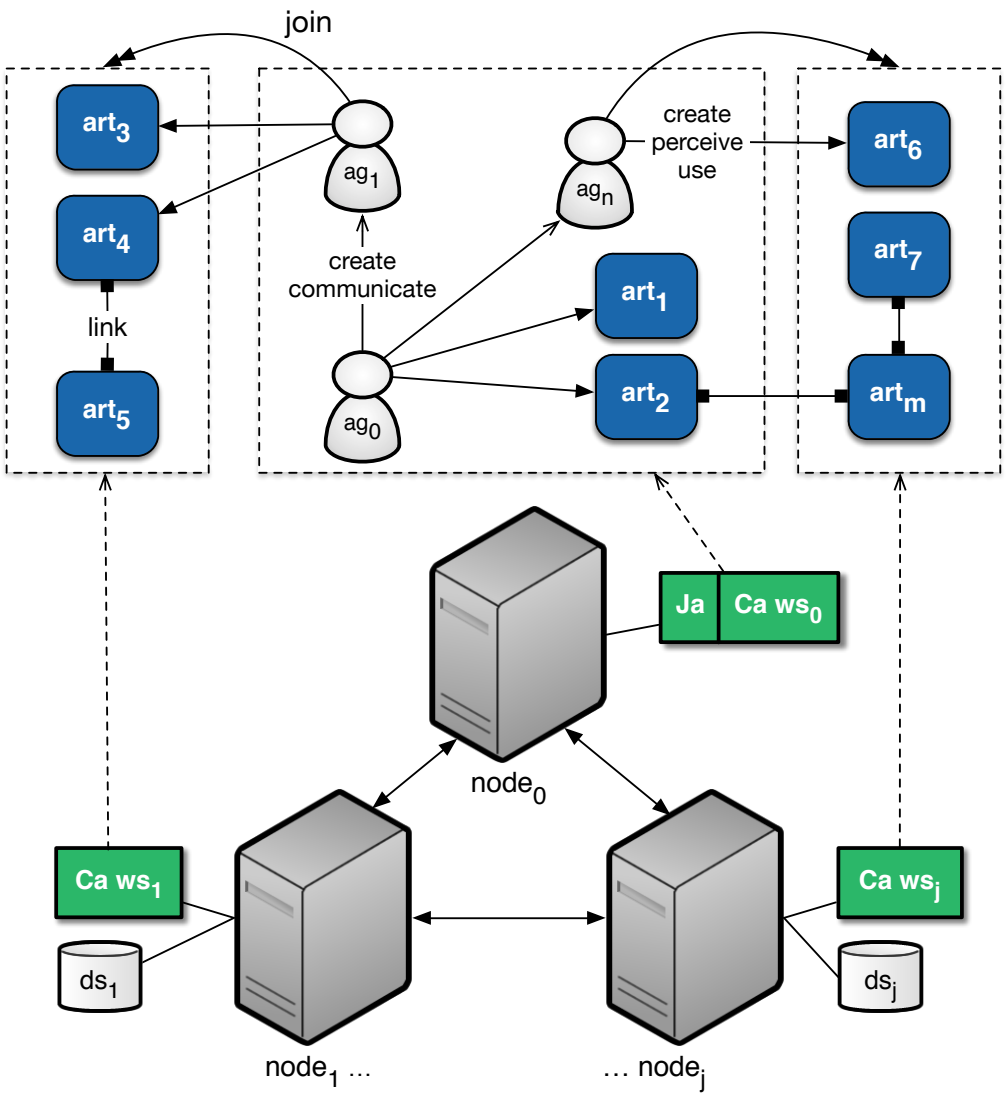
- Se define como $\langle Ags, Arts, Params, ag_1 \rangle$, donde:
  - $Ags = \{ag_1, \ldots, ag_n\}$ es el conjunto de programas de agente.
  - $Arts = \{art_1, \ldots, art_m\}$ es el conjunto de tipos de artefacto.
  - $Params = \{param_1 : tipo_1, \ldots param_k : tipo_k\}$ con los tipos de datos $\in \{int, bool, double, string\}$.
  - $ag_1 \in Ags$ el agente responsable de la estrategia.
- Las interacciones entre agentes se detallan usando diagramas de secuencia UML (o AUML).

Sistema de Despliegue JaCa-DDM

- Se define como $\langle Nodos, DS, Arts, Estrat, Config, ag_0 \rangle$, donde:
  - $Nodos = \{nodo_0, nodo_1, \ldots, nodo_j\}$ son los nodos CArtAgO del sistema. Cada $nodo_i = \langle nodo_i, IPAdr : Port \rangle$.
  - $DS = \{ds_1, \ldots, ds_j\}$ son las fuentes de datos para cada nodo, exceptuando el $nodo_0$.
  - $Arts = \{art_1, \ldots, art_i\}$ son los tipos de artefactos primitivos usados en el sistema de despliegue.
  - $Estrat$ es una estrategia JaCa-DDM.
  - $Config = \langle \delta, \pi \rangle$ denota la configuración, donde:
    - $\delta = \langle (ag, nodo, i), \ldots \rangle$ **son asignaciones de** i **copias de un programa de agente** ag **focalizando en cierto** $nodo$ $(ag\#i)$.
    - $\pi = \{(param : val), \ldots\}$ **son pares parámetro-valor, conforme a** $Estrat$.
- Se genera vía la interfaz de JaCa-DDM (formato XML).

Universidad Veracruzana

Arquitectura JaCa-DDM



Universidad Veracruzana

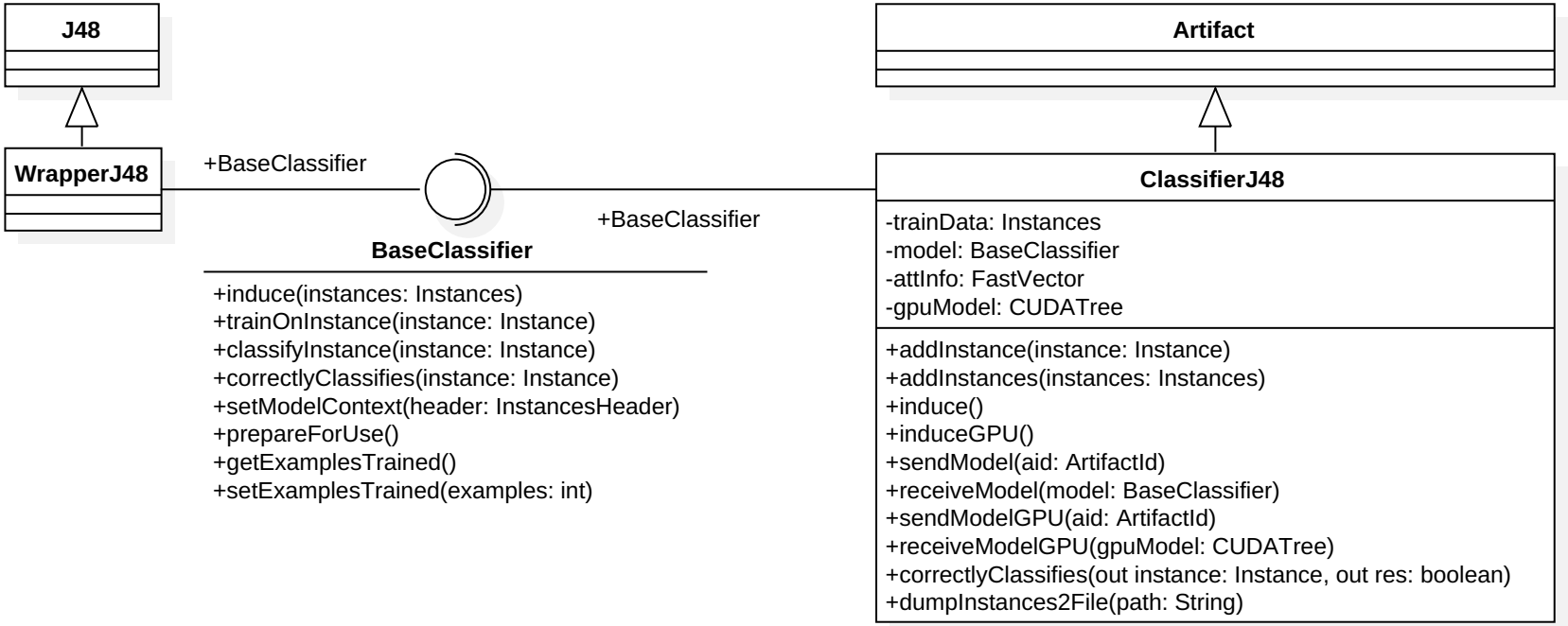
Tipos de artefactos primitivos (1)

Tipo de Artefacto	Descripción
`Classifier`*XXX*	Una herramienta de clasificación, capaz de aprender nuevos modelos y usarlos para clasificar ejemplos, basada en Weka/MOA. *XXX* puede substituirse por alguno de los siguientes algoritmos: J48, VFDT, Bagging, BaggingEnsemble.
`Directory`	Una herramienta de localización de servicios para agentes, artefactos y espacios de trabajo (al estilo páginas amarillas y blancas).
`Evaluator`	Una herramienta para la evaluación de modelos basada en Weka.
`GUI`	Una intefaz gráfica para configurar y lanzar experimentos.
`FileManager`	Una herramienta para escribir archivos ARFF, basado en Weka.
`InstancesBase`	Un repositorio de ejemplos de aprendizaje basado en Weka.
`LogBook`	Una herramienta para generar reportes de resultados de los experimentos.

Tipos de artefactos primitivos (2)

Tipo de Artefacto	Descripción
`Oracle`	Una herramienta para distribuir conjuntos de entramiento centralizados en los diferentes nodos de la arquitectura, basado en Weka.
`TrafficMonitor`	Un sniffer para medir la carga de la red de la arquitectura, basado en utilidades de terceros. Su uso es opcional.
`Utils`	Una navaja suiza para los agentes, con multiples utilidades.

Detalles (ClassifierJ48)



Agentes obligatorios en JaCa-DDM

Responsable de Sistema de Despliegue. Es el encargado de configurar la arquitectura y evaluar los resultados obtenidos (ag_0).

Responsable de estrategia. Es el encargado de iniciar y terminar una estrategia determinada (ag_1).

ag_0 responsable del Sistema de Despliegue (1)

Configurar el experimento. Interactiva, conforme a la estrategia adoptada, gracias a un artefacto de tipo `GUI`, incluye la distribución de los agentes (δ) y la inicialización de parámetros (π).

Distribuir datos dinámicamente. Cuando JaCa-DDM se usa para simular escenarios distriuidos, ag_0 puede crear un artefacto de tipo `Oracle` para distribuir un conjunto de entrenamiento centralizado, en los diferentes nodos de la arquitectura.

Monitoreo de tráfico. Es posible evaluar el tráfico de datos que genera la ejecución de un experimento opcionalmente, usando un artefacto de tipo `TrafficMonitor`.

ag_0 responsable del Sistema de Despliegue (2)

Despliegue de agentes. La creación de *Ags* de la estrategia adoptada. La definición del sistema de despliegue le indica cuantos agentes de cada tipo debe crear en los nodos de la arquitectura. La inicialización de estos agentes consiste en:

- `node(NodeName).`
- `ipNode0(IPAdress:Port).`
- `data(FilePath).`
- `param(ParamId,Val).`
- Planes primitivos.

Universidad Veracruzana

ag_0 responsable del Sistema de Despliegue (3)

Evaluar modelos. A través de artefactos de tipo `Evaluator` y `LogBook`, ag_0 puede evaluar los modelos aprendidos y generar reportes sobre su desempeño. Un mensaje $\langle ag_1, tell, finish(ArtId)\rangle$, donde ag_1 es el agente responsable de la estrategia adoptada y *ArtId* es el artefacto que almacena el modelo obtenido, le avisa que puede proceder a la evaluación.

Limpieza. Si la evaluación de un experimento implica repeticiones iteradas del mismo, como en el caso de la validación cruzada, el ag_0 reinicia agentes y artefactos de acuerdo a los requerimientos del método de evaluación.

Universidad Veracruzana

ag_1 responsable de la estrategia

Iniciar el proceso de aprendizaje. Un mensaje $\langle ag_0, achieve, start \rangle$, le avisa a ag_1 que debe lanzar el proceso de aprendizaje.

Finalizar el proceso de aprendizaje. Una vez finalizado el proceso de aprendizaje, ag_1 debe enviar un mensaje $\langle ag_0, tell, finish(ArtId) \rangle$, avisando al responsable del despliegue ag_0 que el proceso ha finalizado y que el modelo obtenido se encuentra en el artefacto $ArtId$.

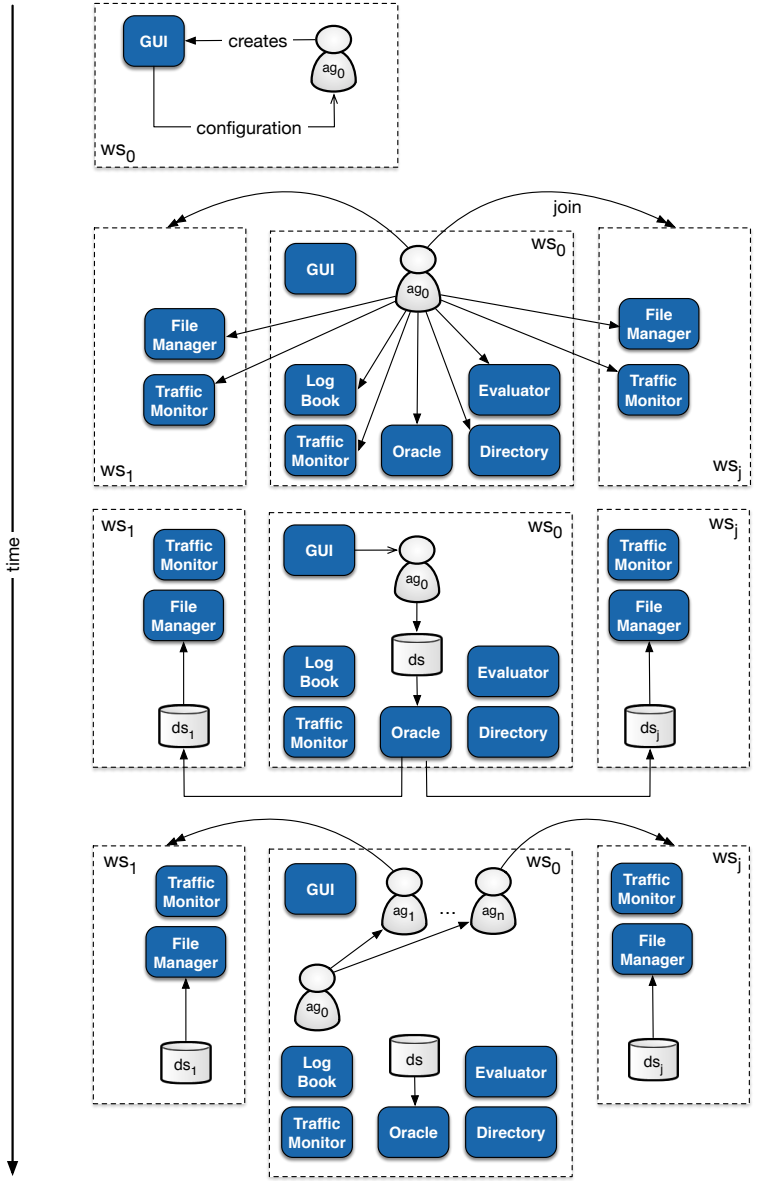
Configuración (1)

- La estrategia *Estrat* adoptada. Al definirse una estrategia, debe generarse un descriptor XML. Ejemplos de estos descriptores pueden encontrarse en los protocolos incluídos en la distribución en el directorio `sampleProtocols`.
- La dirección IP y los puertos usados por los nodos en *Nodos*.
- La distribución de agentes (δ) y la inicialización de parámetros (π).

Configuración (2)

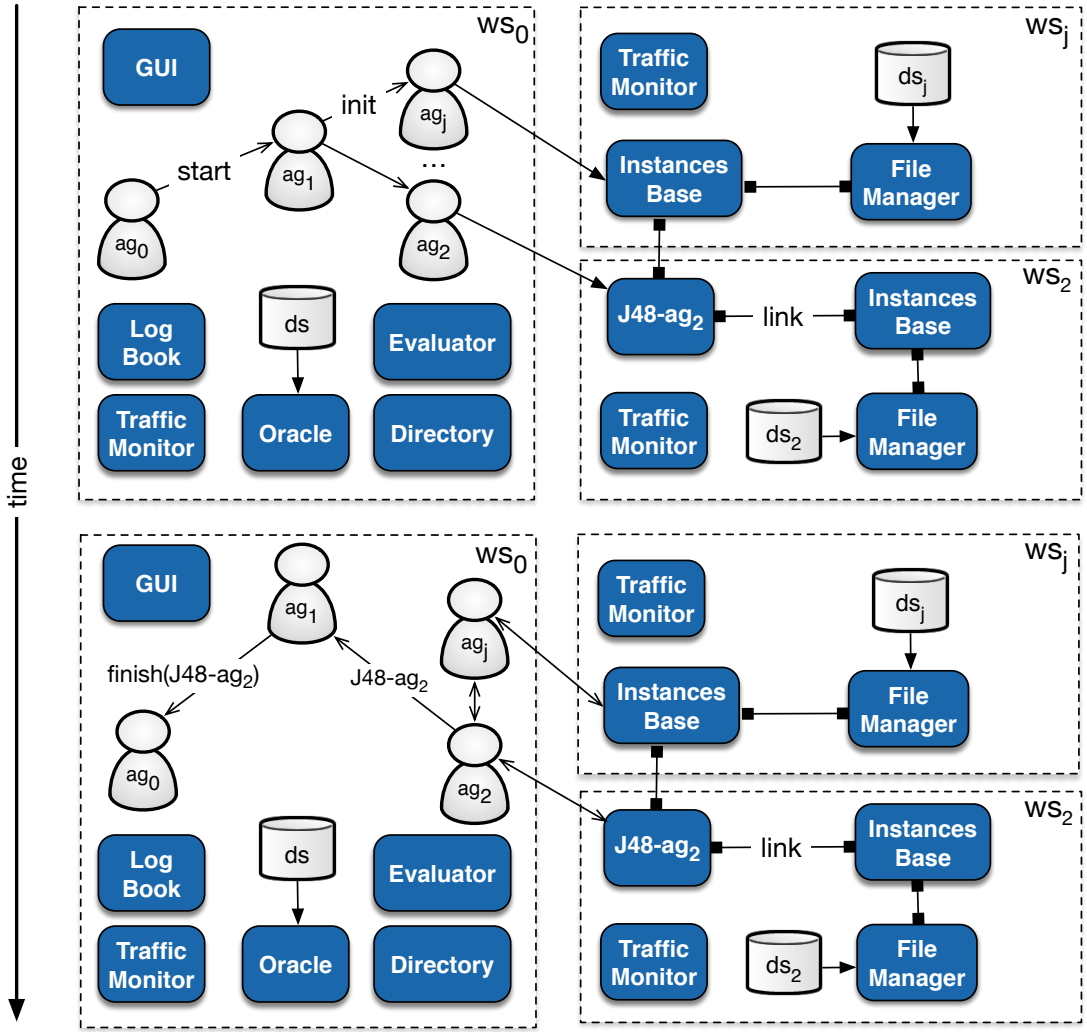
- Distribución dinámica de datos. Un conjunto de entrenamiento se encuentra en el $nodo_0$ y es distribuido en los demás nodos de la arquitectura, usando un artefacto de tipo `Oracle`:
  - `hold-out`. Conforme un parámetro que define el tamaño del conjunto de prueba, en términos de un porcentaje de ejemplos a usarse con este propósito. El resto de los ejemplos disponibles se distribuye de manera estratificada entre los nodos definidos.
  - `cross-validation`. Un parámetro indicando el número de pliegues, determina el radio de ejemplos usados para la prueba y los entrenamientos. Las particiones de entrenamiento se distribuyen en los nodos definidos en el sistema.
- Distribución estática.

Flujo de trabajo (1)



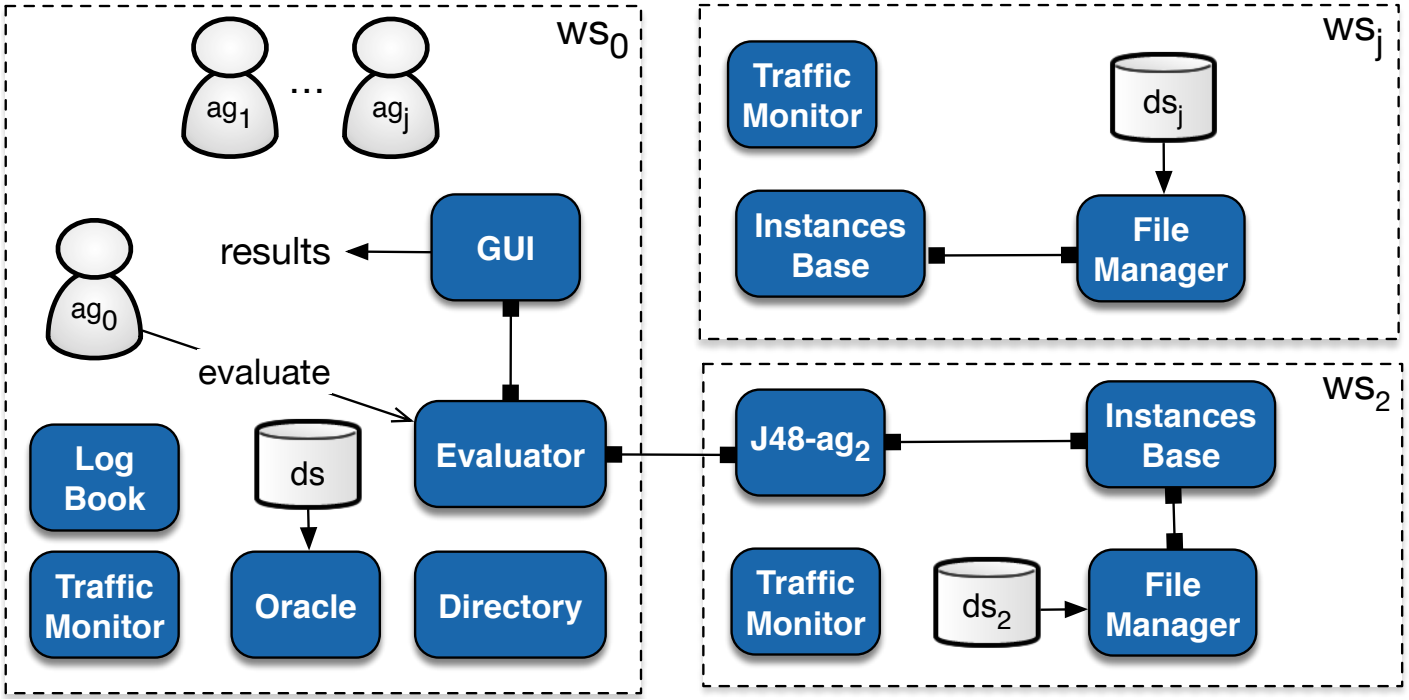
Universidad Veracruzana

Flujo de trabajo (3–4)



Universidad Veracruzana

Flujo de trabajo (5)



Universidad Veracruzana

Estrategias incluídas en JaCa-DDM

- Centralizadas:
  - Centralized
  - Centralized Bagging
- Centralizantes:
  - Centralizing
  - Round
- Basadas en Meta-Aprendizaje:
  - Distributed Bagging
  - Random Forest
- Basadas en Windowing:
  - Counter
  - Round Counter
  - Parallel Round Counter
  - Counter GPU
  - Parallel Counter GPU
  - Parallel Counter GPU Extra

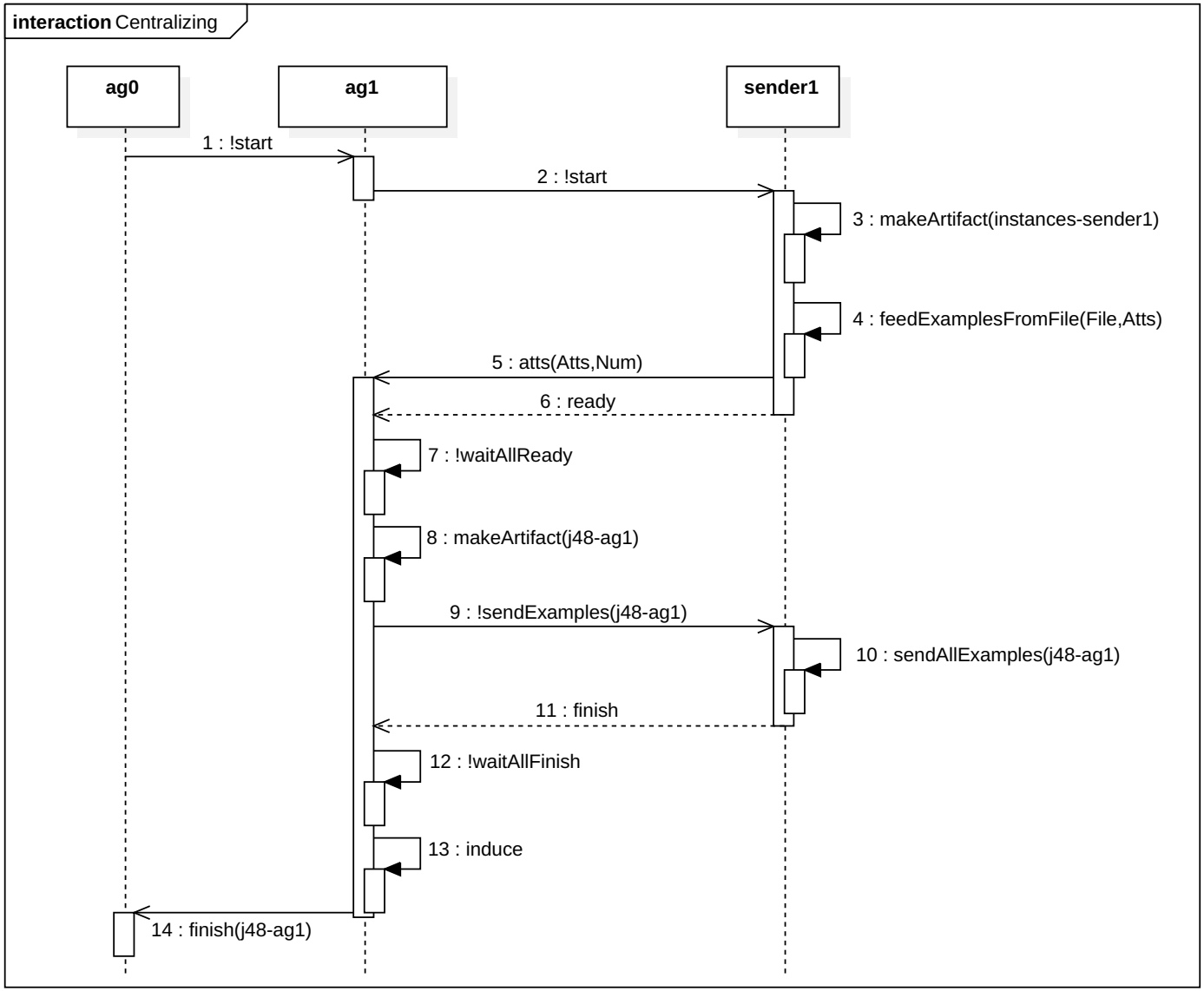
Centralizing (XML)

```xml
<?xml version="1.0"?>
<config xmlns="http://uv.mx/hlimon"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://uv.mx/hlimon sys.xsd">

<name>Centralizing</name>
<agents>
  <agent>
    <program>contactPerson</program>
    <file>../sampleProtocols/centralizing/contactPerson.asl</file>
  </agent>
   <agent>
    <program>sender</program>
    <file>../sampleProtocols/centralizing/sender.asl</file>
  </agent>
</agents>
<params>
  <param>
    <name>Pruning</name>
    <type>boolean</type>
  </param>
    <param>
    <name>Classifier</name>
    <type>string</type>
  </param>
</params>
</config>
```

Universidad Veracruzana

Centralizing (Diagrama de secuencia UML)



Universidad Veracruzana

contactPerson I

```
+begin_process: true <-
    !consult_type_count("sender", NumSenders);
    +num_senders(NumSenders);
    for (.range(X,1,NumSenders))
    {
      .concat("sender",X,SenderX);
      .send(SenderX, achieve, start);
    }.

+atts(N,A)[source(sender1)] : true <-
    !waitAllReady;
    ?atts(N,A);
    ?param("Pruning", ParamVal);
    ?param("Classifier", Classifier);
    if(Classifier == "VFDT")
    {
        makeArtifact("classifier-contactPerson1","artifacts.VFDT",[b(B, a(N, A))],
              IdJ48);
    }
    else
    {
        makeArtifact("classifier-contactPerson1","artifacts.ClassifierJ48",[b(B, a(N,
              A)), ParamVal], IdJ48);
    };
    !register_artifact("classifier-contactPerson1");
    ?num_senders(NumSenders);
    for (.range(X,1,NumSenders))
```

Universidad Veracruzana

contactPerson II

```
    {
      .concat("sender",X,SenderX);
      .send(SenderX, achieve, sendExamples("classifier-contactPerson1"));
    };
    !waitAllFinish;
    induce;
    .send(experimenter, tell, finish("classifier-contactPerson1")).

+ready(Name)[source(Ag)]: true
  <-
  +agent(Name). //to know the name of the agents

+!waitAllReady: num_senders(NumSenders) & .count(ready(Name),E) & E == NumSenders
  <-
  println("All agents are ready").

+!waitAllReady: true
  <- .wait(50);
  !waitAllReady.

+!waitAllFinish: num_senders(NumSenders) & .count(finish(Name),E) & E == NumSenders
  <-
  println("All agents have finished").

+!waitAllFinish: true
  <- .wait(50);
  !waitAllFinish.
```

sender I

```
+!start[source(contactPerson1)]: true <-
    .my_name(Name);
    .concat("instances-", Name, ArtName);
    makeArtifact(ArtName,"artifacts.InstancesBase",[],ArtIdInstances);
    ?data(Path);
    feedExamplesFromFile(Path, Atts)[artifact_id(ArtIdInstances)];
    .term2string(AttsStr, Atts);
    -+AttsStr;
    ?atts(N, A);
    if(Name == sender1)
    {
      .send(contactPerson1, tell, atts(N, A));
    };
    .send(contactPerson1, tell, ready).

+!sendExamples(ArtName): true <-
    ?ipServer(IPSer); //all agents have this believe
    joinRemoteWorkspace("default",IPSer,_); // join node0
    getArtifactId(ArtName, IdArt); // from directory
    quitWorkspace; // return to previous workspace
    sendAllExamples(IdArt);
    .send(ag1, tell, finish).
```

Descargar JaCa-DDM

- Descarga:

  `https://sourceforge.net/projects/jacaddm/files/latest/download`

- Distribución incluye:

  node0. El $nodo_0$ del modelo JaCa-DDM.

  defaultNode. Prototipo de un nodo distribuido. Los nodos $nodo_1, \ldots, nodo_n$ del modelo JaCa-DDM.

  sampleProtocols. Protocolos definidos en el sistema.

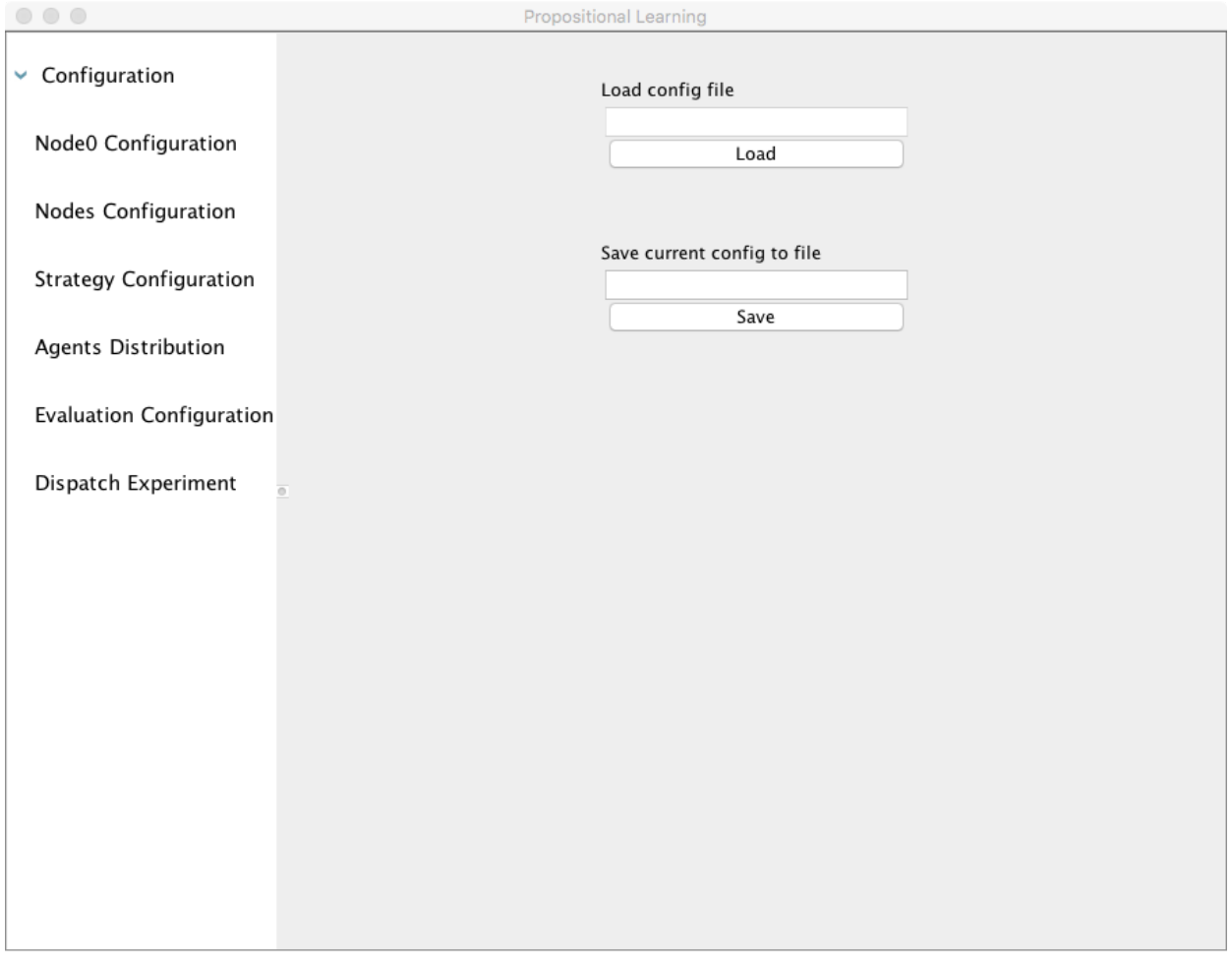
Iniciando los nodos distribuidos

- Copiar el folder `defautNode` a cada computadora a usar.
- Lanzar los nodos:

```
defaultNode> ./run.sh localhost 8080 &
[1] 1046
CArtAgO Node Ready on localhost:8080
defaultNode> ./run.sh localhost 8081 &
[2] 1057
CArtAgO Node Ready on localhost:8081
```

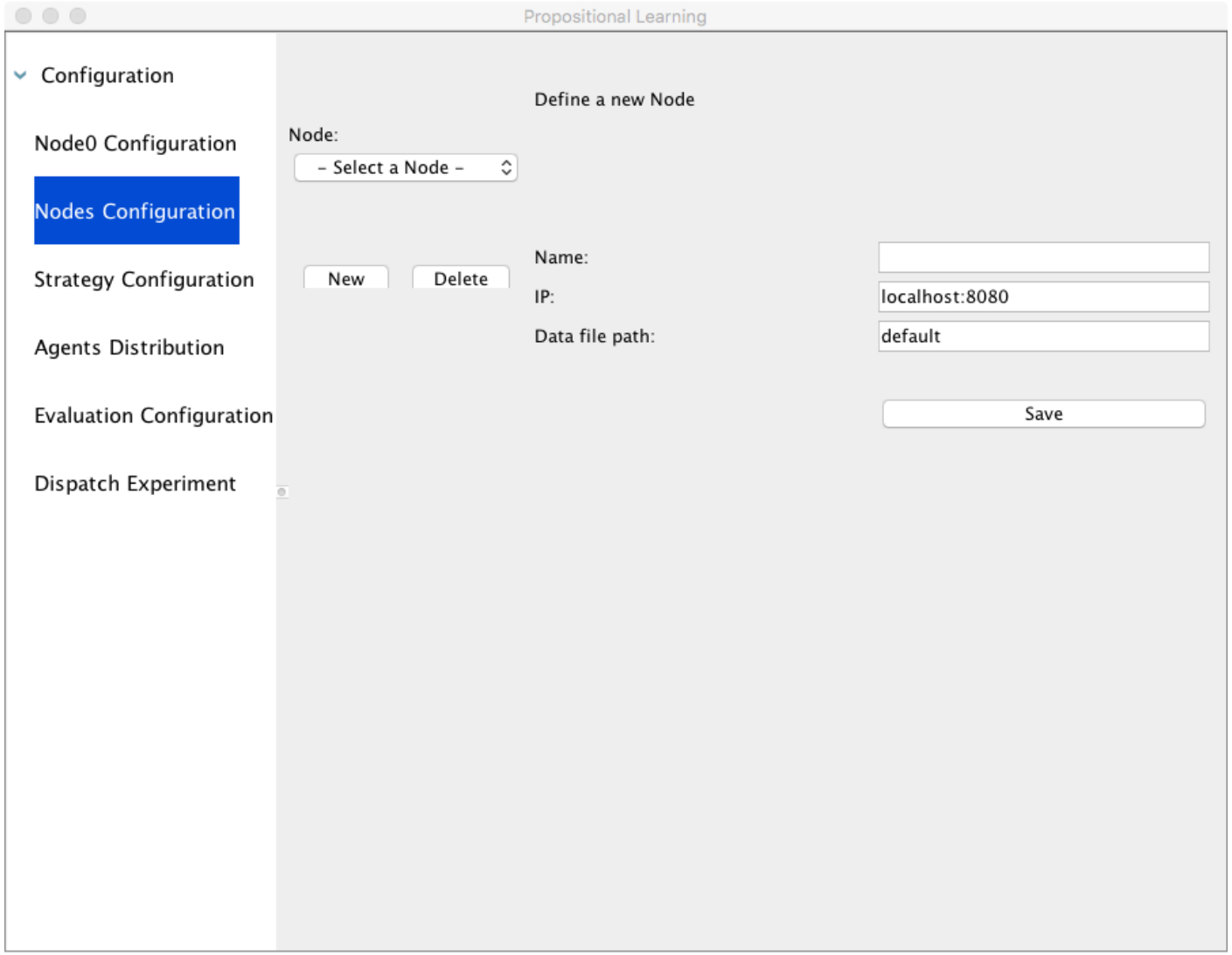
Iniciando JaCa-DDM

- Desde la carpeta `node0`, ejecutar: `./run.sh &`



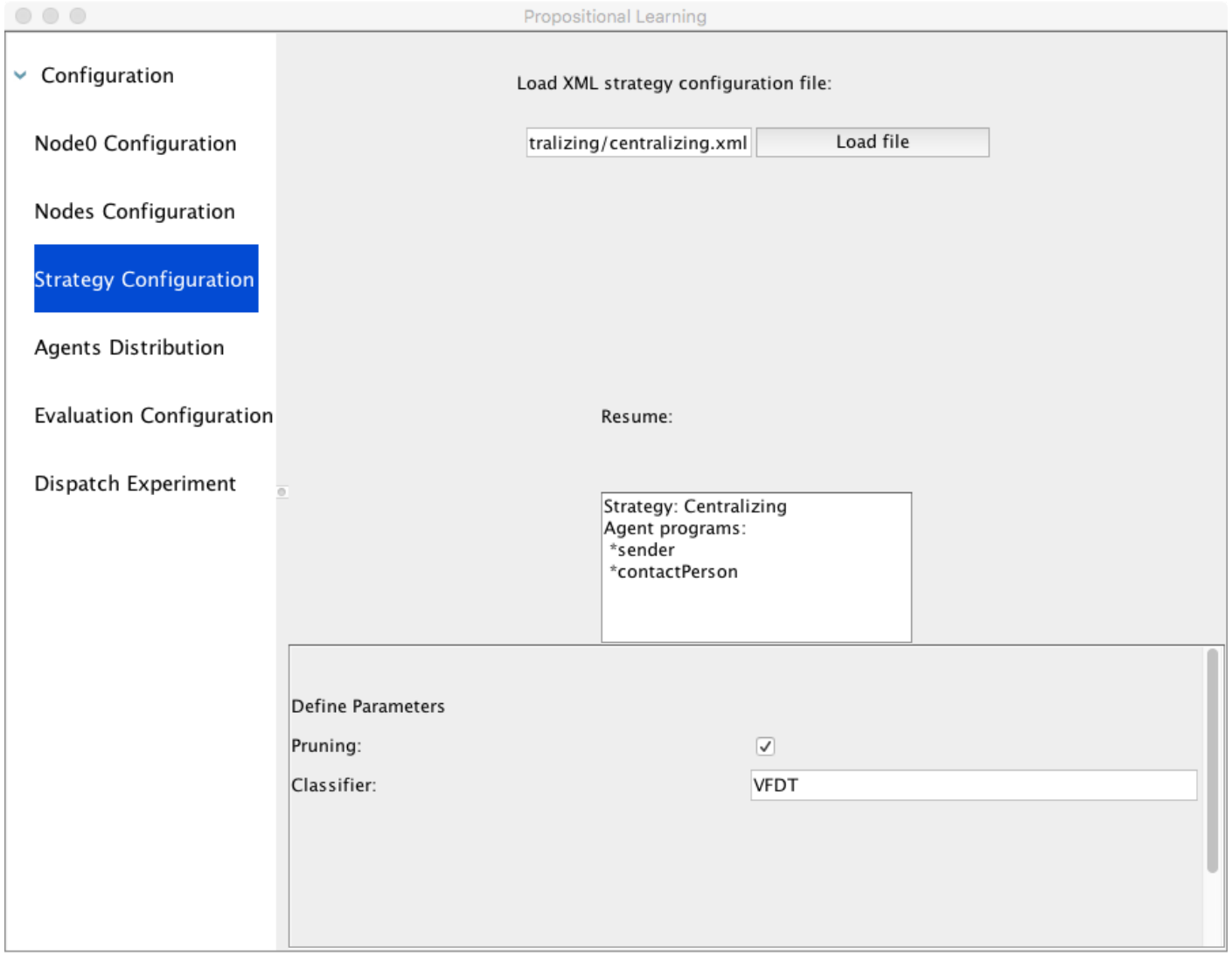
Universidad Veracruzana

Configuración de nodos



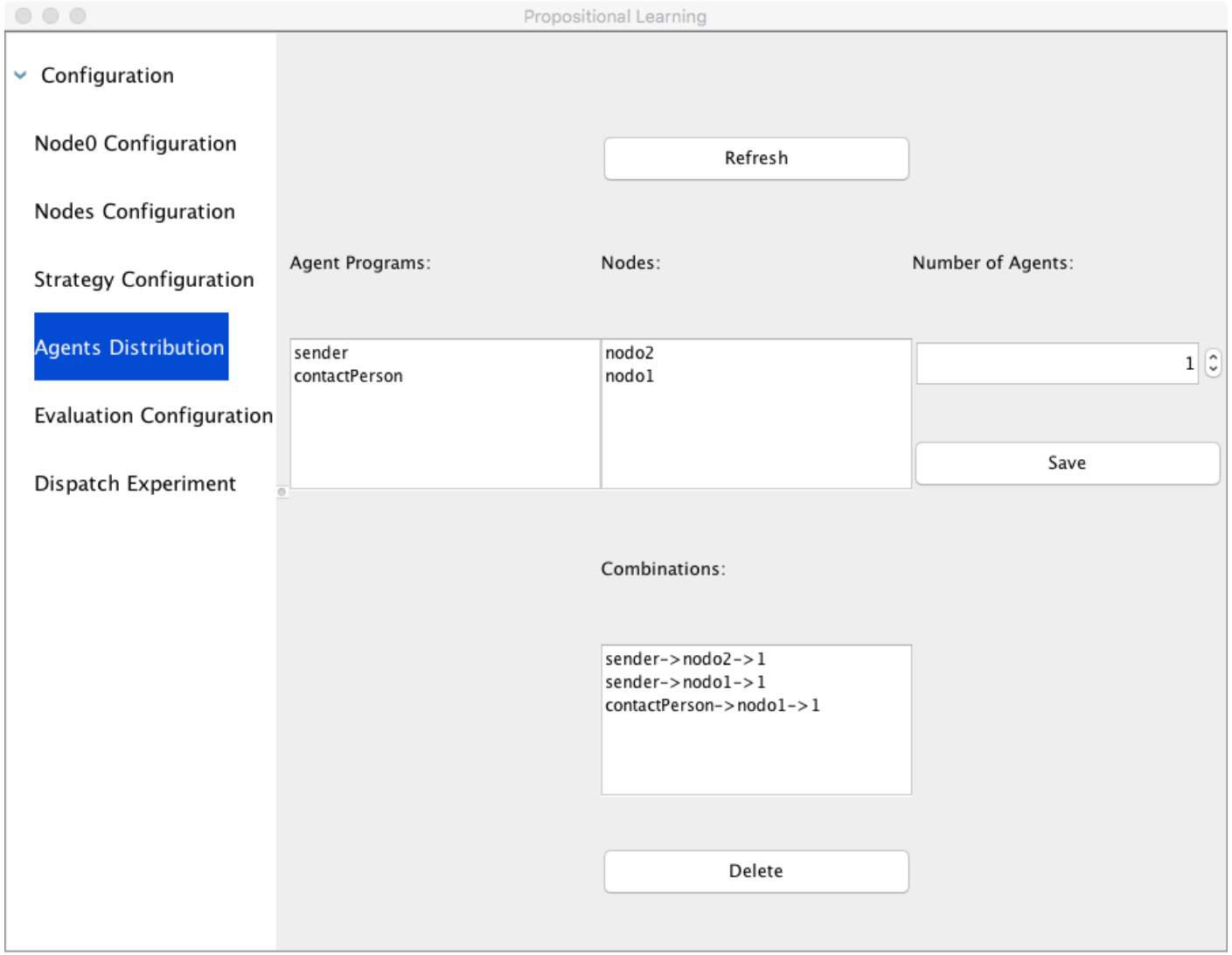
Universidad Veracruzana

Configuración de una estrategia (π)



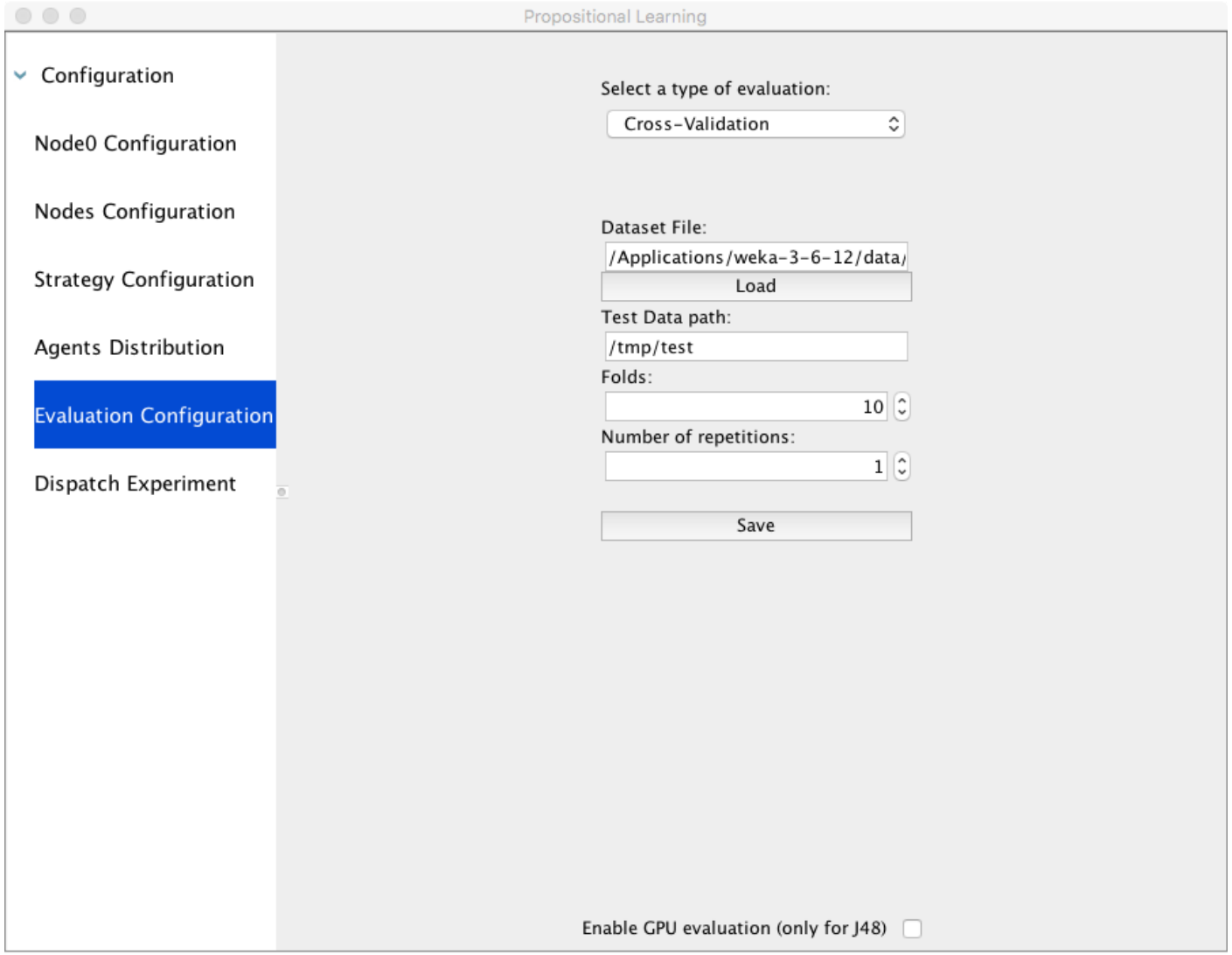
Universidad Veracruzana

Distribución de agentes (δ)



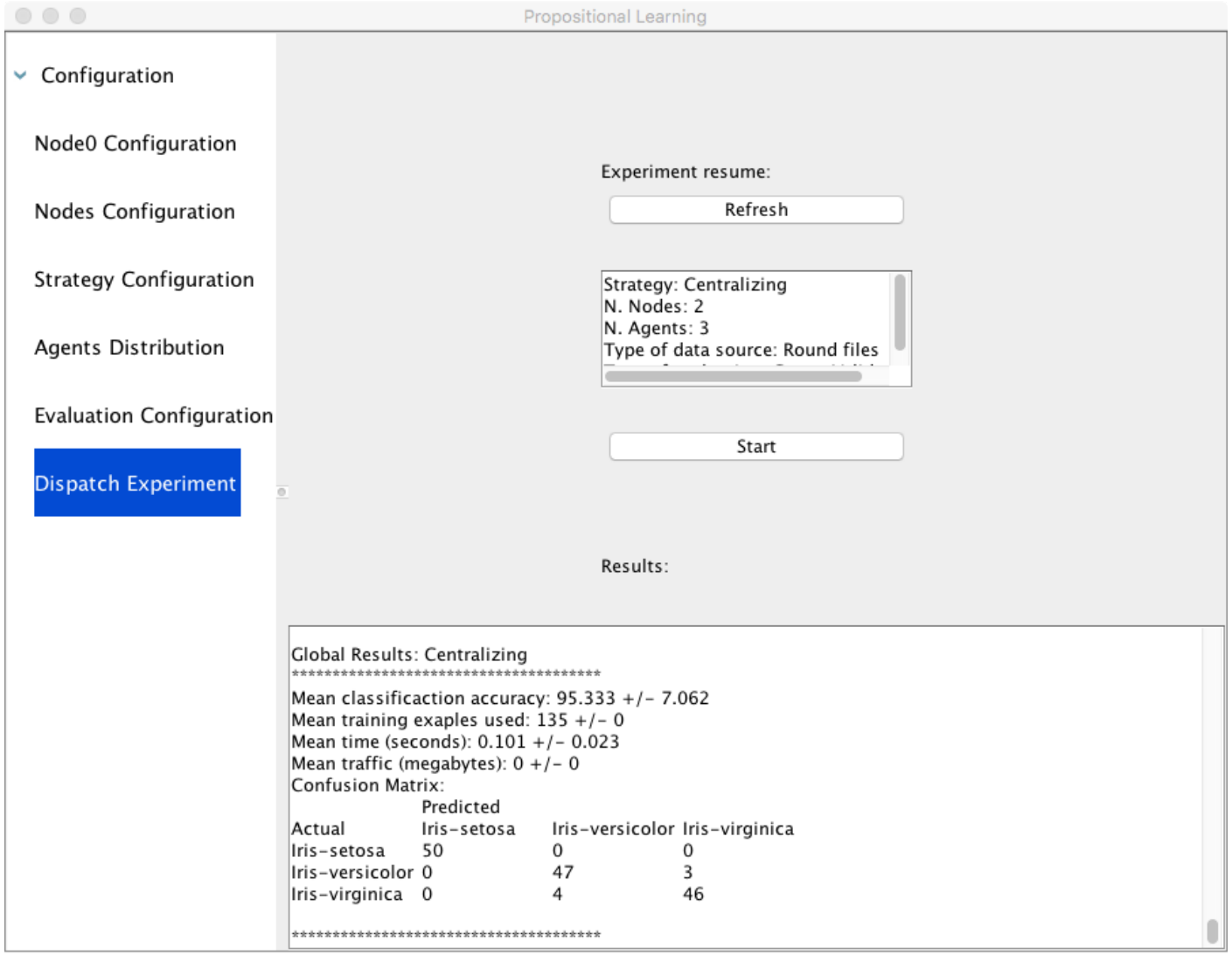
Universidad Veracruzana

Configuración del método de evaluación



Universidad Veracruzana

Ejecución del experimento



Universidad Veracruzana

Windowing[6]

Algoritmo

```
function Windowing(Exs)
    Window ← sample(Exs)
    Exs ← Exs − Window
    repeat
        stopCond ← true
        model ← induce(Window)
        for ex ∈ Exs do
            if classify(model, ex) ≠ class(ex) then
                Window ← Window ∪ {ex}
                Exs ← Exs − {ex}
                stopCond ← false
            end if
        end for
    until stopCond
    return model
end function
```

Ventajas

- Reduce ejemplos.
- Mantiene precisión.

Desventajas

- Costo inducción.
- Costo clasificación.

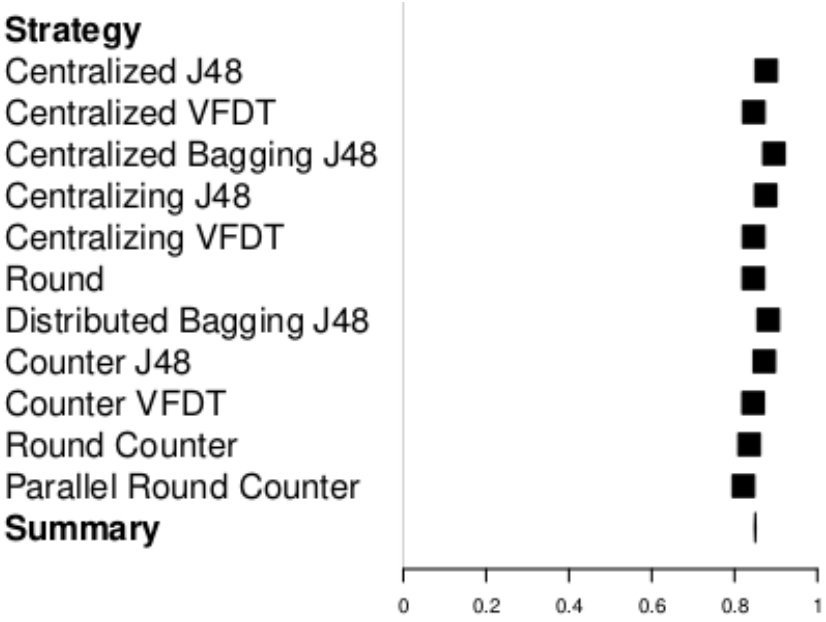
[6] JR Quinlan, *C4. 5: Programs for machine learning*. San Mateo, CA., USA: Morgan kaufmann, 1993, vol. 1.

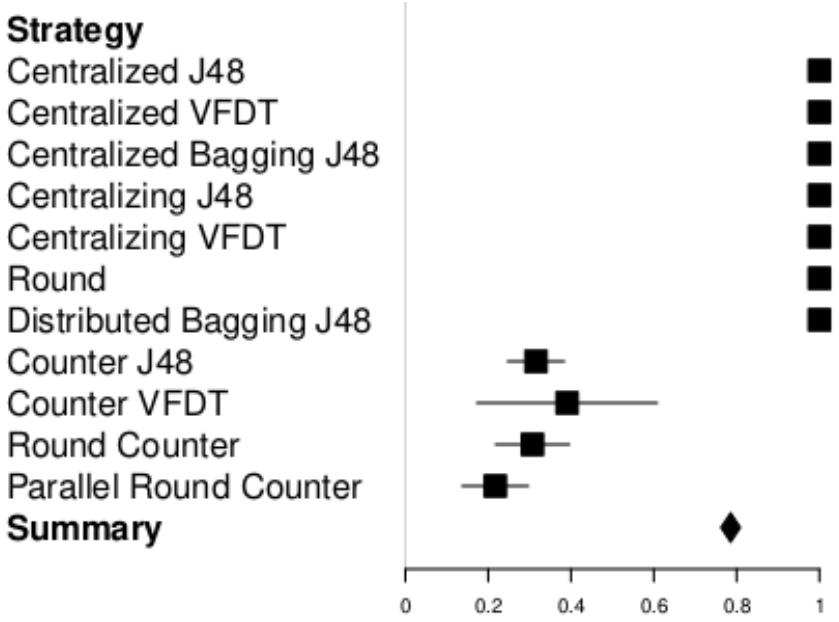
Segundos resultados [7]

Datos	Ejemplos	Atributos	Clases
adult	48842	15	2
australian	690	15	2
breast	683	10	2
credit-g	1000	21	2
covtypeNorm	581012	55	7
diabetes	768	9	2
ecoli	336	8	8
german	1000	21	2
hypothyroid	3772	30	4
imdb-D	120919	1002	2
kr-vs-kp	3196	37	2
letter	20000	17	26
mushroom	8124	23	2
poker	829201	11	10
segment	2310	20	7
sick	3772	30	2
splice	3190	61	3
waveform-5000	5000	41	3

[7] En revisión en KAIS, marzo 2015.

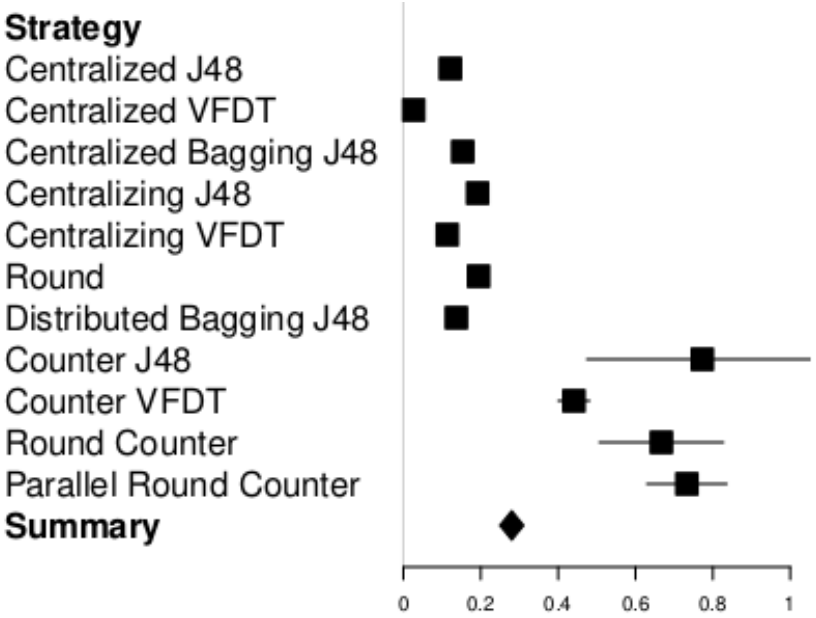
Precisión vs Ejemplos usados (Forest Plots)

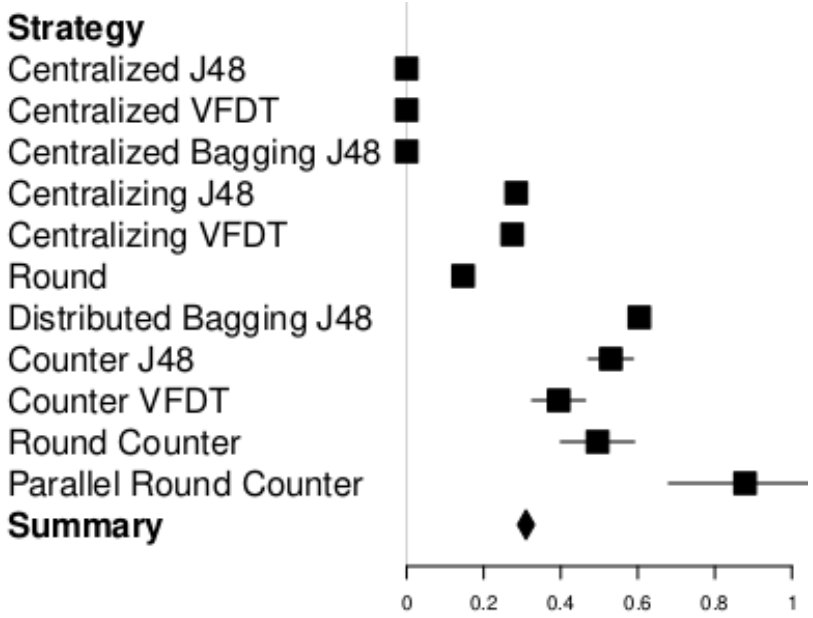




Universidad Veracruzana

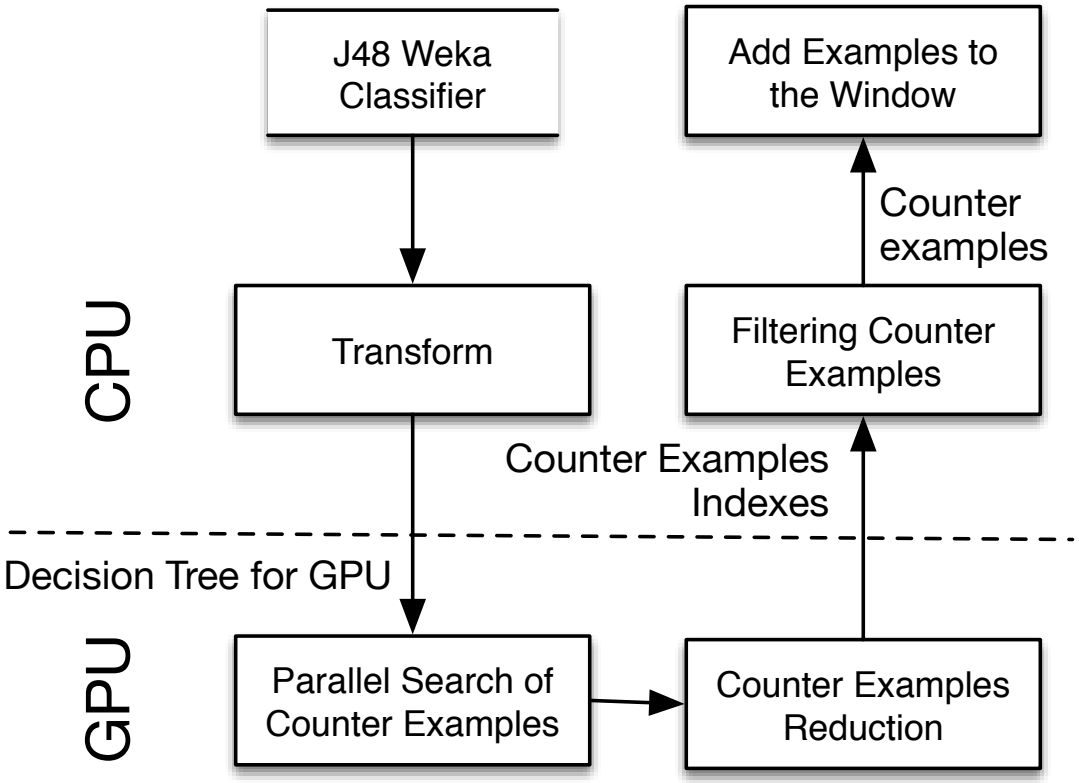
Tiempo vs Tráfico





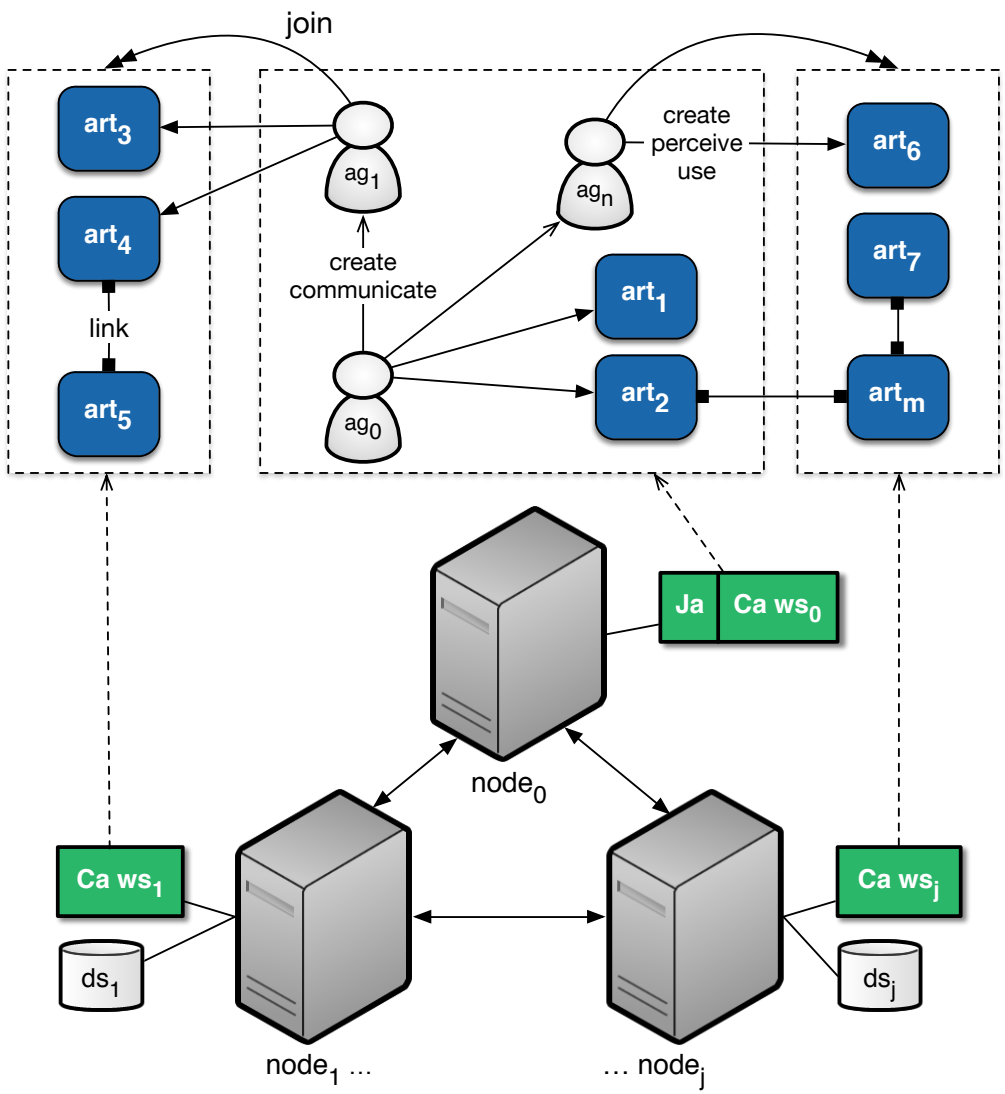
Universidad Veracruzana

Primera revisión: GPU[8]



[8] X Limón, A Guerra-Hernández, N Cruz-Ramírez y col., "A windowing based GPU optimized strategy for the induction of decision trees", en *Arfificial Intelligence Research and Development*, E Armengol, D Boixader y F Grimaldo, eds., ép. Frontiers in Artificial Intelligence and Applications, vol. 277, Amsterdam, NL: IOS Press, 2015, págs. 100-109.

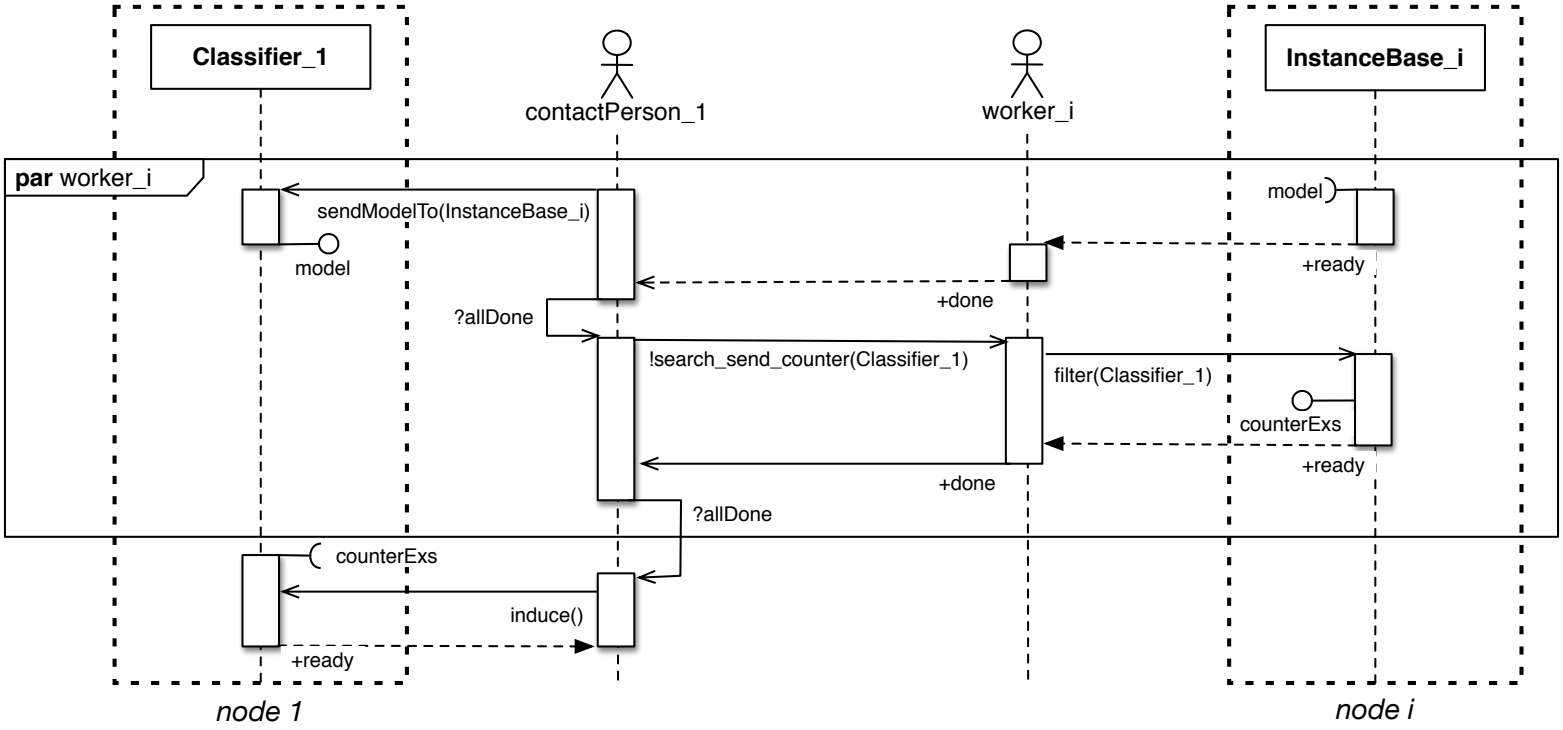
Cluster



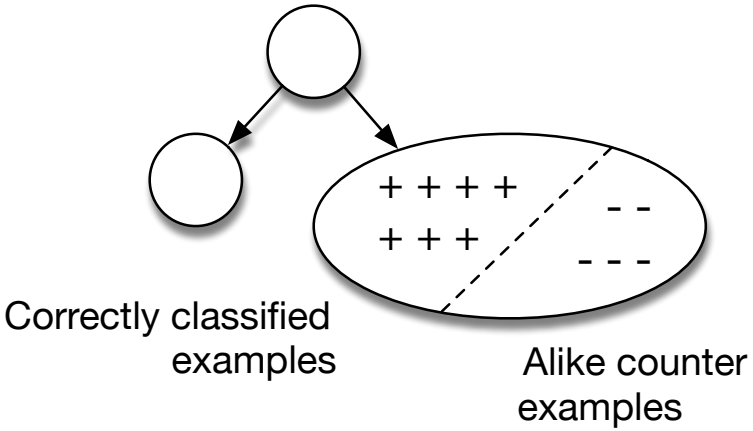
- 2 procesadores Xeon a 2.40 GHz, con 4 cores, dos hilos c/u.
- 24 GB of RAM.
- GPU Tesla C2050 con 448 cores, 6 GB de memoria.

Universidad Veracruzana

Parallel Counter GPU



Segunda revisión: Parallel Counter GPU Extra [9]



[9] Aceptada en PRLetters, ayer :-)

Datos

Datos	Ejemplos	Atributos	Clases
airlines	539383	8	2
covtypeNorm	581012	55	7
KDDCup99	4898431	42	23
poker-lsn	829201	11	10
pixelSegCancer[10]	1434060	31	2

[10] H-G Acosta-Mesa, N Cruz-Ramírez y R Hernández-Jiménez, "Aceto-white temporal pattern classification using k-nn to identify precancerous cervical lesion in colposcopic images", *Computers in biology and medicine*, vol. 39, n.º 9, págs. 778-784, 2009.

Precisión

Dataset	Estrategia	Precisión		Test de Wilcoxon
airlines	**Parallel Counter GPU Extra**	65.36 ±	0.25	–
airlines	Weka Centralized	66.34 ±	0.11	Won
airlines	Parallel Counter GPU	66.26 ±	0.12	Won
airlines	Centralizing VFDT	65.24 ±	0.27	Tie
airlines	Bagging	66.45 ±	0.13	Won
airlines	Random Forest	66.76 ±	0.11	Won
covtypeNorm	**Parallel Counter GPU Extra**	92.17 ±	0.52	–
covtypeNorm	Weka Centralized	94.59 ±	0.04	Won
covtypeNorm	Parallel Counter GPU	93.10 ±	0.34	Won
covtypeNorm	Centralizing VFDT	76.83 ±	0.35	Lost
covtypeNorm	Bagging	94.99 ±	0.10	Won
covtypeNorm	Random Forest	78.34 ±	0.39	Lost
KDDCup99	**Parallel Counter GPU Extra**	99.98 ±	0.01	–
KDDCup99	Weka Centralized	99.99 ±	0.01	Tie
KDDCup99	Parallel Counter GPU	99.96 ±	0.01	Lost
KDDCup99	Centralizing VFDT	99.97 ±	0.01	Lost
KDDCup99	Bagging	99.99 ±	0.01	Tie
KDDCup99	Random Forest	99.97 ±	0.01	Lost
poker-lsn	**Parallel Counter GPU Extra**	99.53 ±	0.59	–
poker-lsn	Weka Centralized	99.78 ±	0.01	Tie
poker-lsn	Parallel Counter GPU	98.67 ±	0.46	Lost
poker-lsn	Centralizing VFDT	87.78 ±	1.92	Lost
poker-lsn	Bagging	99.71 ±	0.01	Tie
poker-lsn	Random Forest	96.73 ±	0.25	Lost

Universidad Veracruzana

Memoria y Tiempo

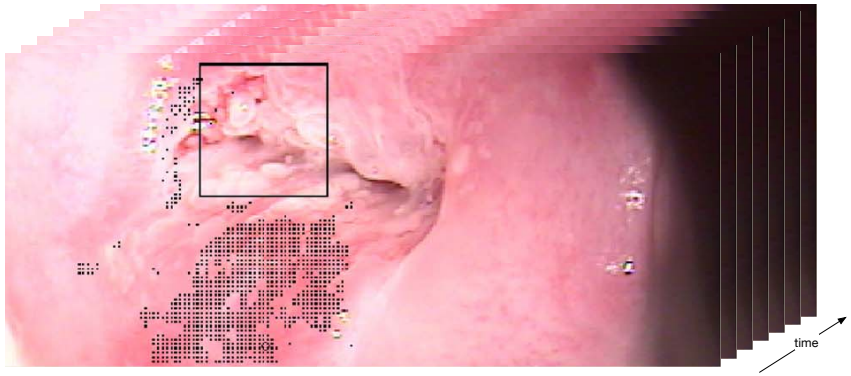
Dataset	Estrategia	%Ejemplos		Tiempo(Segs.)	
airlines	**Parallel Counter GPU Extra**	51.21 ±	0.03	435.04 ±	106.28
airlines	Weka Centralized	100.00 ±	0.00	1164.66 ±	211.76
airlines	Parallel Counter GPU	94.95 ±	0.01	1810.78 ±	446.47
airlines	Centralizing VFDT	100.00 ±	0.00	4.67 ±	0.53
airlines	Bagging	100.00 ±	0.00	144.67 ±	4.47
airlines	Random Forest	100.00 ±	0.00	123.82 ±	3.89
covtypeNorm	**Parallel Counter GPU Extra**	43.28 ±	0.04	817.30 ±	253.27
covtypeNorm	Weka Centralized	100.00 ±	0.00	855.41 ±	97.88
covtypeNorm	Parallel Counter GPU	48.44 ±	0.01	1089.03 ±	277.06
covtypeNorm	Centralizing VFDT	100.00 ±	0.00	8.96 ±	0.56
covtypeNorm	Bagging	100.00 ±	0.00	149.35 ±	5.37
covtypeNorm	Random Forest	100.00 ±	0.00	44.47 ±	4.38
KDDCup99	**Parallel Counter GPU Extra**	4.29 ±	0.01	93.23 ±	6.671
KDDCup99	Weka Centralized	100.00 ±	0.00	1688.91 ±	363.89
KDDCup99	Parallel Counter GPU	9.28 ±	0.01	199.72 ±	45.62
KDDCup99	Centralizing VFDT	100.00 ±	0.00	56.17 ±	1.307
KDDCup99	Bagging	100.00 ±	0.00	371.51 ±	19.39
KDDCup99	Random Forest	100.00 ±	0.00	132.43 ±	21.23
poker-lsn	**Parallel Counter GPU Extra**	8.80 ±	0.01	22.93 ±	3.51
poker-lsn	Weka Centralized	100.00 ±	0.00	174.26 ±	28.55
poker-lsn	Parallel Counter GPU	9.56 ±	0.01	24.90 ±	8.05
poker-lsn	Centralizing VFDT	100.00 ±	0.00	4.25 ±	0.47
poker-lsn	Bagging	100.00 ±	0.00	64.09 ±	5.49
poker-lsn	Random Forest	100.00 ±	0.00	236.34 ±	14.37

Universidad Veracruzana

Segmentación basada en pixeles

Secuencias de imágenes de colposcopia, presentando posibles lesiones cervicales pre-cancerosas:

- 38 pacientes.
- Pre-procesamiento con FIJI [14].
- 1,434,060 pixeles de entrenamiento.
- 30 atributos.
- 6 clases, agrupadas en 2.



Universidad Veracruzana

Precisión, sensibilidad y especificidad

Estrategia	Precisión		Wilcoxon test	Sen	Esp
Parallel Counter GPU Extra	67.61 ±	19.32	–	60.96	64.83
Weka Centralized	63.68 ±	18.44	Lost	60.80	61.60
Centralizing VFDT	53.34 ±	20.58	Lost	53.10	58.51
Bagging	64.25 ±	21.78	Lost	65.40	59.16
Random Forest	58.88 ±	23.71	Lost	68.78	49.34
Acosta2009	67.00 ±	n/a	n/a	71.00	59.00

Universidad Veracruzana

Número de ejemplos y tiempo

Estrategia	% Ejemplos		Tiempo (Seg.)	
Parallel Counter GPU Extra	37.00 ±	3.52	3782.26 ±	1094.21
Weka Centralized	100.00 ±	0.00	6436.64 ±	923.16
Centralizing VFDT	100.00 ±	0.00	32.03 ±	2.61
Bagging	100.00 ±	0.00	1138.83 ±	108.83
Random Forest	100.00 ±	0.00	1817.10 ±	179.18
Acosta2009	n/a		n/a	

Trabajo Futuro

- Mejorar la facilidad de uso:
  - Interfaz como un servicio Web
  - Lenguaje DSL / Gráfico basado en AUML
  - Mejorar transparencia de CArtAgO
- Analizar la relación precisión vs ejemplos.
- Analizar generalización del Windowing.
- Buscar Otras Aplicaciones:
  - Análisis de tendencias en twitter
  - Simulación Social[11]
- Requisitos: Java, Estadística, Weka, Inglés, Ganas
- Oferta: Becas maestría y doctorado (Promedio > 8/10)

[11] JC Díaz-Preciado, A Guerra-Hernández y N Cruz-Ramírez, "Un modelo de red bayesiana de la informalidad laboral en veracruz oriendado a una simulación social basada en agentes", *Research in Computing Science*, vol. 113, n.º 2016, págs. 157-170, 2016.

Referencias I

- H-G Acosta-Mesa, N Cruz-Ramírez y R Hernández-Jiménez, "Aceto-white temporal pattern classification using k-nn to identify precancerous cervical lesion in colposcopic images", *Computers in biology and medicine*, vol. 39, n.º 9, págs. 778-784, 2009.
- KA Albashiri y F Coenen, "Agent-enriched data mining using an extendable framework", en *Agents and Data Mining Interaction*, Springer, 2009, págs. 53-68.
- S Bailey, R Grossman, H Sivakumar y A Turinsky, "Papyrus: A system for data mining over local and wide area clusters and super-clusters", en *Proceedings of the 1999 ACM/IEEE conference on Supercomputing*, ACM, 1999, pág. 63.
- A Bifet, G Holmes, R Kirkby y B Pfahringer, "Moa: Massive online analysis", *The Journal of Machine Learning Research*, vol. 11, págs. 1601-1604, 2010.
- RH Bordini, JF Hübner y M Wooldridge, *Programming multi-agent systems in agent-speak using jason*. John Wiley & Sons Ltd, 2007.

Universidad Veracruzana

Referencias II

- JC Díaz-Preciado, A Guerra-Hernández y N Cruz-Ramírez, "Un modelo de red bayesiana de la informalidad laboral en veracruz oriendado a una simulación social basada en agentes", *Research in Computing Science*, vol. 113, n.º 2016, págs. 157-170, 2016.
- U Fayyad, G Piatetsky-Shapiro y P Smyth, "From data mining to knowledge discovery in databases", *AI Magazine*, vol. 17, n.º 3, págs. 37-54, 1996.
- H Kargupta, DH Byung-Hoon y E Johnson, "Collective data mining: A new perspective toward distributed data analysis", en *Advances in Distributed and Parallel Knowledge Discovery*, Citeseer, 1999.
- X Limón, A Guerra-Hernández, N Cruz-Ramírez y F Grimaldo, "An agents & artifacts approach to distributed data mining", en *MICAI 2013: Eleventh Mexican International Conference on Artificial Intelligence*, F Castro, A Gelbukh y MG Mendoza, eds., ép. Lecture Notes in Artificial Intelligence, vol. 8266, Berlin Heidelberg: Springer Verlag, 2013, págs. 338-349.

Universidad Veracruzana

Referencias III

X Limón, A Guerra-Hernández, N Cruz-Ramírez, HG Acosta-Mesa y F Grimaldo, "A windowing based GPU optimized strategy for the induction of decision trees", en *Arfificial Intelligence Research and Development*, E Armengol, D Boixader y F Grimaldo, eds., ép. Frontiers in Artificial Intelligence and Applications, vol. 277, Amsterdam, NL: IOS Press, 2015, págs. 100-109.

C Moemeng, X Zhu, L Cao y C Jiahang, "I-analyst: An agent-based distributed data mining platform", en *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, IEEE, 2010, págs. 1404-1406.

JR Quinlan, *C4. 5: Programs for machine learning*. San Mateo, CA., USA: Morgan kaufmann, 1993, vol. 1.

A Ricci, M Piunti y M Viroli, "Environment programming in multi-agent systems: An artifact-based perspective", *Autonomous Agents and Multi-Agent Systems*, vol. 23, n.º 2, págs. 158-192, 2011.

J Schindelin, I Arganda-Carreras, E Frise, V Kaynig, M Longair, T Pietzsch, S Preibisch, C Rueden, S Saalfeld, B Schmid y col., "Fiji: An open-source platform for biological-image analysis", *Nature methods*, vol. 9, n.º 7, págs. 676-682, 2012.

Universidad Veracruzana

Referencias IV

SJ Stolfo, AL Prodromidis, S Tselepis, W Lee, DW Fan y PK Chan, "Jam: Java agents for meta-learning over distributed databases.", en *KDD*, vol. 97, 1997, págs. 74-81.

IH Witten, E Frank y MA Hall, *Data mining: Practical machine learning toools and techniques*. Burlington, MA., USA: Morgan Kaufmann Publishers, 2011.

J Xu, L Yao, L Li e Y Chen, "Sampling based multi-agent joint learning for association rule mining", en *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, International Foundation for Autonomous Agents y Multiagent Systems, 2014, págs. 1469-1470.

N Zhong, Y Matsui, T Okuno y C Liu, "Framework of a multi-agent KDD system.", en *IDEAL 2002*, H Yin, ed., ép. Lecture Notes in Computer Science, vol. 2412, Berlin Heidelberg: Springer-Verlag, 2002, págs. 337-346.

Universidad Veracruzana