A Hybrid Construction of a Medical Decision Support System using Semantic Web and Machine Learning Techniques (Atif Khan – PhD Seminar)

Atif Khan    John Doucette    Robin Cohen

David R. Cheriton School of Computer Science
University of Waterloo

September 13, 2012
Motivation

Medical Decision Support Systems (MDSS)

- can a drug/procedure be administered to Alice?

Challenges

- information constraints – access, completeness
- expert knowledge – who is treating Alice
- temporal aspects – emergency medical scenarios
Motivation

Medical Decision Support Systems (MDSS)
- can a drug/procedure be administered to Alice?

Challenges
- information constraints – access, completeness
- expert knowledge – who is treating Alice
- temporal aspects – emergency medical scenarios
Motivation

Medical Decision Support Systems (MDSS)
- can a drug/procedure be administered to Alice?

Challenges
- information constraints – access, completeness
- expert knowledge – who is treating Alice
- temporal aspects – emergency medical scenarios
Motivation

Medical Decision Support Systems (MDSS)
- can a drug/procedure be administered to Alice?

Challenges
- information constraints – access, completeness
- expert knowledge – who is treating Alice
- temporal aspects – emergency medical scenarios
Motivation

Medical Decision Support Systems (MDSS)

- can a drug/procedure be administered to Alice?

Challenges

- information constraints – access, completeness
- expert knowledge – who is treating Alice
- temporal aspects – emergency medical scenarios
Build a medical decision support system with the capability to handle the following *knowledge* features:

- black swan theory & Alice
- Alice’s medical history
- nature of the drug/procedure
- expert knowledge
- information availability
Outline

1. Background
2. Architecture
3. Experimental Validation
4. Conclusion
Background

Medical Decision Support Systems (MDSS)

**Definition**

Computer systems designed to impact clinician decision making about individual patients.

(Berner, 2007)

**Definition**

Clinical decision support systems link health observations with health knowledge to influence health choices by clinicians for improved health care.

(Dr. R. Hayward, 2004)
Medical Decision Support Systems (MDSS)

Definition

computer systems designed to **impact clinician decision** making about **individual** patients.

(Berner, 2007)

Definition

clinical decision support systems **link** health **observations** with health **knowledge** to influence health choices by clinicians for **improved** health **care**.

(Dr. R. Hayward, 2004)
Medical Decision Support Systems (MDSS)

**Definition**

Computer systems designed to impact clinician decision making about individual patients.

(Berner, 2007)

**Definition**

Clinical decision support systems link health observations with health knowledge to influence health choices by clinicians for improved health care.

(Dr. R. Hayward, 2004)
MDSS Classification (Berner, 2007)

Medical Decision Support Systems

Knowledge-based MDSS  Nonknowledge-based MDSS
Characteristics

Knowledge-based MDSS

- structured data representation (schema)
- knowledge is persisted in data-stores
- expert knowledge $\rightarrow$ system rules
  - heuristics based
  - evidence based
- reasoning capacity using inference engines

Nonknowledge-based MDSS

- learn from raw data (semi/un-structured)
- based on machine learning techniques
  - patterns in the data
  - past examples/cases
- learning capacity
- probabilistic prediction capability
Characteristics

Knowledge-based MDSS
- structured data representation *(schema)*
- knowledge is persisted in *data-stores*
- expert knowledge \(\rightarrow\) system rules
  - heuristics based
  - evidence based
- reasoning capacity using *inference* engines

Nonknowledge-based MDSS
- learn from raw data *(semi/un-structured)*
- based on machine learning techniques
  - patterns in the data
  - past examples/cases
- learning capacity
- probabilistic prediction capability
Advantages & Disadvantages

Knowledge-based MDSS

- quite capable and robust when the knowledge base is complete
- system made decisions are
  - based on logical rules/axioms
  - can be easily explained to end users ✓
  - can be verified using logic proofs ✓

Nonknowledge-based MDSS

- generally tolerant to noise ✓
- may mistake weaker signals in data as noise
- computationally expensive to build and maintain
  - require a training phase
  - specific to a line of inquiry
  - require retraining as more information becomes available
Advantages & Disadvantages

Knowledge-based MDSS
- quite capable and robust when the knowledge base is complete
- system made decisions are
  - based on logical rules/axioms
  - can be easily explained to end users ✓
  - can be verified using logic proofs ✓

Nonknowledge-based MDSS
- generally tolerant to noise ✓
- may mistake weaker signals in data as noise
- computationally expensive to build and maintain
  - require a training phase
  - specific to a line of inquiry
  - require retraining as more information becomes available
Ontology-based Structured Knowledge Representation
Ontology-based Structured Knowledge Representation
Ontology-based Structured Knowledge Representation

Patient

Disease

hasDisease

Patient

hasCondition

Condition

Drug

treats

treats

Khan et al. (Computer Science)
Ontology-based Structured Knowledge Representation

- **Patient**
- **Disease**
- **Drug**
- **Condition**

- Patient hasCondition Disease
- Patient hasCondition Condition
- Patient hasContraindication Drug
- Disease treats Drug
- Condition treats Drug
- Drug hasContraindication Disease
- Drug hasContraindication Condition
- Drug hasContraindication Drug
Ontology-based Structured Knowledge Representation

Ontology

Let $V$ be the set of structured vocabulary, and $A_x$ axioms about $V$, which are formulated in formal language $L$.

An ontology is a sign-system: $\text{ont} = (L, V, A_x)$,

where, the symbols of $V$ denote categories, and relations between categories or between their instances.

(Hussain, 2009)
Ontology-based Structured Knowledge Representation

Ontology

Let $V$ be the set of structured vocabulary, and $A_x$ axioms about $V$, which are formulated in formal language $L$.

An ontology is a sign-system: $ont = (L, V, A_x)$,

where, the symbols of $V$ denote categories, and relations between categories or between their instances.

(Hussain, 2009)
Knowledge Inference & Reasoning

Rules-based inference:
- discover implicit knowledge;
  \[ \{ \text{assertions} \} \rightarrow \{ \text{implications} \} \]

Reasoning

- result $\rightarrow$ query answer
- proof $\rightarrow$ based on first order logic, represents a unique traversal path through the knowledge graph
Knowledge Inference & Reasoning

Rules-based inference:
- discover implicit knowledge;
  \( \{\text{assertions}\} \rightarrow \{\text{implications}\} \)

Reasoning

**Result** → query answer

**Proof** →
based on first order logic, represents a unique traversal path through the knowledge graph
A Simple Example – Who has high blood pressure?

Knowledge Base

: Alice a : Patient; : hasSystolic 119; : hasDiastolic 75.
: Kate a : Patient; : hasSystolic 144; : hasDiastolic 91.
: Dave a : Patient; : hasSystolic 120; : hasDiastolic 101.
: Bob a : Patient; : hasCondition : HighBloodPressure.
: John a : Patient.

Inference Rule

{ ?P a : Patient; : hasSystolic ?SYS. ?SYS math:greaterThan 140. }
⇒ { ?P : hasCondition : HighBloodPressure }.

{ ?P a : Patient; : hasDiastolic ?DIA. ?DIA math:greaterThan 90. } 
⇒ { ?P : hasCondition : HighBloodPressure }.

Query

_ : WHO : hasCondition : HighBloodPressure.
A Simple Example – Who has high blood pressure?

Knowledge Base

: Alice a : Patient ; : hasSystolic 119 ; : hasDiastolic 75.
: Kate a : Patient ; : hasSystolic 144 ; : hasDiastolic 91.
: Dave a : Patient ; : hasSystolic 120 ; : hasDiastolic 101.
: Bob a : Patient ; : hasCondition : HighBloodPressure.
: John a : Patient.

Inference Rule


Query

_ : WHO : hasCondition : HighBloodPressure.
A Simple Example – Who has high blood pressure?

Knowledge Base

:Alice a :Patient ; :hasSystolic 119 ; :hasDiastolic 75 .
:Kate a :Patient ; :hasSystolic 144 ; :hasDiastolic 91 .
:Dave a :Patient ; :hasSystolic 120 ; :hasDiastolic 101 .
:Bob a :Patient ; :hasCondition :HighBloodPressure .
:John a : Patient .

Inference Rule

{?P a :Patient ; :hasSystolic ?SYS . ?SYS math :greaterThan 140 .}
⇒ {?P :hasCondition :HighBloodPressure}.

{?P a :Patient ; :hasDiastolic ?DIA . ?DIA math :greaterThan 90 .}
⇒ {?P :hasCondition :HighBloodPressure}.

Query

_:WHO :hasCondition :HighBloodPressure .
A Simple Example – Who has high blood pressure?

Result & Proof

{ :Bob :hasCondition :HighBloodPressure} e:evidence <kb.n3#_12>.

{ { :Kate a :Patient} e:evidence <kb.n3#_10>.
  { :Kate :hasSystolic 144} e:evidence <kb.n3#_10>.
  {144 math:greaterThan 140} e:evidence <math#kb>} => { 
  { :Kate :hasCondition :HighBloodPressure} e:evidence <rules.n3#_3>}. 

{ { :Dave a :Patient} e:evidence <kb.n3#_11>.
  { :Dave :hasDiastolic 101} e:evidence <kb.n3#_11>.
  {101 math:greaterThan 90} e:evidence <math#kb>} => { 
  { :Dave :hasCondition :HighBloodPressure} e:evidence <rules.n3#_7>}. 

But what about John? → open vs. closed world assumptions.
A Simple Example – Who has high blood pressure?

Result & Proof

{:Bob :hasCondition :HighBloodPressure} e:evidence <kb.n3#_12>.

{:Kate a:Patient} e:evidence <kb.n3#_10>.
{:Kate :hasSystolic 144} e:evidence <kb.n3#_10>.
{144 math:greaterThan 140} e:evidence <math#kb>} => 
{:Kate :hasCondition :HighBloodPressure} e:evidence <rules.n3#_3>.

{:Dave a:Patient} e:evidence <kb.n3#_11>.
{:Dave :hasDiastolic 101} e:evidence <kb.n3#_11>.
{101 math:greaterThan 90} e:evidence <math#kb>} => 
{:Dave :hasCondition :HighBloodPressure} e:evidence <rules.n3#_7>.

But what about John? → open vs. closed world assumptions.
A Simple Example – Who has high blood pressure?

Result & Proof

{:Bob :hasCondition :HighBloodPressure} e:evidence <kb.n3#_12>.

{{:Kate a :Patient} e:evidence <kb.n3#_10>.
{:Kate :hasSystolic 144} e:evidence <kb.n3#_10>.
{144 math:greaterThan 140} e:evidence <math#kb>} \Rightarrow \{
{:Kate :hasCondition :HighBloodPressure} e:evidence <rules.n3#_3>\}.

{{:Dave a :Patient} e:evidence <kb.n3#_11>.
{:Dave :hasDiastolic 101} e:evidence <kb.n3#_11>.
{101 math:greaterThan 90} e:evidence <math#kb>} \Rightarrow \{
{:Dave :hasCondition :HighBloodPressure} e:evidence <rules.n3#_7>\}.

But what about John? → open vs. closed world assumptions.
A Simple Example – Who has high blood pressure?

Result & Proof

{:Bob :hasCondition :HighBloodPressure} e:evidence <kb.n3#_12>.

{{:Kate a :Patient} e:evidence <kb.n3#_10>.
{:Kate :hasSystolic 144} e:evidence <kb.n3#_10>.
{144 math:greaterThan 140} e:evidence <math#kb>} \implies \{
{:Kate :hasCondition :HighBloodPressure} e:evidence <rules.n3#_3>\}.

{{:Dave a :Patient} e:evidence <kb.n3#_11>.
{:Dave :hasDiastolic 101} e:evidence <kb.n3#_11>.
{101 math:greaterThan 90} e:evidence <math#kb>} \implies \{
{:Dave :hasCondition :HighBloodPressure} e:evidence <rules.n3#_7>\}.

But what about John? \rightarrow open vs. closed world assumptions.
Machine Learning

Key Tasks:

- *classification*: predict the class of an instance of data
- *regression*: prediction of a numeric value
- *clustering*: group similar items together
Machine Learning

Key Tasks:

1. **supervised learning**
   - *classification*: predict the class of an instance of data
   - *regression*: prediction of a numeric value

2. **unsupervised learning**
   - *clustering*: group similar items together
Machine Learning

Our focus:

1. supervised learning
   - classification: predict the class of an instance of data
Let $X$ be the input space and $Y$ be the output space. Then a training set of examples can be defined as:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}.$$ 

The machine learning task is to induce a function $p : X \rightarrow Y$ that best explains the training data.

where,

- **best** $\rightarrow$ minimizing “loss”, via a loss function $L = f(p(x_i), y_i)$
- $p(x_i)$ is predicted output, and $y_i$ is actual output.
- $x_i$ is represented as a feature vector.

(Lin and Kolcz, 2012)
Machine Learning–Classification

Definition

Let $X$ be the input space and $Y$ be the output space. Then a training set of examples can be defined as:

$$D = \{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}.$$  

The machine learning task is to induce a function $p: X \rightarrow Y$ that best explains the training data.

where,

- **best** → minimizing “loss”, via a loss function $L = f(p(x_i), y_i)$
- $p(x_i)$ is predicted output, and $y_i$ is actual output.
- $x_i$ is represented as a feature vector.

(Lin and Kolcz, 2012)
Machine Learning–Classification

Definition

Let $X$ be the input space and $Y$ be the output space. Then a training set of examples can be defined as:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}.$$ 

The machine learning task is to induce a function $p : X \rightarrow Y$ that best explains the training data.

where,

**best** $\rightarrow$ minimizing “loss”, via a loss function $L = f(p(x_i), y_i)$

$p(x_i)$ is predicted output, and $y_i$ is actual output.

$x_i$ is represented as a feature vector.

(Lin and Kolcz, 2012)
Steps

- data collection & pre-processing
- data analysis (abnormal values, outliers etc.)
- feature selection & labelling
- train – build a classifier based on the training examples
- test – evaluate the classifier based on the test examples
- system integration of the classifier
Outline

1. Background
2. Architecture
3. Experimental Validation
4. Conclusion
Proposed Solution

Design Goals

1. **Patient-centric**, evidence-based
2. Automated (machine processable)
3. Operate in constrained environments
4. Decisions are easy to explain and validate
5. Tolerant to noise in patient data → *information challenge*

**Note:** A knowledge-based MDSS meets 1-4 design objectives but fails to meet 5
Proposed Solution

Design Goals

1. **patient-centric**, evidence-based
2. automated (machine processable)
3. operate in constrained environments
4. decisions are easy to explain and validate
5. tolerant to noise in patient data → *information challenge*

Note: A *knowledge-based MDSS* meets 1-4 design objectives but *fails to meet* 5
Proposed Solution: OMeD – Knowledge-based MDSS

(Khan et al., 2011)

Design Characteristics
- ontological data representation
- expert knowledge as inference rules
- logic-based decision making

Drawbacks
- susceptible to noise in data
Proposed Solution: OMeD – Knowledge-based MDSS

Design Characteristics
- ontological data representation
- expert knowledge as inference rules
- logic-based decision making

Drawbacks
- susceptible to noise in data

(Khan et al., 2011)
Proposed Solution: OMeD – Knowledge-based MDSS

(Khan et al., 2011)
Proposed Solution: Nonknowledge-based Engine

Recall: ML techniques are tolerant to noise

**Design update:** Replace semantic reasoner with a ML-based classifier

**Validation strategy**
- *Line of inquiry:* drug prescription
- Synthetic dataset: \{Patient, Drug, Disease\}
  - patient-to-drug interactions
  - drug-to-drug interactions
  - disease-to-drug interactions
- *Result:* ML based classifiers performed poorly at prescribing the right drugs to the right patients

(Doucette, Khan, and Cohen, 2012)
Proposed Solution: Nonknowledge-based Engine

Recall: ML techniques are tolerant to noise

Design update: Replace semantic reasoner with a ML-based classifier

Validation strategy

- line of inquiry: drug prescription
- synthetic dataset: \{Patient, Drug, Disease\}
  - patient-to-drug interactions
  - drug-to-drug interactions
  - disease-to-drug interactions
- result: ML based classifiers performed poorly at prescribing the right drugs to the right patients

(Doucette, Khan, and Cohen, 2012)
Proposed Solution – Hybrid MDSS

[Diagram showing the architecture of a Hybrid MDSS system, including components like query execution engine, semantic reasoner, communication interface module, knowledge management component, and different types of data representation (semantics, structured, semi-structured)].

Khan et al. (Computer Science)
Proposed Solution – Hybrid MDSS
Proposed Solution – Hybrid MDSS
Algorithm

**Algorithm 1: Hybrid Decision Making Algorithm**

**Input:** query, KB: knowledge base, rules: set of inference rules, reasoner: semantic reasoning, mlrecommender: machine learning based classifier.

**Output:** result, proof, conf confidence in the result.

```plaintext
response[r, p] = reasoner.doProof(query, KB, rules);
if response[r, p] not empty then
    return (response.r, response.p, 1.0);
end
noresult = inspectForFalseModel(proof);
unknownresult = inspectForCounterModel(proof);
if noresult and not unknownresult then
    return (null, null, 1.0);
end
// predict missing values
if unknownresult then
    predictedValues[] = Use mlrecommender to predict values of the missing attributes.
    KB = KB + predictedValues[];
    response[r, p] = reasoner.doProof(query, KB, rules);
    conf = based on the combined probabilities of all recommended attributes used
    return (response.r, response.p, conf);
end
```
Algorithm 1: Hybrid Decision Making Algorithm

**Input:** query, KB: knowledge base, rules: set of inference rules, reasoner: semantic reasoning, mltrecomposer: machine learning based classifier.

**Output:** result, proof, conf confidence in the result.

```plaintext
response[r, p] = reasoner.doProof(query, KB, rules);
if response[r, p] not empty then
    return (response.r, response.p, 1.0);
end

noresult = inspectForFalseModel(proof);
unknownresult = inspectForCounterModel(proof);
if noresult and not unknownresult then
    return (null, null, 1.0);
end

// predict missing values
if unknownresult then
    predictedValues[] = Use mltrecomposer to predict values of the missing attributes.
    KB = KB + predictedValues[];
    response[r, p] = reasoner.doProof(query, KB, rules);
    conf = based on the combined probabilities of all recommended attributes used
    return (response.r, response.p, conf);
end
```
Algorithm 1: Hybrid Decision Making Algorithm

**Input:** query, KB: knowledge base, rules: set of inference rules, reasoner: semantic reasoning, mlrecommender: machine learning based classifier.

**Output:** result, proof, conf confidence in the result.

```plaintext
reponse[r, p] = reasoner.doProof(query, KB, rules);
if reponse[r, p] not empty then
    return (reponse.r, reponse.p, 1.0);
end

noresult = inspectForFalseModel(proof);
unknownresult = inspectForCounterModel(proof);
if noresult and not unknownresult then
    return (null, null, 1.0);
end

//predict missing values
if unknownresult then
    predictedValues[] = Use mlrecommender to predict values of the missing attributes.
    KB = KB + predictedValues[];
    reponse[r, p] = reasoner.doProof(query, KB, rules);
    conf = based on the combined probabilities of all recommended attributes used
    return (reponse.r, reponse.p, conf);
end
```

**False model:** deduction fails due to the facts themselves

**Counter model:** deduction fails due to incomplete facts
Algorithm 1: Hybrid Decision Making Algorithm


Output: result, proof, conf confidence in the result.

reponse[r, p] = reasoner.doProof(query, KB, rules);
if reponse[r, p] not empty then
    return (reponse.r, reponse.p, 1.0);
end

noresult = inspectForFalseModel(proof);
unknownresult = inspectForCounterModel(proof);
if noresult and not unknownresult then
    return (null, null, 1.0);
end

// predict missing values
if unknownresult then
    predictedValues[] = Use mlrecommender to predict values of the missing attributes.
    KB = KB + predictedValues[];
    reponse[r, p] = reasoner.doProof(query, KB, rules);
    conf = based on the combined probabilities of all recommended attributes used
    return (reponse.r, reponse.p, conf);
end
Outline

1 Background

2 Architecture

3 Experimental Validation

4 Conclusion
Line of Inquiry: *Sleeping pill prescription*

which patients can be prescribed what sleep medications?

prescribing sleep medication is not trivial
Dataset – BRFSS

Patient Records

- Behavioral Risk Factor Surveillance System (BRFSS)
  Center of Disease Control and Prevention
- 2010 dataset (records: 450K+, features: 400+)
- multi-dimensional
  - demographic information
    (age, race, sex, geographic location)
  - medical information
    (cancer, asthma, mental illness, diabetes)
  - behavioural Information
    (alcohol consumption, drug use, sleep deprivation)
Dataset – BRFSS

Patient Records

- Behavioral Risk Factor Surveillance System (BRFSS)
  Center of Disease Control and Prevention
- 2010 dataset (records: 450K+, features: 400+)
- multi-dimensional
  - demographic information
    (age, race, sex, geographic location)
  - medical information
    (cancer, asthma, mental illness, diabetes)
  - behavioural Information
    (alcohol consumption, drug use, sleep deprivation)
Experimental Validation

Additional Datasets: Expert Knowledge

Mayo clinic sleeping pill prescription protocol
- describes expert rules dictating what sleeping drugs can be administered under a given set of medical conditions
- available online (HTML format)

Drug-to-drug interaction rules
- from drug.com online registry
- pain and sleeping medication interactions
Ontological Knowledge Representation
Ontological Knowledge Representation
BRFSS Data to Patient Records

Data mapping
- BRFSS code book defined the **semantics** of the raw values
- the raw values were then mapped to **ontological concepts**
Mayo Clinic Sleeping Pill Prescription Protocol

- Pregnancy
- BreastFeeding
- AlcoholAbuse
- Elderly
- Propoxyphene

- Ramelteon
- Eszopiclone

relationships:
- hasContraindication
- prescribedFor

- SleepApnea
- Asthma
- Insomnia
- LungDisease
- Depression
- LiverDisease

Khan et al. (Computer Science)
Experimental Validation

Expert Knowledge Representation

Drug-to-Drug Interactions

: Propoxyphene a : Drug;
  : isPrescribedFor : Pain;
  : isContraIndictive : Eszopiclone.

: Wygesic a : Drug;
  : isPrescribedFor : Pain;
  : isContraIndictive : Eszopiclone.

: Trycet a : Drug;
  : isPrescribedFor : Pain;
  : isContraIndictive : Eszopiclone.

: Propacet100 a : Drug;
  : isPrescribedFor : Pain;
  : isContraIndictive : Eszopiclone.

: Aspirin a : Drug;
  : isPrescribedFor : Pain.

: Tylenol1 a : Drug;
  : isPrescribedFor : Pain.

: Tylenol2 a : Drug;
  : isPrescribedFor : Pain;
  : isContraIndictive : SleepingMedication.

N3 Tripple representation
Inference Rules

Drug-to-Drug Interactions

If a patient is taking an existing drug $D1$ and $D1$ has contraindication to another drug $D2$ then drug $D2$ should not be prescribed to the patient.

N3 Representation

$$\{ \text{?P a : Patient.} \text{, ?D1 a : Drug.} \text{, ?D2 a : Drug.} \text{, ?P : isTaking ?D1.} \text{, ?D1 : hasContraindication ?D2.} \} \implies \{ \text{?P : cannotBeGiven ?D2} \}.$$
Inference Rules

Drug-to-Disease Interactions

If a patient has a condition that has a contraindication to a drug then the patient **should not** be given the drug

N3 Representation

\[
\{ ?P : \text{Patient} . \\
?D : \text{Drug} . \\
?P : \text{hasDisease} \ ?DIS. \\
?D : \text{hasContraindication} \ ?DIS. \} \Rightarrow \{ ?P : \text{cannotBeGiven} \ ?D \}. 
\]
Putting it All Together

DataSets
- BRFSS-2010
- Mayo Clinic sleeping pill prescription protocol
- sleeping pill-to-pain medication interaction

Knowledge engineering:
- Resource Description Framework (RDF)/Notation-3 (N3) based ontological model
- scenario specific ontology
- inference rules

Semantic Reasoner
- EulerSharp

Machine Learning toolkit
- Weka
Evaluation Criteria

**Sensitivity**
identify true positives

\[ Sens = \frac{tp}{tp + fn} \]

**Specificity**
identify true negatives

\[ Spec = \frac{tn}{tn + fp} \]

**Balanced Accuracy**
simple average of specificity and sensitivity

\[ balAcc = \frac{Spec + Sens}{2} \]

where,

- \( tp \) = true positive,
- \( fp \) = false positive,
- \( tn \) = true negative,
- \( fn \) = false negative,
**Evaluation Criteria**

**Sensitivity**
- identify true positives

\[
Sens = \frac{tp}{tp + fn}
\]

**Specificity**
- identify true negatives

\[
Spec = \frac{tn}{tn + fp}
\]

**Balanced Accuracy**
- simple average of specificity and sensitivity

\[
bal\text{Acc} = \frac{Spec + Sens}{2}
\]

where,
- \(tp\) = true positive,
- \(fp\) = false positive,
- \(tn\) = true negative,
- \(fn\) = false negative,
### Evaluation Criteria

<table>
<thead>
<tr>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Balanced Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>identify true positives</td>
<td>identify true negatives</td>
<td>simple average of specificity and sensitivity</td>
</tr>
<tr>
<td>$Sens = \frac{tp}{tp + fn}$</td>
<td>$Spec = \frac{tn}{tn + fp}$</td>
<td>$balAcc = \frac{Spec + Sens}{2}$</td>
</tr>
</tbody>
</table>

where,

- $tp = \text{true positive}$,
- $fp = \text{false positive}$,
- $tn = \text{true negative}$,
- $fn = \text{false negative}$,
System Evaluation

3 Stage Experiment:

1. evaluate machine learning based MDSS on BRFSS patient dataset
2. introduce *information challenge*
3. evaluate the hybrid construction
**Ex 1 – ML Evaluation**

**Goal**

determine the best performing machine learning algorithm for BRFSS dataset to prescribe sleeping aids

**Setup**

1. **algorithms**: decision stump, C4.5-J8, Bagging and AdaBoost
2. **example data**: 50 different randomly selected training sets (of two sizes: 2500 exemplars and 5000 exemplars)
3. **features**: *information gain-based* feature selection algorithm (Yang and Pedersen, 1997) to select 30 features
4. **labelling**: ground truth was established using the output of the knowledge-based reasoner *where possible*
Ex 1 – ML Evaluation

Training Performance with 5000 Training Exemplars

Test Performance with 5000 Training Exemplars

Bagging seems to overfit. AdaBoost performs the best but still not good enough.
Ex 1 – ML Evaluation

Training Performance with 5000 Training Exemplars

Test Performance with 5000 Training Exemplars

Bagging seems to overfit
Ex 1 – ML Evaluation

**Training Performance with 5000 Training Exemplars**

- **bagging** seems to overfit

**Test Performance with 5000 Training Exemplars**

- **AdaBoost** performs the best but still not good enough
Ex 2 – Tolerance to Missing Information

**Goal**

study the impact of data *missingness* \((\epsilon)\) for **AdaBoost** based classifiers

**Setup**

1. **noise \(\rightarrow\) missing data:** removing known values from the patient records
2. **noise factor** \(\epsilon\): describes the probability of introducing noise at random across all insomnia related features
3. **information challenge:** for each value of \(\epsilon\),
   - create sample datasets
     - (50 sets of 5000 exemplars from the noised data)
   - train AdaBoost based classifier
Ex 2 – Tolerance to Missing Information

Training Performance for Experiment 2

Test Performance for Experiment 2
Ex 2 – Tolerance to Missing Information

AdaBoost based classifiers are tolerant to ‘missingness’
Ex 3 – Hybrid Construction Evaluation

**Goal**

hybrid construction to impute missing information

Data imputation:

\[ R_{org} = \{ f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9 \} \]

\[ R_{nei} = \{ f_1, \Box, f_3, f_4, \Box, \Box, f_7, f_8, f_9 \} \]

\[ R_{imp} = \{ f_1, p_2, f_3, f_4, p_5, p_6, f_7, f_8, f_9 \} \]
Experimental Validation

Ex 3 – Hybrid Construction Evaluation

Goal

hybrid construction to impute missing information

Data imputation:

\[
R_{org} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9\}
\]

\[
R_{nei} = \{f_1, □, f_3, f_4, □, □, f_7, f_8, f_9\}
\]

\[
R_{imp} = \{f_1, p_2, f_3, f_4, p_5, p_6, f_7, f_8, f_9\}
\]
Goal

hybrid construction to impute missing information

Data imputation:

$$R_{org} = \{ f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9 \}$$

$$R_{n\epsilon_i} = \{ f_1, \Box, f_3, f_4, \Box, \Box, f_7, f_8, f_9 \}$$

$$R_{imp} = \{ f_1, p_2, f_3, f_4, p_5, p_6, f_7, f_8, f_9 \}$$
Experimental Validation

Ex 3 – Hybrid Construction Evaluation

Setup

For a given $\epsilon$:

1. transform $R_{org} \rightarrow R_{n\epsilon}$
2. from $R_{n\epsilon}$ generate an example dataset for training and testing
3. $R_{org}$ is used for establishing ground truth for labelling
4. learn an AdaBoost classifier for each missing feature to impute
5. predict the missing value using the feature classifier
6. observe the impact of missingness on the knowledge-based MDSS
Experimental Validation

Ex 3 – Hybrid Construction Evaluation

Setup

For a given $\epsilon$:

1. transform $R_{org} \rightarrow R_{n\epsilon}$
2. from $R_{n\epsilon}$ generate an example dataset for training and testing
3. $R_{org}$ is used for establishing ground truth for labelling
4. learn an AdaBoost classifier for each missing feature to impute
5. predict the missing value using the feature classifier
6. observe the impact of missingness on the knowledge-based MDSS
Ex 3 – Hybrid Construction Evaluation

Setup

For a given $\epsilon$:

1. transform $R_{\text{org}} \rightarrow R_{n\epsilon}$
2. from $R_{n\epsilon}$ generate an example dataset for training and testing
3. $R_{\text{org}}$ is used for establishing ground truth for labelling
4. learn an AdaBoost classifier for each missing feature to impute
5. predict the missing value using the feature classifier
6. observe the impact of missingness on the knowledge-based MDSS
Ex 3 – Hybrid Construction Evaluation

### Setup

For a given $\epsilon$:

1. transform $R_{\text{org}} \rightarrow R_{n\epsilon}$
2. from $R_{n\epsilon}$ generate an example dataset for training and testing
3. $R_{\text{org}}$ is used for establishing ground truth for labelling
4. learn an AdaBoost classifier for each missing feature to impute
5. predict the missing value using the feature classifier
6. observe the impact of missingness on the knowledge-based MDSS
Ex 3 – Hybrid Construction Evaluation

Setup

For a given $\epsilon$:

1. transform $R_{org} \rightarrow R_{n\epsilon}$
2. from $R_{n\epsilon}$ generate an example dataset for training and testing
3. $R_{org}$ is used for establishing ground truth for labelling
4. learn an AdaBoost classifier for each missing feature to impute
5. predict the missing value using the feature classifier
6. observe the impact of missingness on the knowledge-based MDSS
Ex 3 – Hybrid Construction Evaluation

Setup

For a given $\epsilon$:

1. transform $R_{org} \rightarrow R_{n\epsilon}$
2. from $R_{n\epsilon}$ generate an example dataset for training and testing
3. $R_{org}$ is used for establishing ground truth for labelling
4. learn an AdaBoost classifier for each missing feature to impute
5. predict the missing value using the feature classifier
6. observe the impact of missingness on the knowledge-based MDSS
**Setup**

For a given $\epsilon$:

1. **transform** $R_{\text{org}} \rightarrow R_{n\epsilon}$
2. from $R_{n\epsilon}$ generate an example dataset for training and testing
3. $R_{\text{org}}$ is used for establishing ground truth for labelling
4. learn an AdaBoost classifier for each missing feature to impute
5. predict the missing value using the feature classifier
6. observe the impact of *missingness* on the knowledge-based MDSS

repeated for top four $\epsilon$ values
Experimental Validation

Ex 3 – Hybrid Construction Evaluation

Our proposed hybrid construction outperforms both the knowledge-based and the nonknowledge-based (ML) decision support systems in the presence of noise. The performance of the knowledge-based solution degrades rapidly as noise is injected, while the machine learning-based DSS perform poorly independent of noise.
our proposed hybrid construction outperforms both the knowledge-based and the nonknowledge-based (ML) decision support systems in the presence of noise
our proposed hybrid construction outperforms both the knowledge-based and the nonknowledge-based (ML) decision support systems in the presence of noise. Performance of the knowledge-based solution degrades rapidly as noise is injected.
our proposed hybrid construction outperforms both the knowledge-based and the nonknowledge-based (ML) decision support systems in the presence of noise. Performance of the knowledge-based solution degrades rapidly as noise is injected. Machine learning based DSS perform poorly independent of noise.
Outline

1 Background
2 Architecture
3 Experimental Validation
4 Conclusion
Conclusion

Hybrid Construction for MDSS

- demonstrated the value of a hybrid MDSS that combines ontological and machine learning approaches on real-world datasets
- the hybrid construction fulfils all design goals
Future Work

1. **False Information**
   - missing vs. false information
     (what if the patient provides wrong details)
   - the hybrid construction fulfils all design goals

2. **Confidence Estimations**

3. **POC**
   - deployable implementation
Thank You!
References I


References II
