Constructing an Expert System Using Fuzzy Cognitive Mapping For Screening Self-Reported Multiple Sclerosis Patients
Multiple Sclerosis

One of several diseases known as “demyelinating” which is where the sheathing (myelin) around the nerve fibre deteriorates.

Characterized by both spine and brain lesions, in particular brain.

Key component is the “disseminating” component - lesions and attacks are separated in time and space.

Generally accepted four courses: “Recurring Remitting”, “Primary Progressive”, “Secondary Progressive”, “Progressive Relapsing”.

R-R is the most common, with many ending up Secondary Progressive. Primary Progressive is often fatal in 5 to 10 years.
Multiple Sclerosis

For a research project, Peter has proposed examining if there is “clustering” in MS patients, i.e., do patients share pairs of characteristics such as developing MS after both living in the same general area, having the same childhood illness, et al.

This data will be self-reported - large scale verification of the patient’s medical history will be difficult if not impossible.

I propose that we need an automated “expert” system that can screen the data and give a probability that the data is from a person with only MS.

The National Multiple Sclerosis reports one clustering study (Galion, Ohio) found 24% of patients (6) did not have MS.
Expert System

Use a standard criteria for diagnosing MS

Be “fuzzy” about it - no need for “74.3% probability of MS”
MacDonald et al (2001) established the accepted criteria for diagnosing multiple sclerosis (through consensus at the International Panel on MS Diagnosis)

Critical point - eliminate all other diseases BEFORE diagnosing “MS”

Laboratory tests - high emphasis on imaging (MRI) and cerebro-spinal fluid (CSF)

Clinical diagnosis - allowance of clinical assessment with specific criteria
“MacDonald Criteria”

How many other diseases have MS-like symptoms?

Leber’s hereditary optic neuropathy (LHON - Optic neuritis)
Transverse myelitis
Neuromyelitis Optica (Devic’s disease)
Acute disseminated encephalomyelitis (ADEM)
Acute haemorrhagic encephalomyelitis
Foix-Alajouanine syndrome
HTLV-1 (viral)
Adrenoleucodystrophy and metachromatic leucodystrophy
Systemic lupus erythematosus
Sjögren’s syndrome
Neurosarcoidosis
Behçet’s disease
Whipple’s disease
...

Wednesday, November 23, 2011
“MacDonald Criteria”

“For the purpose of diagnosing MS, CSF abnormality is defined (preferably using isoelectric focusing) by the presence of oligoclonal IgG bands different from any such bands in serum and/or the presence of an elevated IgG index.”

Q: Does the presence of oligoclonal banding (OCB) alone indicate MS?
“MacDonald Criteria”

“For the purpose of diagnosing MS, CSF abnormality is defined (preferably using isoelectric focusing) by the presence of oligoclonal IgG bands different from any such bands in serum and/or the presence of an elevated IgG index.”

Q: Does the presence of oligoclonal banding (OCB) alone indicate MS?

A: No. Mayringer, et al, (2005) found OCB occurred in

- 98.6% of MS patients
- 100% of Transverse Myelitis patients
- 94.1% of Optic Neuritis patients
“MacDonald Criteria”

“For the purpose of diagnosing MS, CSF abnormality is defined (preferably using isoelectric focusing) by the presence of oligoclonal IgG bands different from any such bands in serum and/or the presence of an elevated IgG index.”

Q: What is the impact of “an elevated IgG index?”
“MacDonald Criteria”

“For the purpose of diagnosing MS, CSF abnormality is defined (preferably using isoelectric focusing) by the presence of oligoclonal IgG bands different from any such bands in serum and/or the presence of an elevated IgG index.”

Q: What is the impact of “an elevated IgG index?”

A: Mayringer, et al, (2005) found “the frequency of OCB was nearly 100% in (Demyelinating Disease) ... irrespective of the IgG index.”
"MacDonald Criteria"

“For the purpose of diagnosing MS, CSF abnormality is defined (preferably using isoelectric focusing) by the presence of oligoclonal IgG bands different from any such bands in serum and/or the presence of an elevated IgG index.”

Q: What is the impact of “an elevated IgG index?”

A: Mayringer, et al. (2005) found “the frequency of OCB was nearly 100% in (Demyelinating Disease) ... irrespective of the IgG index.

...but they also found “the frequency of the diagnosis of DMD correlated positively with the IgG index, whereas there was a negative correlation between the IgG index and the frequency of all other diseases.”
"MacDonald Criteria"

“For the purpose of diagnosing MS, CSF abnormality is defined (preferably using isoelectric focusing) by the presence of oligoclonal IgG bands different from any such bands in serum and/or the presence of an elevated IgG index.”

\[ A = \text{OCB “AND” NOT Abnormal Blood serum} \]
For the purpose of diagnosing MS, CSF abnormality is defined (preferably using isoelectric focusing) by the presence of (oligoclonal IgG bands different from any such bands in serum) and/or the presence of an (elevated IgG index).

\[
A = \text{OCB "AND" NOT Abnormal Blood serum}
\]

\[
B = \text{Elevated IgG Index}
\]

\[
\text{Abnormal CSF} = A \text{ OR } B
\]
Combinations

**CSF:**
1) Oligoclonal banding
2) Lymphocytic Pleocytosis count < 50/mm$^3$
3) Elevated IgG Index
4) Normal blood serum
5) No raised neutrophil counts

**MRI:**
1) Brain lesions larger than 3mm
2) 1 Gadolinium-enhanced or 9 T2 hyperintensive lesions
3) 3 or more Periventricular lesions
4) 1 or more Infratentorial lesions
5) 1 or more Juxtacortical lesions
6) Eliminate age with presence of Vascular-Ischaemic lesions
Combinations

**CSF:**
1) Oligoclonal banding
2) Lymphocytic Pleocytosis count < 50/mm³
3) Elevated IgG Index
4) Normal blood serum
5) No raised neutrophil counts

**MRI:**
1) Brain lesions larger than 3mm
2) 1 Gadolinium-enhanced or 9 T2 hyperintensive lesions
3) 3 or more Periventricular lesions
4) 1 or more Infratentorial lesions
5) 1 or more Juxtacortical lesions
6) Eliminate age with presence of Vascular-Ischaemic lesions

3 of 4
Substitutions

**CSF:** Abnormal CSF = 1 brain lesion type

**MRI:** 1 spine lesion = 1 brain lesion type
Creating the Fuzzy Cognitive Map

Use social network analysis software (I used Gephi)

Each test is a node, and each edge represents a connection within a combination.
Creating the Fuzzy Cognitive Map code

1) Enter data

2) Evaluate

3) Return results
Creating the Fuzzy Cognitive Map code

1) construct the data structure

2) construct the enterable web form around the data structure

3) send the data to the web server/php script

2) write the logic statements based on the data structure

5) return the results
Creating the Fuzzy Cognitive Map code

1) construct the data structure

2) construct the enterable web form around the data structure

3) send the data to the web server/php script

2) write the logic statements based on the data structure

5) return the results
Creating the Fuzzy Cognitive Map code

Build a data structure around six “fuzzy” diagnosis:

“Definitely Not MS”
“Very Probably Not MS”
“Probably Not MS”

“Probably MS”
“Very Probably MS”
“Definitely MS”

Data is entered by test:

“MRI”  “CSF”  “Blood”  “Evoked Potential”
Creating the Fuzzy Cognitive Map code

Use PHP as scripting language

Built-in support for associative arrays

\$A[0] as well as \$A["Red"]

Create the structure as associative arrays:

\$EvaluationRound["Definitely MS"]
The Fuzzy Cognitive Map data structure -
data in

$LabTest["Blood"] = array("Present" => null, "Mitochondrial markers" => null, "NMOIgG" => null, "NormalBloodSerum" => null);

$LabTest["EvokedPotential"] = array("Present" => null, "NormalEPs" => null, "AbnormalOptical" => null, "AbnormalAuditory" => null, "AbnormalSensory" => null);

$LabTest["CSF"] = array("Present" => null, "NormalCSF" => null, "OligoclonalBanding" => null, "LymphociticPleocytosisLT50" => null, "ElevatedIgGIndex" => null, "RaisedNeutrophilCount" => null);

$LabTest["MRI"] = array("Present" => null, "NormalBrainMRI" => null, "NormalSpineMRI" => null,
        "BrainLesions" => array("Present" => null, "T2Hyperintensive" => null, "GadoliniumEnhanced" => null, "MoreThan2Periventricular" => null, "Infratentorial" => null, "Juxtacortical" => null, "VascularIschaemic" => null),
        "SpineLesions" => array("Present" => null, "MTypeSpinalCordLesions" => false, "MoreThan2AsymptomaticMTypeSpinalCordLesions" => null, "TransverseMyelitis" => null),
    );
The Fuzzy Cognitive Map data structure - data in

$\text{LabTest["Blood"] = array("Present" => null, "Mitochondrial markers" => null, "NMOIgG" => null, "NormalBloodSerum" => null);}
Creating the Fuzzy Cognitive Map code

1) construct the data structure

2) construct the enterable web form around the data structure

3) send the data to the web server/php script

4) write the logic statements based on the data structure

5) return the results
The Fuzzy Cognitive Map data structure - web form

<tr>
<td>
Were IgG antibodies for Neuromyelitis Optica (NMO) present? </td>
</tr>
<tr>
<td>
<input type="radio" name="LabTest99Blood99NMOIgG" value="true" /> Yes
<input type="radio" name="LabTest99Blood99NMOIgG" value="false" /> No
<input type="radio" name="LabTest99Blood99NMOIgG" value="U" CHECKED > Unknown
</td>
</tr>
The Fuzzy Cognitive Map data structure - web form

Use AJAX (Javascript) to send data to PHP script

```javascript
function SubmitData(dataForm) {

    var memo = "submit_data";
    var url="MS_interface.php";
    var data = "";

    for (d in dataForm) {

        if ((dataForm[d] !== null)
            && (dataForm[d].checked === true)
            && (dataForm[d].value != "U")) {

            data=data+dataForm[d].name+"="+dataForm[d].value+"&";
        }
    }

    xmlreqPOST(url, data, memo);
}
```
The Fuzzy Cognitive Map data structure - web form

<table>
<thead>
<tr>
<th>Patient Information</th>
<th>Less than 60 years of age</th>
<th>60 or older</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>○</td>
<td>□</td>
<td>☐</td>
</tr>
<tr>
<td>Female</td>
<td>□</td>
<td>○</td>
<td>☐</td>
</tr>
<tr>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Have blood tests been done?</th>
<th>Did the blood test show markers for mitochondrial disease?</th>
<th>Yes</th>
<th>No</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Have Evoked Potentials been done?</th>
<th>Were the EPs normal?</th>
<th>Yes</th>
<th>No</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Has the patient's CSF been tested?</th>
<th>Was the CSF normal?</th>
<th>Yes</th>
<th>No</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Has the patient had an MRI?</th>
<th>Was the brain MRI normal?</th>
<th>Yes</th>
<th>No</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Unknown</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Were there brain lesions &gt;3mm in diameter?</th>
<th>Yes</th>
<th>No</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2 Hyperintensive Lesions?</td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Gadolinium-enhanced Lesions?</td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>More Than 2 Periventricular Lesions?</td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Infratentorial Lesions?</td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Juxtacortical Lesions?</td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Vascular Ischaemic Lesions?</td>
<td></td>
<td></td>
<td>Unknown</td>
</tr>
</tbody>
</table>

| Were spinal lesions present? | Yes | No | Unknown |
|================--------------|-----|----|---------|
| Were they MS-type Spinal Cord Lesions? | Yes | No | Unknown |
| Were there more than 2 asymptomatic MS-type Spinal Cord Lesions? | Yes | No | Unknown |
Creating the Fuzzy Cognitive Map code

1) construct the data structure

2) construct the enterable web form around the data structure

3) send the data to the web server/php script

4) write the logic statements based on the data structure

5) return the results
Creating the Fuzzy Cognitive Map code

1) Enter data

2) Evaluate
The Fuzzy Cognitive Map data structure - PHP script

Data arrives inside the global variable $_REQUEST


[LabTest99Blood99NMOIgG] is parsed to locate the element

“99” is the demarcation

LabTest99Blood99NMOIgG = $LabTest[“Blood”][“NMOIgG”]
include_once "MS_DataStructure.php";

function &dive($arry, $element) {
    $a = &$arry[$element];
    return $a;
}

foreach ($_REQUEST as $k => $v) {
    $arry = explode("99", $k);
    $n = count($arry);
    $d = $arry[0];
    if ($d == "LabTest")
        $test = &LabTest;
    elseif ($d == "ClinicalDiagnosis")
        $test = &ClinicalDiagnosis;
    elseif ($d == "Patient")
        $test = &Patient;
    else
        unset($test);
    for ($i = 1; $i < ($n-1); $i++) {
        $test = &dive(&$test, $arry[$i]);
    }
    if ($v != "U")
        $test[$arry[$n-1]] = ($v == "true");
}

include_once "MSMap_Rules.php";
Creating the Fuzzy Cognitive Map code

1) construct the data structure

2) construct the enterable web form around the data structure

3) send the data to the web server/php script

4) write the logic statements based on the data structure

5) return the results
Creating the Fuzzy Cognitive Map code

Rather than lots and lots of “if..then” statements, how about “coalescing” the diagnosis?

Feed in all of the data at one time, and then look for “true” conditions at the end.

\[
C = A \text{ and } B \\
E = C \text{ and } D \\
F = C \text{ or } E
\]

is F true or false?
Creating the Fuzzy Cognitive Map code

Think of it like an electronic circuit - put in the inputs, let the circuits do their AND and OR, then look at the output.
Creating the Fuzzy Cognitive Map code

Use PHP as scripting language

Built-in support for associative arrays

Interpreted language - data structures can be built dynamically
include_once "MS_DataStructure.php";

function &dive($arry, $element) {
    $a = &$arry[$element];
    return $a;
}

foreach ($_REQUEST as $k => $v) {
    $arry = explode("99", $k);
    $n = count($arry);

    $d = $arry[0];
    if ($d == "LabTest")
        $test = &$LabTest;
    elseif ($d == "ClinicalDiagnosis")
        $test = &$ClinicalDiagnosis;
    elseif ($d == "Patient")
        $test = &$Patient;
    else
        unset($test);

    for ($i = 1; $i < ($n-1); $i++) {
        $test = &dive(&$test, $arry[$i]);
    }

    if ($v != "U")
        $test[$arry[$n-1]] = ($v == "true");
}

include_once "MSMap_Rules.php";
Creating the Fuzzy Cognitive Map code

How do we handle missing data/unknown test results?

Use three state logic - True, False and Unknown
# Ternary (Three state) Logic

<table>
<thead>
<tr>
<th>A AND B</th>
<th>TRUE</th>
<th>FALSE</th>
<th>UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>UNK</td>
</tr>
<tr>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>UNK</td>
<td>UNK</td>
<td>FALSE</td>
<td>UNK</td>
</tr>
</tbody>
</table>

Ternary (Three state) Logic

<table>
<thead>
<tr>
<th>A OR B</th>
<th>TRUE</th>
<th>FALSE</th>
<th>UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>UNK</td>
</tr>
<tr>
<td>UNK</td>
<td>TRUE</td>
<td>UNK</td>
<td>UNK</td>
</tr>
</tbody>
</table>
Ternary (Three state) Logic

<table>
<thead>
<tr>
<th>A</th>
<th>NOT A</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>FALSE</td>
</tr>
<tr>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td>UNK</td>
<td>UNK</td>
</tr>
</tbody>
</table>

http://en.wikipedia.org/wiki/Ternary_logic
Creating the Fuzzy Cognitive Map code

- Use PHP as scripting language
- Built-in support for associative arrays
- Interpreted language - data structures can be built dynamically
- Built-in support for ternary logic - true, false, null
// ReturnTernaryValue
function RTV($A, $operation, $B = null) {
    if ($operation == "OR") {
        if ($A === true)
            return true;
        elseif ($B === true)
            return true;
        elseif (($A === false) && ($B === false))
            return false;
        else  // only thing left is $A or $B === unknown/null
            return null;
    }
    elseif ($operation == "AND") {
        if ($A === false)
            return false;
        elseif ($B === false)
            return false;
        elseif (($A === true) && ($B === true))
            return true;
        else  // only thing left is $A or $B === unknown/null
            return null;
    }
    elseif ($operation == "NOT") {
        // ignore $B
        if ($A === false)
            return true;
        elseif ($A === true)
            return false;
        else
            return $A;
    }
    else
        return null;
}
Recall our original fuzzy outcomes:

"Definitely Not MS"
"Very Probably Not MS"
"Probably Not MS"

"Probably MS"
"Very Probably MS"
"Definitely MS"

These are represented as array elements, i.e.,

$EvaluationRound["Definitely Not MS"]
The Fuzzy Cognitive Map data structure - PHP script

$EvaluationRound["Definitely Not MS"]
$EvaluationRound["Very Probably Not MS"]
$EvaluationRound["Probably Not MS"]

$EvaluationRound["Probably MS"]
$EvaluationRound["Very Probably MS"]
$EvaluationRound["Definitely MS"]

Time to build the logic statements...
The Fuzzy Cognitive Map data structure - PHP script

A problem, though -

PHP can not refer to other elements of an array while constructing the array

In the case of combinations, those outcomes have to be calculated one or more steps before they are referenced.

Each $EvaluationRound structure needs three substructures:

"Dependencies", "Combinations", and "Diagnostics"

Combinations can reference Dependencies from that EvaluationRound or any substructure from an earlier EvaluationRound
### The Fuzzy Cognitive Map data structure - PHP script

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Combinations</th>
<th>Diagnostics</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Definitely Not MS&quot;</td>
<td>&quot;Definitely MS&quot;</td>
<td>&quot;Definitely Not MS&quot;</td>
<td>&quot;Definitely MS&quot;</td>
</tr>
<tr>
<td>&quot;Probably Not MS&quot;</td>
<td>&quot;Very Probably Not MS&quot;</td>
<td>&quot;Very Probably Not MS&quot;</td>
<td>&quot;Probable MS&quot;</td>
</tr>
<tr>
<td>&quot;Very Probably MS&quot;</td>
<td>&quot;Definitely MS&quot;</td>
<td>&quot;Definitely MS&quot;</td>
<td>&quot;Definitely MS&quot;</td>
</tr>
</tbody>
</table>
The Fuzzy Cognitive Map data structure - PHP script

$EvaluationRound[“Definitely Not MS”]

“Dependencies”
“Combinations”
“Diagnostics”
“Outcome”

$EvaluationRound[“Very Probably Not MS”]

“Dependencies”
“Combinations”
“Diagnostics”
“Outcome”

$EvaluationRound[“Probably Not MS”]

“Dependencies”
“Combinations”
“Diagnostics”
“Outcome”

$EvaluationRound[“Definitely MS”]

“Dependencies”
“Combinations”
“Diagnostics”
“Outcome”

$EvaluationRound[“Very Probably MS”]

“Dependencies”
“Combinations”
“Diagnostics”
“Outcome”

$EvaluationRound[“Probably MS”]

“Dependencies”
“Combinations”
“Diagnostics”
“Outcome”
The Fuzzy Cognitive Map data structure - PHP script

$EvaluationRound["Definitely Not MS"]
  "Dependencies"
  "Combinations"
  "Diagnostics"
  "Outcome"

$EvaluationRound["Very Probably Not MS"]
  "Dependencies"
  "Combinations"
  "Diagnostics"
  "Outcome"

$EvaluationRound["Probably Not MS"]
  "Dependencies"
  "Combinations"
  "Diagnostics"
  "Outcome"

$EvaluationRound["Definitely MS"]
  "Dependencies"
  "Combinations"
  "Diagnostics"
  "Outcome"

$EvaluationRound["Very Probably MS"]
  "Dependencies"
  "Combinations"
  "Diagnostics"
  "Outcome"

$EvaluationRound["Probably MS"]
  "Dependencies"
  "Combinations"
  "Diagnostics"
  "Outcome"
The Fuzzy Cognitive Map data structure - PHP script

$EvaluationRound["Definitely Not MS"]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$EvaluationRound["Very Probably Not MS"]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$EvaluationRound["Probably Not MS"]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$EvaluationRound["Definitely MS"]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$EvaluationRound["Very Probably MS"]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$EvaluationRound["Probably MS"]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

Wednesday, November 23, 2011
$EvaluationRound["Definitely Not MS"]

"Dependencies"
"Combinations"
"Diagnostics"
"Outcome"

$EvaluationRound["Very Probably Not MS"]

"Dependencies"
"Combinations"
"Diagnostics"
"Outcome"

$EvaluationRound["Probably Not MS"]

"Dependencies"
"Combinations"
"Diagnostics"
"Outcome"

$EvaluationRound["Definitely MS"]

"Dependencies"
"Combinations"
"Diagnostics"
"Outcome"

$EvaluationRound["Very Probably MS"]

"Dependencies"
"Combinations"
"Diagnostics"
"Outcome"

$EvaluationRound["Probably MS"]

"Dependencies"
"Combinations"
"Diagnostics"
"Outcome"
The Fuzzy Cognitive Map data structure - PHP script

$evaluationRound[“Definitely Not MS”]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$evaluationRound[“Very Probably Not MS”]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$evaluationRound[“Probably Not MS”]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$evaluationRound[“Definitely MS”]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$evaluationRound[“Very Probably MS”]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”

$evaluationRound[“Probably MS”]
  “Dependencies”
  “Combinations”
  “Diagnostics”
  “Outcome”
Combinations

CSF:
1) Oligoclonal banding
2) Lymphocitic Pleocytosis count < 50/mm³
3) Elevated IgG Index
4) Normal blood serum
5) No raised neutrophil counts

MRI:
1) Brain lesions larger than 3mm
2) 1 Gadolinium-enhanced or 9 T2 hyperintensive lesions
3) 3 or more Periventricular lesions
4) 1 or more Infratentorial lesions
5) 1 or more Juxtacortical lesions
6) Eliminate age with presence of Vascular-Ischaemic lesions
Counting with Ternary Logic
“There are 10 kinds of people in the world. Those that understand binary, and those that don’t.”
Counting with Ternary Logic

“There are 10 kinds of people in the world. Those that understand binary, and those that don’t.”

Q: How does one count using logic statements?
Counting with Ternary Logic

“There are 10 kinds of people in the world. Those that understand binary, and those that don’t.”

Q: How does one count using logic statements?

A: Slowly.
Counting with Ternary Logic

How do we find 3 of 4 using logic?
Counting with Ternary Logic

How do we find 3 of 4 using logic?

Turn it into binary statements:

\[ A \text{ AND } B = 0 \text{ or } 2 \]
\[ A \text{ OR } B = 0 \text{ or } 1 \]
Counting with Ternary Logic

How do we find 3 of 4 using logic?

Turn it into binary statements:

\[ A \text{ AND } B = 0 \text{ or } 2 \]
\[ A \text{ OR } B = 0 \text{ or } 1 \]

\[ 1011 = 1101 = 3 \text{ of } 4 \]
Counting with Ternary Logic

How do we find 3 of 4 using logic?

Turn it into binary statements:

\[
\begin{align*}
A \text{ AND } B &= 0 \text{ or } 2 \\
A \text{ OR } B &= 0 \text{ or } 1
\end{align*}
\]

\[
1011 = 1101 = 3 \text{ of } 4
\]

\[
0111 = 1110 = 3 \text{ of } 4
\]
How do we find 3 of 4 using logic?

Turn it into binary statements:

A AND B = 0 or 2
A OR B = 0 or 1

1011 = 1101 = 3 of 4
0111 = 1110 = 3 of 4
Counting with Ternary Logic

How do we find 3 of 4 using logic?

Turn it into binary statements:

A AND B = 0 or 2
A OR B = 0 or 1

\[ \begin{align*}
1011 &= 1101 = 3 \text{ of } 4 \\
0111 &= 1110 = 3 \text{ of } 4
\end{align*} \]
Counting with Ternary Logic

Gadolinium-enhanced or >9 T2 Hyperintensive = A
>2 Periventricular lesions = B
>0 Infratentorial lesions = C
>0 Juxtacortical lesions = D

“(A OR B) AND (C AND D)
OR
(A AND B) AND (C OR D)”

7 logic statements to find “3 of 4”
$EvaluationRound["Probably Not MS"] = array(

    // these can be true or false, and are needed by Combinations
    "Dependencies" => array(

        "ElevatedIgGIndex" => RTV($LabTest["CSF"]["ElevatedIgGIndex"], "AND", true),

        ">2BrainLesionTypes" => RTV($EvaluationRound["Very Probably Not MS"]
        ["Combinations"]["P>2&Infra&T2ORJ"], "OR", $EvaluationRound["Very Probably Not
        MS"]["Combinations"]["T2&J&P>2ORInfra"],

        "MSAbnormalEP" => RTV(RTV($LabTest["EvokedPotential"]["AbnormalOptical"],
        "OR", $LabTest["EvokedPotential"]["AbnormalAuditory"], "OR",
        $LabTest["EvokedPotential"]["AbnormalSensory"],

        "MSAbnormalCSF" => RTV(RTV($EvaluationRound["Very Probably Not MS"]
        ["Combinations"]["OB&LP<50"], "AND",$LabTest["Blood"]["NormalBloodSerum"],
        "AND", $LabTest["CSF"]["ElevatedIgGIndex"]),

        "MSAbnormalMRI" => RTV($EvaluationRound["Definitely Not MS"]["Combinations"]
        ["T2OrG-eBrainLesionORJuxtacortical"], "OR", $EvaluationRound["Definitely Not
        MS"]["Combinations"]["MoreThan2PeriventricularORInfratentorial"]),

    )
)
The Fuzzy Cognitive Map data structure - data in

">2BrainLesionTypes" => RTV($EvaluationRound["Very Probably Not MS"] ["Combinations"] ["P>2&Infra&T2ORJ"], "OR", $EvaluationRound["Very Probably Not MS"] ["Combinations"] ["T2&J&P>2ORInfra"])

Wednesday, November 23, 2011
The Fuzzy Cognitive Map data structure - PHP script

// one or more of these have to be true
$EvaluationRound["Very Probably Not MS"]['Diagnostics'] = array(
    "Transverse Myelitis & Normal Brain MRI" => RTV($LabTest["MRI"]['SpineLesions']["TransverseMyelitis"], "AND", $LabTest["MRI"]['NormalBrainMRI']),
);
The Fuzzy Cognitive Map data structure - PHP script

(Number entries/Logic statements)

$EvaluationRound["Definitely Not MS"]

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Combinations</th>
<th>Diagnostics</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>(7/10)</td>
<td>(7/9)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

$EvaluationRound["Very Probably Not MS"]

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Combinations</th>
<th>Diagnostics</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>(4/5)</td>
<td>(2/2)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

$EvaluationRound["Probably Not MS"]

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Combinations</th>
<th>Diagnostics</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5/7)</td>
<td>(7/10)</td>
<td>(4/4)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

$EvaluationRound["Definitely MS"]

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Combinations</th>
<th>Diagnostics</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>(5/7)</td>
<td>(2/2)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

$EvaluationRound["Very Probably MS"]

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Combinations</th>
<th>Diagnostics</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>(3/3)</td>
<td>(1/2)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

$EvaluationRound["Probably MS"]

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Combinations</th>
<th>Diagnostics</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>(2/5)</td>
<td>(2/2)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

56 entries/74 logic statements
function Diagnose($arry) {
    $a = null; $v = null;
    foreach($arry as $k =>$v)
        $a = RTV($a, "OR", $v);
    unset($v);
    return $a;
}

$EvaluationRound["Definitely MS"]["Outcome"] = Diagnose($EvaluationRound["Definitely MS"]["Diagnostics"]);

if $EvaluationRound["Very Probably Not MS"]["Outcome"]
evaluates to "true" then it's very probably not MS
The Fuzzy Cognitive Map data structure - PHP script

Examine the Evaluation Rounds in a specific order, attempting to eliminate MS first, and then determine the highest probability of MS

"Definitely Not MS"
"Very Probably Not MS"
"Probably Not MS"

"Definitely MS"
"Very Probably MS"
"Probably MS"
The Fuzzy Cognitive Map data structure - PHP script

```php
$Outcome = array(
    "Definitely Not MS" => $EvaluationRound["Definitely Not MS"],
    "Very Probably Not MS" => $EvaluationRound["Very Probably Not MS"],
    "Probably Not MS" => $EvaluationRound["Probably Not MS"],
    "Definitely MS" => $EvaluationRound["Definitely MS"],
    "Very Probably MS" => $EvaluationRound["Very Probably MS"],
    "Probably MS" => $EvaluationRound["Probably MS"]
);

$resolution = false;
foreach ($Outcome as $round => $criteria) {
    if ($criteria["Outcome"] === true) {
        $resolution = true;
        print($round . " : ");
        foreach ($criteria["Diagnostics"] as $d => $v)
            if ($criteria["Diagnostics"][$d] === true)
                print($d . "<br />
        break;
    }
}
if ($resolution === false)
    print("Indeterminate outcome - not enough information.<br />");
```
The Fuzzy Cognitive Map data structure - PHP script

To get to Definitely MS you need MRI, CSF, EP and Blood work

How close can you get with more than one missing test results?

“Probably MS” w/o MRI, “Very Probably MS” with it

How close can you get with almost all of the test results except blood work?

“Very Probably MS” for males (need NMO-IgG)
“Very Probably MS” for females (need mitochondrial test & NMO-IgG)
Creating the Fuzzy Cognitive Map code

1) construct the data structure

2) construct the enterable web form around the data structure

3) send the data to the web server/php script

4) write the logic statements based on the data structure

5) return the results
Creating the Fuzzy Cognitive Map code

2) Evaluate

3) Return results
The Fuzzy Cognitive Map data structure - web form

Use AJAX (Javascript) to receive data from PHP script

```javascript
function handle_response(xmlhttp) {
    document.getElementById("diagnosis").innerHTML=xmlhttp.responseText;
}
```
Creating the Fuzzy Cognitive Map code

2) Evaluate

3) Return results