Human Computation

Core Research Questions and State of the Art

part 1

Edith Law
Carnegie Mellon University
Human Computation

in a nutshell
“Some problems are hard, even for the most sophisticated AI algorithms.”
Human Computation
in a nutshell

“Some problems are hard, even for the most sophisticated AI algorithms.”

“Let humans solve it ...”
Human Computation

you are a human computer
The Norwich line steamboat train, from New-London for Boston, this morning ran off the track seven miles north of New-London.

Type the two words:

morning overlooks
Human Computation is an old idea.
The Past

a very brief history

Halley Comet
1750’s

source: the Yerkes Observatory

“When Computers were Human” by David A. Grier
The Past
a very brief history

Halley Comet
1750’s

source: the Yerkes Observatory

Nautical Almanac
1770’s

“When Computers were Human” by David A. Grier
The Past

a very brief history

Halley Comet
1750's

source: the Yerkes Observatory

Nautical Almanac
1770's

Math Tables Project
1930's

courtesy: David A. Grier

“When Computers were Human” by David A. Grier
The Past
a very brief history

Halley Comet
1750's

source: the Yerkes Observatory

Babbage
Difference Machine
1820's

Nautical Almanac
1770's

source: the Yerkes Observatory

Hollerith
Machine
1890's

source: IBM

Math Tables
Project
1930's

courtesy: David A. Grier

“When Computers were Human” by David A. Grier
The Past
a very brief history

- **Halley Comet**
  - 1750's
  - Source: the Yerkes Observatory

- **Nautical Almanac**
  - 1770's

- **Babbage Difference Machine**
  - 1820's

- **Telegraph & Weather**
  - 1850's
  - Source: IBM

- **Hollerith Machine**
  - 1890's
  - Courtesy: David A. Grier

- **Math Tables Project**
  - 1930's
  - Courtesy: David A. Grier

"When Computers were Human" by David A. Grier
The Web changed everything.
The Present scale

(von Ahn and Dabbish, 2004)
The Present
reach
The Present
reach
The Present
reach
The Present
reach
The Present
reach
The Present

pervasiveness
Human Computation

a growing field

1st Human Computation Workshop  KDD 2009
Crowdsourcing for Search Evaluation  SIGIR 2010
2nd Human Computation Workshop  KDD 2010
  Advancing Computer Vision with Humans in the Loop  CVPR 2010
  Creating Speech and Language Data with Amazon’s Mechanical Turk  NAACL 2010
  Computational Social Science and Wisdom of the Crowds  NIPS 2010
  Workshop on Ubiquitous Crowdsourcing  Ubicomp 2010
  Enterprise Crowdsourcing Workshop  ICWE 2010
Collaborative Translation Technology, Crowdsourcing and the Translator  AMTA 2010
Crowdsourcing for Search and Data Mining  WSDM 2010
Workshop on Crowdsourcing for Information Retrieval  SIGIR 2011
Workshop on Social Computing and User Generated Content  EC 2011
Workshop on Crowdsourcing and Human Computation  CHI 2011
3rd Human Computation Workshop  AAAI 2011

Mechanical Turk for Computer Vision  CVPR 2010
Crowdsourcing for Relevance Evaluation  ECIR 2010
Managing Crowdsourced Human Computation  WWW 2011
Crowdsourcing 101: Putting the WSDM of Crowds to Work for You  WSDM 2011
Crowdsourcing Applications and Platforms  VLDB 2011
Crowdsourcing for Information Retrieval: Principles, Methods and Applications  SIGIR 2011
Quality Crowdsourcing for Human Computer Interaction Research  HCIC 2011
Crowdsourcing for Fun and Profit  CrowdConf 2011
Human Computation: Core Research Questions and State of the Art  AAAI 2011
Human Computation
multi-disciplinary

HUMAN COMPUTATION

- Machine Learning
- Psychology
- Economics
- Statistics
- Social Science
- Theory
- Artificial Intelligence
- Mechanism Design
Introduce a framework for human computation with a set of concepts, core research questions, existing work and open problems.
Introduce a framework for human computation with a set of concepts, core research questions, existing work and open problems.
# Tutorial

**a rough schedule**

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>DESIGN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>part 1</strong></td>
<td><strong>part 2</strong></td>
</tr>
<tr>
<td>2:00-3:30</td>
<td>4:00-5:30</td>
</tr>
<tr>
<td>Edith Law</td>
<td>Luis von Ahn</td>
</tr>
</tbody>
</table>
A FRAMEWORK FOR HUMAN COMPUTATION

Concepts • Scope
Computation

a general definition

The process of mapping input to output.
Computational Problems

examples

multiplication
  two numbers ➔ product

sorting
  set of objects ➔ set of objects sorted

medical diagnosis
  x-ray, lab tests ➔ diagnosis

object recognition
  image ➔ tag

translation
  source sentence ➔ target sentence

editing
  text ➔ corrected text

planning
  goal, constraints ➔ sequence of actions
Human Computation

a general definition

Computation that is carried out by a human.
Human Computation

problem statement
Given a computational problem, design a solution using human computers and automated computers.
Related Concepts
definitions
The shared or group intelligence that emerges from the collaboration and competition of many individuals (bacteria, animals, humans, computer agents).
Related Concepts

definitions

**COLLECTIVE INTELLIGENCE**

The shared or group intelligence that emerges from the collaboration and competition of many individuals (bacteria, animals, humans, computer agents).

**SOCIAL COMPUTING**

Technology for supporting social behavior and interactions (e.g., blog, email, Instant messaging) or group computation (e.g., collaborative filtering, auctions, prediction markets).

<table>
<thead>
<tr>
<th>Related Concepts definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COLLECTIVE INTELLIGENCE</strong></td>
</tr>
<tr>
<td>The shared or <strong>group intelligence</strong> that emerges from the collaboration and competition of many individuals (bacteria, animals, humans, computer agents).</td>
</tr>
<tr>
<td><strong>SOCIAL COMPUTING</strong></td>
</tr>
<tr>
<td>Technology for supporting <strong>social behavior and interactions</strong> (e.g., blog, email, Instant messaging) or <strong>group computation</strong> (e.g., collaborative filtering, auctions, prediction markets).</td>
</tr>
<tr>
<td><strong>CROWDSOURCING</strong></td>
</tr>
<tr>
<td>Outsourcing tasks through an <strong>open call</strong>.</td>
</tr>
</tbody>
</table>
Collective Intelligence

Human Computation

Social Computing

Crowdsourcing

Related Concepts boundaries
Three Distinguishing Features of human computation
Three Distinguishing Features of human computation
Three Distinguishing Features of human computation

“Human” In The Loop
not bacteria, not ants, not fish.
Three Distinguishing Features of human computation

“Human” In The Loop
not bacteria, not ants, not fish.
Three Distinguishing Features
of human computation

“Human” In The Loop
not bacteria, not ants, not fish.

Conscious Effort
humans are actively computing something, not merely
carrier of sensors and computational devices.
Three Distinguishing Features
of human computation

“Human” In The Loop
not bacteria, not ants, not fish.

Conscious Effort
humans are actively computing something, not merely
carrier of sensors and computational devices.
Three Distinguishing Features of human computation

“Human” In The Loop
not bacteria, not ants, not fish.

Conscious Effort
humans are actively computing something, not merely carrier of sensors and computational devices.

Explicit Control
the outcome of the computation is determined by an algorithm, and not the natural dynamics of the crowd.
Core Research Questions

“What”, “who”, “how”
Core Research Questions
from the definition

Given a computational problem, design a solution using human computers and automated computers.
Core Research Questions
“what”, “who”, “how”
Core Research Questions
from the definition

“How hard is the problem? Is it efficiently solvable?”

Given a computational problem, design a solution using human computers and automated computers.
Core Research Questions from the definition

“Is the human computation algorithm correct and efficient?”

Given a computational problem, design a solution using human computers and automated computers.
Given a computational problem, design a solution using human computers and automated computers.

“How do we aggregate the outputs of many human computers?”
Core Research Questions
from the definition

Given a computational problem, design a solution using human computers and automated computers.

“How to make the tradeoff between human versus machine?”
Core Research Questions

“what”, “who”, “how”
Core Research Questions

from the definition

“To whom do we route each task, and how?”

Given a computational problem, design a solution using human computers and automated computers.
Core Research Questions
"what", "who", "how"
Core Research Questions
from the definition

Given a computational problem, design a solution using human computers and automated computers.

“How to design tasks, motivate participation and incentivize truthful outputs?”
Core Research Questions from the definition

“How to meet the needs and wants of the requesters?”

Given a computational problem, design a solution using human computers and automated computers.
Core Research Questions

“what”, “who”, “how”
HUMAN COMPUTATION ALGORITHMS

Definition • Properties
What are algorithms?
Algorithms

an example

function **quicksort**(A)
   initialize empty lists L and G
   if (length(A) ≤ 1)
      return A
   pivot = A.remove(find_pivot(A));
   for x in A
      if compare(x, pivot)
         L.add(x)
      else
         G.add(x)
   return concatenate(quicksort(L), pivot, quicksort(G))

function **pivot**(A)
   return randomIndex(A);

function **compare**(x, pivot)
   return (x < pivot)
function `quicksort`(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function `pivot`(A)
    return randomIndex(A);

function `compare`(x, pivot)
    return (x < pivot)
an example

function **quicksort**(*A*)
    initialize empty lists *L* and *G*
    if (length(*A*) ≤ 1)
        return *A*
    pivot = *A*.remove(find_pivot(*A*));
    for *x* in *A*
        if compare(*x*, pivot)
            *L*.add(*x*)
        else
            *G*.add(*x*)
    return concatenate(quicksort(*L*), pivot, quicksort(*G*))

function **pivot**(*A*)
    return randomIndex(*A*);

function **compare**(*x*, pivot)
    return (*x* < pivot)
function **quicksort**\( (A) \)

initialize empty lists \( L \) and \( G \)
if \( \text{length}(A) \leq 1 \)  
return \( A \)

\( \text{pivot} = A.\text{remove}(\text{find_pivot}(A)) \);
for \( x \) in \( A \)
    if \( \text{compare}(x, \text{pivot}) \)
        \( L.\text{add}(x) \)
    else
        \( G.\text{add}(x) \)
return \( \text{concatenate}(\text{quicksort}(L), \text{pivot}, \text{quicksort}(G)) \)

function **pivot**\( (A) \)
return \( \text{randomIndex}(A) \);
Algorithms

an example

function \textbf{quicksort}(A)
\hspace{1em} initialize empty lists L and G
\hspace{1em} if \((\text{length}(A) \leq 1))
\hspace{1em} \hspace{1em} \text{return } A
\hspace{1em} pivot = A.remove(\text{find\_pivot}(A));
\hspace{1em} \text{for } x \text{ in } A
\hspace{1.5em} \text{if } \text{compare}(x, \text{pivot})
\hspace{2em} \hspace{1em} \text{L.add}(x)
\hspace{1.5em} \text{else}
\hspace{2em} \hspace{1em} \text{G.add}(x)
\hspace{1em} \text{return concatenate} (\text{quicksort}(L), \text{pivot}, \text{quicksort}(G))

function \textbf{pivot}(A)
\hspace{1em} \text{return } \text{randomIndex}(A);

function \textbf{compare}(x, \text{pivot})
\hspace{1em} \text{return } (x < \text{pivot})
function **quicksort** (A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function **pivot** (A)
    return randomIndex(A);

function **compare** (x, pivot)
    return (x < pivot)
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

**Input**, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, **Finiteness**, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973
What are human computation algorithms?
function `quicksort(A)`
  initialize empty lists L and G
  if (length(A) ≤ 1)
    return A
  pivot = A.remove(find_pivot(A));
  for x in A
    if compare(x, pivot)
      L.add(x)
    else
      G.add(x)
  return concatenate(quicksort(L), pivot, quicksort(G))

function `pivot(A)`
  return randomIndex(A);

function `compare(x, pivot)`
  return (x < pivot)
Human Computation Algorithms
human-driven operation

function **quicksort**(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function **pivot**(A)
    return randomIndex(A);
function **quicksort**(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function **pivot**(A)
    return randomIndex(A);

function **compare**(x, pivot)
    return human_compare(x, pivot)
Human Computation Algorithms
human-driven operation

function **quicksort**(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function **pivot**(A)
    return randomIndex(A);

function **compare**(x, pivot)
    return human_compare(x, pivot)
function **quicksort**\(A\)
   initialize empty lists \(L\) and \(G\)
   if \((\text{length}(A) \leq 1)\)
      return \(A\)
   \(\text{pivot} = A\).remove(find_pivot(A));
   for \(x\) in \(A\)
      if compare\((x, \text{pivot})\)
         \(L\).add\((x)\)
      else
         \(G\).add\((x)\)
   return concatenate\((\text{quicksort}(L), \text{pivot}, \text{quicksort}(G))\)

function **pivot**\(A\)
   return randomIndex\(A\);

function **compare**\((x, \text{pivot})\)
   return **human_compare**\((x, \text{pivot})\)
function \texttt{quicksort}(A) 
  initialize empty lists \( L \) and \( G \) 
  if \((\text{length}(A) \leq 1)\) 
    return \( A \) 
  pivot = \( A.\text{remove}(\text{find\_pivot}(A)) \); 
  for \( x \) in \( A \) 
    if compare\((x, \text{pivot})\) 
      \( L.\text{add}(x) \) 
    else 
      \( G.\text{add}(x) \) 
  return concatenate(\text{quicksort}(L), \text{pivot}, \text{quicksort}(G)) 

function \texttt{pivot}(A) 
  return \text{randomIndex}(A); 

function \texttt{compare}(x, \text{pivot}) 
  return \texttt{human\_compare}(x, \text{pivot}) 

Mechanical Turk Task

\textbf{Instructions}
You are shown two images. You must select the image that is more indicative of suspicious activities.

\textbf{Task}
Imagine that you are a security guard and you are monitoring two places. Someone informed you that there are suspicious activities in one of the places, but you were not told which one. Which place will you attend to? 

\text{Submit}
Human Computation Algorithms
human-driven operation

function **quicksort**\(A\)
  initialize empty lists \(L\) and \(G\)
  if \((\text{length}(A) \leq 1)\)
    return \(A\)
  pivot = \(A\).remove(find\_pivot\((A)\));
  for \(x\) in \(A\)
    if compare\((x, \text{pivot})\)
      \(L\).add\((x)\)
    else
      \(G\).add\((x)\)
  return concatenate\((\text{quicksort}(L), \text{pivot}, \text{quicksort}(G))\)

function **pivot**\(A\)
  return random\_index\((A)\);

function **compare**\((x, \text{pivot})\)
  return human\_compare\((x, \text{pivot})\)

TurKit (Little et al., 2010); Boto (http://code.google.com/p/boto/)
function **quicksort**(A)
    initialize empty lists L and G
    if (length(A) \(\leq 1\))
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function **pivot**(A)
    return randomIndex(A);

function **compare**(x, pivot)
    return **human_compare**(x, pivot)
function **quicksort**(A)
  initialize empty lists L and G
  if (length(A) ≤ 1)
    return A
  pivot = A.remove(find_pivot(A));
  for x in A
    if compare(x, pivot)
      L.add(x)
    else
      G.add(x)
  return concatenate(quicksort(L), pivot, quicksort(G))

function **pivot**(A)
  return randomIndex(A);

function **compare**(x, pivot)
  return human_compare(x, pivot)
Human Computation Algorithms

human-driven controls

sequence

conditional

repetition
Human Computation Algorithms
human-driven controls

sequence
conditional
repetition
parallel
Human Computation Algorithms

human-driven controls

sequence

conditional

true
false

repetition

parallel
Human Computation Algorithms

human-driven controls

sequence

conditional

repetition

parallel
Human Computation Algorithms
human-driven controls

sequence

conditional

true
false

repetition

parallel
Human Computation Algorithms
human-driven controls

- sequence
- conditional true false
- repetition
- parallel
Human Computation Algorithms
human-driven controls

sequence

conditional

true  false

repetition

parallel
Human Computation Algorithms

human-driven controls

sequence

conditional

ture false

repetition

parallel
Human Computation Algorithms
human-driven controls

sequence

true
false

conditional

repetition

parallel
Human Computation Algorithms

human-driven controls

sequence

conditional

ture false

repetition

parallel
Human Computation Algorithms

human-driven program synthesis
Human Computation Algorithms

human-driven program synthesis

plan a wedding
Human Computation Algorithms

human-driven program synthesis

plan a wedding

solvable?
Human Computation Algorithms
human-driven program synthesis

plan a wedding

solvable?

T

solve
Human Computation Algorithms
human-driven program synthesis

plan a wedding

solvable?

T  F

solve  decompose
Human Computation Algorithms
human-driven program synthesis

plan a wedding

solvable?

T → solve
F → decompose

ceremony

reception
Human Computation Algorithms
human-driven program synthesis

plan a wedding

solvable?

T
solve
F
decompose

ceremony
reception
Human Computation Algorithms
human-driven program synthesis

plan a wedding

solvable?

T  F

solve    decompose

ceremony reception

Turkomatic (Kulkarni et al., 2011)
CrowdForge (Kittur et al., 2011)
Human Computation Algorithms
automated design

Another improvement task?  How many votes?

TurKontrol (Dai et al., 2010; 2011)
PROPERTIES
Is the algorithm correct?
Correctness
Theoretical Analysis

What does it mean for a human computation algorithm to be correct?

What guarantees can we give regarding the correctness of a human computation algorithm?
Correctness
Three Points of Intervention
Correctness

Three Points of Intervention

input -> output

output aggregation after computation
Correctness

Three Points of Intervention
Correctness

Three Points of Intervention

task design
\textit{during computation}

input

\textit{before computation}

\textit{after computation}

output aggregation

output
Is the algorithm efficient?
Efficiency
Three Measures

Time Complexity
How long does it take?

Query Complexity
How many queries to the human computers?

Cost Effectiveness
How much does it cost?
Efficiency

Time Complexity

Operation Complexity
How does the number of operations scale?

VS

Clock Time
How much time does it actually take?
Efficiency
The need for real-time

vizwiz & quikTurKit (Bigham et al., 2010)

Retainer Model (Bernstein et al., 2011)
Efficiency
Query Complexity
Efficiency
Query Complexity

1. Repeated Labeling
For each input object, how many human computers do we query?
Efficiency
Query Complexity

1. Repeated Labeling
For each input object, how many human computers do we query?

(Sheng et al., 2009; Kumar and Lease, 2011)
Efficiency

Query Complexity
2. Active Learning
Which input should we process? What questions should we ask?
Active Learning

a short introduction

“The learner can select the data from which it learns.”

(Settles, 2011)
Active Learning

A short introduction

“The learner can select the data from which it learns.”
(Settles, 2011)

A single perfect oracle

Label / feature / feature value
Active Learning
a short introduction

“The learner can select the data from which it learns.”
(Settles, 2011)

a single perfect oracle

label / feature / feature value
Active Learning
a short introduction

“The learner can select the data from which it learns.”
(Settles, 2011)

multiple imperfect oracles
a single perfect oracle

label / feature / feature value
Active Learning
a short introduction

“The learner can select the data from which it learns.”
(Settles, 2011)

multiple imperfect oracles

a single perfect oracle

label / feature feature value
Active Learning
a short introduction

“The learner can select the data from which it learns.”
(Settles, 2011)

multiple imperfect oracles

a single perfect oracle

richer, different kinds of questions

label / feature feature value
Active Learning
example # 1

(a) Australia, China, United States

(b) World flags

(Tamuz et al., 2011)
Active Learning

example # 2

(Branson et al., 2010)
Efficiency
Cost Effectiveness
How do we price each task?
Will the total cost be within budget?
What is the total cost in the worst case?
Can we minimize cost?
What is the cost-benefit tradeoff?
TAKE-HOME

“human computation algorithms ↔ automated algorithms”
OUTPUT AGGREGATION

Motivation • Simple Outputs • Complex Outputs
Outputs can be aggregated by humans or automatically.
Correctness
Three Points of Intervention

- **Task Design**
  - *during computation*

- **Input**
- **Task Router**
  - *before computation*

- **Output**
- **Output Aggregation**
  - *after computation*
Correctness
Three Points of Intervention

- Task design *during computation*
- Task router *before computation*
- Output aggregation *after computation*
MOTIVATION
Outputs generated by human computers can be noisy.
Noise

is not only about inaccuracy
Noise is not only about inaccuracy

(Law and von Ahn, 2009)
<table>
<thead>
<tr>
<th>CLASSICAL</th>
<th>GRUITAR</th>
<th>FEMALE</th>
<th>RENNAISSANCE</th>
<th>STOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUITAR</td>
<td>PRIMAL</td>
<td>VOCAL</td>
<td>SWING</td>
<td>SKIPPY</td>
</tr>
<tr>
<td>PIANO</td>
<td>ACCUSTIC</td>
<td>QUIET</td>
<td>SCI-FI</td>
<td>FOREIGN</td>
</tr>
<tr>
<td>VIOLIN</td>
<td>ACTIVE</td>
<td>SITAR</td>
<td>HIPPIE</td>
<td>CHRISTM</td>
</tr>
<tr>
<td>ROCK</td>
<td>MEOW</td>
<td>CLASSIC</td>
<td>LULLABY</td>
<td>MAHY</td>
</tr>
<tr>
<td>SLOW</td>
<td>OHOHOH</td>
<td>SOFT</td>
<td>ANGELIC</td>
<td>CLAPPY</td>
</tr>
<tr>
<td>STRINGS</td>
<td>GRADUAL</td>
<td>CELLO</td>
<td>DOWNBEAT</td>
<td>CLOU</td>
</tr>
<tr>
<td>TECHNO</td>
<td>CLIMATIC</td>
<td>WOMAN</td>
<td>RELAXATION</td>
<td>SEASIDE</td>
</tr>
<tr>
<td>OPERA</td>
<td>PENSIVE</td>
<td>MALE</td>
<td>GLOOMY</td>
<td>MAMBO</td>
</tr>
<tr>
<td>DRUMS</td>
<td>HOUSY</td>
<td>SINGING</td>
<td>ROYAL</td>
<td>MANDOLIN</td>
</tr>
<tr>
<td>SAME</td>
<td>INSTRUMENTAL</td>
<td>VOCALS</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>FLUTE</td>
<td>CALMISH</td>
<td>SOLO</td>
<td>RHYTHMIC</td>
<td>FOLK</td>
</tr>
<tr>
<td>FAST</td>
<td>FEMALE OPERA</td>
<td>LOUD</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>DIFF</td>
<td>VARIED</td>
<td>CHOIR</td>
<td>RAGTIME</td>
<td>MELODY</td>
</tr>
<tr>
<td>ELECTRONIC</td>
<td>HEALING</td>
<td>VIOLINS</td>
<td>MUFFLED</td>
<td>HARMONICA</td>
</tr>
<tr>
<td>AMBIENT</td>
<td>WAVEY</td>
<td>HARP</td>
<td>FANTASY</td>
<td>ITALIAN</td>
</tr>
<tr>
<td>BEAT</td>
<td>DRIPPING</td>
<td>BEATS</td>
<td>HISPANIC</td>
<td>DRAMATIC</td>
</tr>
<tr>
<td>HARPSICORD</td>
<td>HEBREW</td>
<td>NOT ROCK</td>
<td>FANTASY</td>
<td>GENTLE</td>
</tr>
<tr>
<td>SYNTH</td>
<td>ANIMALS</td>
<td>WIERD</td>
<td>BEATLES</td>
<td>BLUEGRASS</td>
</tr>
<tr>
<td>INDIAN</td>
<td>REEDS</td>
<td>DANCE</td>
<td>SYNCOPATED</td>
<td>SPACESHIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MID-TEMPO</td>
<td>DESCENDING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RATTLE</td>
<td>COOKIE MONSTER</td>
</tr>
</tbody>
</table>

(Law, Settles and Mitchell, 2010)
Noise
example from TagATune

<table>
<thead>
<tr>
<th>CLASSICAL</th>
<th>GRUITAR</th>
<th>FEMALE</th>
<th>RENNAISSANCE</th>
<th>STOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUITAR</td>
<td>PRIMAL</td>
<td>VOCAL</td>
<td>SWING</td>
<td>SKIPPY</td>
</tr>
<tr>
<td>PIANO</td>
<td>ACCUSTIC</td>
<td>QUIET</td>
<td>SCI-FI</td>
<td>FOREIGN</td>
</tr>
<tr>
<td>VIOLIN</td>
<td>ACTIVE</td>
<td>SITAR</td>
<td>HIPPIE</td>
<td>CHRISTMASSY</td>
</tr>
<tr>
<td>ROCK</td>
<td>MEOW</td>
<td>CLASSIC</td>
<td>LULLABY</td>
<td>CLAPPY</td>
</tr>
<tr>
<td>SLOW</td>
<td>OHOHOH</td>
<td>SOFT</td>
<td>ANGELIC</td>
<td>CLOUDY</td>
</tr>
<tr>
<td>STRINGS</td>
<td>GRADUAL</td>
<td>CELLO</td>
<td>DOWNBEAT</td>
<td>SEASIDE</td>
</tr>
<tr>
<td>TECHNO</td>
<td>CLIMATIC</td>
<td>WOMAN</td>
<td>RELAXATION</td>
<td>MAMBO</td>
</tr>
<tr>
<td>OPERA</td>
<td>PENSIVE</td>
<td>MALE</td>
<td>GLOOMY</td>
<td>MANDOLIN</td>
</tr>
<tr>
<td>DRUMS</td>
<td>HOUSY</td>
<td>SINGING</td>
<td>ROYAL</td>
<td>FOLK</td>
</tr>
<tr>
<td>SAME</td>
<td>INSTRUMENTAL</td>
<td>VOCALS</td>
<td>RHYTHMIC</td>
<td>NO VIOLINS</td>
</tr>
<tr>
<td>FLUTE</td>
<td>CALMISH</td>
<td>SOLO</td>
<td>MUFFLED</td>
<td>MELODY</td>
</tr>
<tr>
<td>FAST</td>
<td>FEMALE OPERA</td>
<td>LOUD</td>
<td>RAGTIME</td>
<td>HARMONICA</td>
</tr>
<tr>
<td>DIFF</td>
<td>VARIED</td>
<td>CHOIR</td>
<td>TUDOR</td>
<td>ITALIAN</td>
</tr>
<tr>
<td>ELECTRONIC</td>
<td>HEALING</td>
<td>VIOLINS</td>
<td>FANTASY</td>
<td>DRAMATIC</td>
</tr>
<tr>
<td>AMBIENT</td>
<td>WAVEY</td>
<td>HARP</td>
<td>HISPANIC</td>
<td>BLUEGRASS</td>
</tr>
<tr>
<td>BEAT</td>
<td>DRIPPING</td>
<td>BEATS</td>
<td>BEATLES</td>
<td>GENTLE</td>
</tr>
<tr>
<td>HARPSCORD</td>
<td>HEBREW</td>
<td>NOT ROCK</td>
<td>SYNCOPATED</td>
<td>SPACESHIP DESCENDING</td>
</tr>
<tr>
<td>SYNTH</td>
<td>ANIMALS</td>
<td>WIERD</td>
<td>MID-TEMPO</td>
<td>COOKIE MONSTER VOCAL</td>
</tr>
<tr>
<td>INDIAN</td>
<td>REEDS</td>
<td>DANCE</td>
<td>RATTLE</td>
<td>VAMPIRES AT A DINNER PARTY</td>
</tr>
</tbody>
</table>

(Law, Settles and Mitchell, 2010)
Noise
example from TagATune

(Law, Settles and Mitchell, 2010)
Noise

example from TagATune

CLASSICAL  GRUITAR  FEMALE  RENNAISSANCE  STOMP
GUITAR     PRIMAL   VOCAL   SWING     SKIPPY
PIANO      ACCUSTIC QUIET   SCI-FI     FOREIGN
VIOLIN     ACTIVE   SITAR   HIPPIE     CHRISTMASSY
ROCK       MEOW     CLASSIC LULLABY    CLAPPY
SLOW       OHOHOH   SOFT    ANGELIC    CLOUDY
STRINGS    GRADUAL  CELLO   DOWNBEAT  SEASIDE
TECHNO     CLIMATIC  CELLO   WOMAN     RELAXATION
OPERA      PENSIVE  SOLO    MALE      GLOOMY
DRUMS      HOUSY    SINGING SINGING    ROYAL
SAME       INSTRUMENTAL VOCALS  ANGELIC    FOLK
DIFF       CALMISH  VOCIOMS  MUFFLED    NO VIOLINS
ELECTRONIC healer  HEALING  RHYTHMIC   MELODY
AMBIENT    WAVEY    LOUD    RAGTIME    HARMONICA
BEAT       DRIPPING CHOIR   TUDOR      ITALIAN
HARP       HEBREW   BEATS   BEATLES     DRAMATIC
SYNTHE     ANIMALS  NOT ROCK  SYNCPATATED GENTLE
INDIAN     REEDS    WIERD   MID-TEMPO  SPACESHIP DESCENDING
            (Law, Settles and Mitchell, 2010)
<table>
<thead>
<tr>
<th>CLASSICAL</th>
<th>GRUITAR</th>
<th>FEMALE</th>
<th>RENNAISSANCE</th>
<th>STOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUITAR</td>
<td>PRIMAL</td>
<td>VOCAL</td>
<td>SWING</td>
<td>SKIPPY</td>
</tr>
<tr>
<td>PIANO</td>
<td>ACCUSTIC</td>
<td>QUIET</td>
<td>SCI-FI</td>
<td>FOREIGN</td>
</tr>
<tr>
<td>VIOLIN</td>
<td>ACTIVE</td>
<td>SITAR</td>
<td>HIPPIE</td>
<td>CHRISTMASSY</td>
</tr>
<tr>
<td>ROCK</td>
<td>MEOW</td>
<td>CLASSIC</td>
<td>LULLABY</td>
<td>CLAPPY</td>
</tr>
<tr>
<td>SLOW</td>
<td>OHOOHOH</td>
<td>SOFT</td>
<td>ANGELIC</td>
<td>CLOUDY</td>
</tr>
<tr>
<td>STRINGS</td>
<td>GRADUAL</td>
<td>CELLO</td>
<td>DOWNBEAT</td>
<td>SEASIDE</td>
</tr>
<tr>
<td>TECHNO</td>
<td>CLIMATIC</td>
<td>WOMAN</td>
<td>RELAXATION</td>
<td>MAMBO</td>
</tr>
<tr>
<td>OPERA</td>
<td>PENIVE</td>
<td>MALE</td>
<td>GLOOMY</td>
<td>MANDOLIN</td>
</tr>
<tr>
<td>DRUMS</td>
<td>HOUSY</td>
<td>SINGING</td>
<td>ROYAL</td>
<td>FOLK</td>
</tr>
<tr>
<td>SAME</td>
<td>INSTRUMENTAL</td>
<td>VOCALS</td>
<td>RHYTHMIC</td>
<td>NO VIOLINS</td>
</tr>
<tr>
<td>FLUTE</td>
<td>CALMISH</td>
<td>SOLO</td>
<td>MUFFLED</td>
<td>MELODY</td>
</tr>
<tr>
<td>FAST</td>
<td>FEMALE OPERA</td>
<td>LOUD</td>
<td>RAGTIME</td>
<td>HARMONICA</td>
</tr>
<tr>
<td>DIFF</td>
<td>VARIED</td>
<td>CHOIR</td>
<td>TUDOR</td>
<td>ITALIAN</td>
</tr>
<tr>
<td>ELECTRONIC</td>
<td>HEALING</td>
<td>VIOLINS</td>
<td>FANTASY</td>
<td>DRAMATIC</td>
</tr>
<tr>
<td>AMBIENT</td>
<td>WAVEY</td>
<td>HARP</td>
<td>HISPANIC</td>
<td>BLUEGRASS</td>
</tr>
<tr>
<td>BEAT</td>
<td>DRIPPING</td>
<td>BEATS</td>
<td>BEATLES</td>
<td>GENTLE</td>
</tr>
<tr>
<td>HARPSCORD</td>
<td>HEBREW</td>
<td>NOT ROCK</td>
<td>SYNCOPATED</td>
<td>SPACESHIP DESCENDING</td>
</tr>
<tr>
<td>SYNTH</td>
<td>ANIMALS</td>
<td>WIERD</td>
<td>MID-TEMPO</td>
<td>COOKIE MONSTER VOCAL</td>
</tr>
<tr>
<td>INDIAN</td>
<td>REEDS</td>
<td>DANCE</td>
<td>RATTLE</td>
<td>VAMPIRES AT A DINNER PARTY</td>
</tr>
</tbody>
</table>

(Law, Settles and Mitchell, 2010)
Noise example from TagATune

<table>
<thead>
<tr>
<th>CLASSICAL</th>
<th>ACCUSTIC</th>
<th>FEMALE</th>
<th>OPERA</th>
<th>STOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUITAR</td>
<td>PRIMAL</td>
<td>VOCAL</td>
<td>PENSIVE</td>
<td>SKIPPY</td>
</tr>
<tr>
<td>PIANO</td>
<td>ACCUSTIC</td>
<td>QUIET</td>
<td>MEOW</td>
<td>FOREIGN</td>
</tr>
<tr>
<td>VIOLIN</td>
<td>ACTIVE</td>
<td>SITAR</td>
<td>CLASSIC</td>
<td>CHRISTMAS</td>
</tr>
<tr>
<td>ROCK</td>
<td>MEOW</td>
<td>SITAR</td>
<td>SOFT</td>
<td>CLAPPY</td>
</tr>
<tr>
<td>SLOW</td>
<td>OHOHOH</td>
<td>CELLO</td>
<td>Cello</td>
<td>CLOUDY</td>
</tr>
<tr>
<td>STRINGS</td>
<td>GRADUAL</td>
<td>WOMAN</td>
<td>DOWNBEAT</td>
<td>SEASIDE</td>
</tr>
<tr>
<td>TECHNO</td>
<td>CLIMATIC</td>
<td>MALE</td>
<td>RELAXATION</td>
<td>MAMBO</td>
</tr>
<tr>
<td>OPERA</td>
<td>PENSIVE</td>
<td>SINGING</td>
<td>GLOOMY</td>
<td>MANDOLIN</td>
</tr>
<tr>
<td>DRUMS</td>
<td>HOUSEY</td>
<td>VOCALS</td>
<td>ROYAL</td>
<td>FOLK</td>
</tr>
<tr>
<td>SAME</td>
<td>INSTRUMENTAL</td>
<td>SOLO</td>
<td>MUFFLED</td>
<td>NO VIOLINS</td>
</tr>
<tr>
<td>FLUTE</td>
<td>CALMISH</td>
<td>SINGING</td>
<td>RHYTHMIC</td>
<td>MELODY</td>
</tr>
<tr>
<td>FAST</td>
<td>FEMALE OPERA</td>
<td>LOUD</td>
<td>RAGTIME</td>
<td>HARMONICA</td>
</tr>
<tr>
<td>DIFF</td>
<td>VARIED</td>
<td>CHOIR</td>
<td>TUDOR</td>
<td>ITALIAN</td>
</tr>
<tr>
<td>ELECTRONIC</td>
<td>HEALING</td>
<td>VIOLINS</td>
<td>FANTASY</td>
<td>DRAMATIC</td>
</tr>
<tr>
<td>AMBIENT</td>
<td>WAVEY</td>
<td>HARP</td>
<td>HISPANIC</td>
<td>BLUEGRASS</td>
</tr>
<tr>
<td>BEAT</td>
<td>DRIPPING</td>
<td>BEATS</td>
<td>BEATLES</td>
<td>GENTLE</td>
</tr>
<tr>
<td>HARPICODE</td>
<td>HEBREW</td>
<td>BEATS</td>
<td>SYMPHONIC</td>
<td>SPACESHIP DESCENDING</td>
</tr>
<tr>
<td>SYNTH</td>
<td>ANIMALS</td>
<td>NOT ROCK</td>
<td>SYNCOPATED</td>
<td>COOKIE MONSTER VOCAL</td>
</tr>
<tr>
<td>INDIAN</td>
<td>REEDS</td>
<td>DANCE</td>
<td>MID-TEMPO</td>
<td>VAMPIRES AT A DINNER PARTY</td>
</tr>
</tbody>
</table>

(Law, Settles and Mitchell, 2010)
### Noise

**example from TagATune**

<table>
<thead>
<tr>
<th>CLASSICAL</th>
<th>GRUITAR</th>
<th>FEMALE</th>
<th>RENNAISSANCE</th>
<th>STOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUITAR</td>
<td>PRIMAL</td>
<td>VOCAL</td>
<td>SWING</td>
<td>SKIPPY</td>
</tr>
<tr>
<td>PIANO</td>
<td>ACCUSTIC</td>
<td>QUIET</td>
<td>SCI-FI</td>
<td>FOREIGN</td>
</tr>
<tr>
<td>VIOLIN</td>
<td>ACTIVE</td>
<td>SITAR</td>
<td>HIPPIE</td>
<td>CHRISTMASSY</td>
</tr>
<tr>
<td>ROCK</td>
<td>MEOW</td>
<td>CLASSIC</td>
<td>LULLABY</td>
<td>CLAPPY</td>
</tr>
<tr>
<td>SLOW</td>
<td>OHOHOH</td>
<td>SOFT</td>
<td>ANGELIC</td>
<td>CLOUDY</td>
</tr>
<tr>
<td>STRINGS</td>
<td>GRADUAL</td>
<td>CELLO</td>
<td>DOWNBEAT</td>
<td>SEASIDE</td>
</tr>
<tr>
<td>TECHNO</td>
<td>CLIMATIC</td>
<td>WOMAN</td>
<td>RELAXATION</td>
<td>MAMBO</td>
</tr>
<tr>
<td>OPERA</td>
<td>PENSIVE</td>
<td>MALE</td>
<td>GLOOMY</td>
<td>MANDOLIN</td>
</tr>
<tr>
<td>DRUMS</td>
<td>HOUSY</td>
<td>SINGING</td>
<td>ROYAL</td>
<td>FOLK</td>
</tr>
<tr>
<td>SAME</td>
<td>INSTRUMENTAL</td>
<td>VOCALS</td>
<td>RHYTHMIC</td>
<td>NO VIOLINS</td>
</tr>
<tr>
<td>FLUTE</td>
<td>CALMISH</td>
<td>SOLO</td>
<td>MUDDLED</td>
<td>MELODY</td>
</tr>
<tr>
<td>FAST</td>
<td>FEMALE OPERA</td>
<td>LOUD</td>
<td>RAGTIME</td>
<td>HARMONICA</td>
</tr>
<tr>
<td>DIFF</td>
<td>VARIED</td>
<td>CHOIR</td>
<td>TUDOR</td>
<td>ITALIAN</td>
</tr>
<tr>
<td>ELECTRONIC</td>
<td>HEALING</td>
<td>VIOLINS</td>
<td>FANTASY</td>
<td>DRAMATIC</td>
</tr>
<tr>
<td>AMBIENT</td>
<td>WAVEY</td>
<td>HARP</td>
<td>HISPANIC</td>
<td>GENTLE</td>
</tr>
<tr>
<td>BEAT</td>
<td>DRIPPING</td>
<td>BEATS</td>
<td>BEATLES</td>
<td>SPACESHIP DESCENDING</td>
</tr>
<tr>
<td>HARPSCORD</td>
<td>HEBREW</td>
<td>NOT ROCK</td>
<td>SYNCOPATED</td>
<td>COOKIE MONSTER VOCAL</td>
</tr>
<tr>
<td>SYNTH</td>
<td>ANIMALS</td>
<td>WIERD</td>
<td>MID-TEMPO</td>
<td>VAMPIRES AT A DINNER PARTY</td>
</tr>
<tr>
<td>INDIAN</td>
<td>REEDS</td>
<td>DANCE</td>
<td>RATTLE</td>
<td></td>
</tr>
</tbody>
</table>

(Law, Settles and Mitchell, 2010)
# Noise

**example from TagATune**

<table>
<thead>
<tr>
<th>CLASSICAL</th>
<th>GRUITAR</th>
<th>FEMALE</th>
<th>RENNAISSANCE</th>
<th>STOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUITAR</td>
<td>PRIMAL</td>
<td>VOCAL</td>
<td>SWING</td>
<td>SKIPPY</td>
</tr>
<tr>
<td>PIANO</td>
<td>ACCUSTIC</td>
<td>QUIET</td>
<td>SCI-FI</td>
<td>FOREIGN</td>
</tr>
<tr>
<td>VIOLIN</td>
<td>ACTIVE</td>
<td>SITAR</td>
<td>HIPPIE</td>
<td>CHRISTMASSY</td>
</tr>
<tr>
<td>ROCK</td>
<td>MEOW</td>
<td>CLASSIC</td>
<td>LULLABY</td>
<td>CLAPPY</td>
</tr>
<tr>
<td>SLOW</td>
<td>OHOHOH</td>
<td>SOFT</td>
<td>ANGELIC</td>
<td>CLOUDY</td>
</tr>
<tr>
<td>STRINGS</td>
<td>GRADUAL</td>
<td>CELLO</td>
<td>DOWNBEAT</td>
<td>SEASIDE</td>
</tr>
<tr>
<td>TECHNO</td>
<td>CLIMATIC</td>
<td>WOMAN</td>
<td>RELAXATION</td>
<td>MAMBO</td>
</tr>
<tr>
<td>OPERA</td>
<td>PENSIVE</td>
<td>MALE</td>
<td>GLOOMY</td>
<td>MANDOLIN</td>
</tr>
<tr>
<td>DRUMS</td>
<td>HOUSY</td>
<td>SINGING</td>
<td>ROYAL</td>
<td>FOLK</td>
</tr>
<tr>
<td>SAME</td>
<td>INSTRUMENTAL</td>
<td>VOCALS</td>
<td>Rythmic</td>
<td>no violins</td>
</tr>
<tr>
<td>FLUTE</td>
<td>CALMISH</td>
<td>SOLO</td>
<td>MUFFLED</td>
<td>MELODY</td>
</tr>
<tr>
<td>FAST</td>
<td>FEMALE OPERA</td>
<td>LOUD</td>
<td>RAGTIME</td>
<td>HARMONICA</td>
</tr>
<tr>
<td>DIFF</td>
<td>VARIED</td>
<td>CHOIR</td>
<td>TUDOR</td>
<td>ITALIAN</td>
</tr>
<tr>
<td>ELECTRONIC</td>
<td>HEALING</td>
<td>VIOLINS</td>
<td>FANTASY</td>
<td>DRAMATIC</td>
</tr>
<tr>
<td>AMBIENT</td>
<td>WAVEY</td>
<td>HARP</td>
<td>HISPANIC</td>
<td>BLUEGRASS</td>
</tr>
<tr>
<td>BEAT</td>
<td>DRIPPING</td>
<td>BEATS</td>
<td>BEATLES</td>
<td>GENTLE</td>
</tr>
<tr>
<td>HARPSCPORD</td>
<td>HEBREW</td>
<td>NOT ROCK</td>
<td>SYNCOPATED</td>
<td>SPACESHIP DESCENDING</td>
</tr>
<tr>
<td>SYNTH</td>
<td>ANIMALS</td>
<td>WIERD</td>
<td>MID-TEMPO</td>
<td>COOKIE MONSTER VOCAL</td>
</tr>
<tr>
<td>INDIAN</td>
<td>REEDS</td>
<td>DANCE</td>
<td>RATTLE</td>
<td>VAMPIRES AT A DINNER PARTY</td>
</tr>
</tbody>
</table>

*(Law, Settles and Mitchell, 2010)*
<table>
<thead>
<tr>
<th>CLASSICAL</th>
<th>GRUITAR</th>
<th>FEMALE</th>
<th>RENNAISSANCE</th>
<th>STOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUITAR</td>
<td>PRIMAL</td>
<td>VOCAL</td>
<td>SWING</td>
<td>SKIPPY</td>
</tr>
<tr>
<td>PIANO</td>
<td>ACCUSTIC</td>
<td>QUIET</td>
<td>SCI-FI</td>
<td>FOREIGN</td>
</tr>
<tr>
<td>VIOLIN</td>
<td>ACTIVE</td>
<td>SITAR</td>
<td>HIPPIE</td>
<td>CHRISTMASSY</td>
</tr>
<tr>
<td>ROCK</td>
<td>MEOW</td>
<td>CLASSIC</td>
<td>LULLABY</td>
<td>CLAPPY</td>
</tr>
<tr>
<td>SLOW</td>
<td>OHOHOH</td>
<td>SOFT</td>
<td>ANGELIC</td>
<td>CLOUDY</td>
</tr>
<tr>
<td>STRINGS</td>
<td>GRADUAL</td>
<td>CELLO</td>
<td>DOWNBEAT</td>
<td>SEASIDE</td>
</tr>
<tr>
<td>TECHNO</td>
<td>CLIMATIC</td>
<td>WOMAN</td>
<td>RELAXATION</td>
<td>MAMBO</td>
</tr>
<tr>
<td>OPERA</td>
<td>PENSIVE</td>
<td>MALE</td>
<td>GLOOMY</td>
<td>MANDOLIN</td>
</tr>
<tr>
<td>DRUMS</td>
<td>HOUSY</td>
<td>SINGING</td>
<td>ROYAL</td>
<td>FOLK</td>
</tr>
<tr>
<td>SAME</td>
<td>INSTRUMENTAL</td>
<td>VOCALS</td>
<td>RHYTHMIC</td>
<td>NO VIOLINS</td>
</tr>
<tr>
<td>FLUTE</td>
<td>CALMISH</td>
<td>SOLO</td>
<td>MUDDLED</td>
<td>MELODY</td>
</tr>
<tr>
<td>FAST</td>
<td>FEMALE OPERA</td>
<td>LOUD</td>
<td>RAGTIME</td>
<td>HARMONICA</td>
</tr>
<tr>
<td>DIFF</td>
<td>VARIED</td>
<td>CHOIR</td>
<td>TUDOR</td>
<td>ITALIAN</td>
</tr>
<tr>
<td>ELECTRONIC</td>
<td>HEALING</td>
<td>VIOLINS</td>
<td>FANTASY</td>
<td>DRAMATIC</td>
</tr>
<tr>
<td>AMBIENT</td>
<td>WAVEY</td>
<td>HARP</td>
<td>HISPANIC</td>
<td>BLUEGRASS</td>
</tr>
<tr>
<td>BEAT</td>
<td>DRIPPING</td>
<td>BEATS</td>
<td>BEATLES</td>
<td>GENTLE</td>
</tr>
<tr>
<td>HARPSCORD</td>
<td>HEBREW</td>
<td>NOT ROCK</td>
<td>SYNCOPATED</td>
<td>SPACESHIP DESCENDING</td>
</tr>
<tr>
<td>SYNTH</td>
<td>ANIMALS</td>
<td>WIERD</td>
<td>MID-TEMPO</td>
<td>COOKIE MONSTER VOCAL</td>
</tr>
<tr>
<td>INDIAN</td>
<td>REEDS</td>
<td>DANCE</td>
<td>RATTLE</td>
<td>VAMPIRES AT A DINNER PARTY</td>
</tr>
</tbody>
</table>

(Law, Settles and Mitchell, 2010)
The “truth” exists, and through redundancy we can find it.
Truth
objective versus cultural

Objective Truth

Cultural Truth
Truth
objective versus cultural

Objective Truth

a definitive answer exists beyond human judgments, but hard to reach.

   e.g., cancer or not number of volcanos on Venus location or time of a photo

Cultural Truth
Truth
objective versus cultural

Objective Truth

a definitive answer exists beyond human judgments, but hard to reach.

  e.g., cancer or not
  number of volcanos on Venus
  location or time of a photo

Cultural Truth

shared beliefs of a group of people, often involving perceptual judgments.

  e.g., is this music calm?
  is this image pornographic?
  is this disease contagious?
Statistical Measures of Agreement
(Artstein and Poesio, 2008)
The simplest way to aggregate is majority vote.
Statistical Measures of Agreement
(Artstein and Poesio, 2008)

The simplest way to aggregate is majority vote.

But how much agreement is really there?
Majority Vote
as a graphical model

$Y_n$  
true label

$O_{n,m}$  
observed label

N classification questions, M workers
Hidden Factors

that influence the annotation process
Hidden Factors
that influence the annotation process

Task Characteristics
Quality (e.g., blurry pictures)
Difficulty (e.g., transcription of non-native speech)
Hidden Factors
that influence the annotation process

Task Characteristics
  Quality (e.g., blurry pictures)
  Difficulty (e.g., transcription of non-native speech)

Worker Characteristics
  Expertise (e.g., bird identification)
  Bias (e.g., mother vs college students)
  Physical Conditions (e.g., fatigue)
Latent Class Model
for classification
Latent Class Model
for classification

Dawid and Skeen, 1979
Uebersax et al., 1993
Carpenter, 2008
Whitehill et al., 2009

Ipeirotis et al., 2010
Raykar et al., 2010
Welinder and Perona, 2010
Ipeirotis et al., 2010
Latent Class Model

an example

Grebe?

(Welinder et al., 2010)
Latent Class Model

an example

(Welinder et al., 2010)
Latent Class Model
an example

(Welinder et al., 2010)
### Cultural Consensus Theory in Anthropology

from (Romney, 1999)

(Romney et al., 1986; Karabatsos and Batchelder, 2003; Weller, 2007)

|                           | 10 | 17 | 9 | 23 | 22 | 6 | 18 | 2 | 3 | 5 | 8 | 21 | 1 | 14 | 7 | 13 | 16 | 4 | 20 | 12 | 19 | 15 | 
|---------------------------|----|----|---|----|----|---|----|---|---|---|---|----|---|----|---|----|----|---|----|----|---|----|----|---|----|
| Allergies                 |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Kidney pain               |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Gastritis                 |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Amoebas                   |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Appendicitis              |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Hepatitis                 |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Mumps                     |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Rubella                   |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Measles                   |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Smallpox                  |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Cancer                    |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Diabetes                  |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Intestinal influenza      |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Tetanus                   |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Chicken pox               |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Tonsillitis               |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Folio                     |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Diarrhea                  |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Typhoid fever             |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Diphtheria                |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Arthritis                 |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Whooping cough            |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Tuberculosis              |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Malaria                   |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Colic                     |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Rheumatism                |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Flu                       |    |    |   |    |    |   |    |   |   |   |   |     |   |     |   |     |     |   |     |     |   |     |     |   |     |
| Errors                    | 2  | 3  | 7 | 6 | 4 | 7 | 4 | 10 | 6 | 6 | 9 | 8 | 4 | 6 | 8 | 6 | 9 | 8 | 9 | 6 | 7 | 12 | 12 |
Other Challenges

What if we cannot assume repeated labeling?
Other Challenges

What if we cannot assume repeated labeling?

“Learn a hypothesis to simulate the aggregate output, and prune away workers that don’t agree”

(Dekel and Shamir, 2009)
Complex Outputs
and challenges

ranking & clustering
structured outputs
beliefs
Challenge #1: deciding how to decompose the problem
Ranking Aggregation

individual rankings ➞ full ranking

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)
Ranking Aggregation

individual rankings ➔ full ranking

paired comparison

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)
Ranking Aggregation

individual rankings ➔ full ranking

paired comparison

ing rating (scale 1-4)

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)
Ranking Aggregation

individual rankings $\rightarrow$ full ranking

paired comparison  rating (scale 1-4)  ordering

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)
Ranking Aggregation

an example

(Hacker et al., 2009)
Consensus Clustering
individual clusterings ➔ single clustering

(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)
Consensus Clustering

individual clusterings ➔ single clustering

(complete clustering)

(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)
Consensus Clustering

individual clusterings $\Rightarrow$ single clustering

complete clustering

link / cannot-link constraints

(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)
Consensus Clustering

Individual clusterings ➔ single clustering

Complete clustering

Link / cannot-link constraints

How similar are these? (scale 1-5)

(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)
Consensus Clustering

an example

(Parent and Eskenazi, 2010)
Do the following definitions of the word aid have the same or different meaning?

- a piece of equipment that helps you to do something.
- something such as a machine or tool that helps someone do something.

(Consensus Clustering: an example)
Do the following definitions of the word ‘aid’ have the same or different meaning?

- a piece of equipment that helps you to do something.
- something such as a machine or tool that helps someone do something.

You have to group the definitions for the word ‘code’. There are 2 general meanings.

- to mark a group of things with different colors so that you can tell the difference between them.
- to put a message in code so that it is secret.
- to put a set of numbers, letters, or signs on something to show that it is or give information about it.
- to represent a message in code so that it can only be understood by the person who is meant to receive it.
Challenge #2:

the correspondence problem
Structured Outputs

transcription, translation and description

least difficult

most difficult
Structured Outputs
transcription, translation and description

least difficult

Transcription
ROVER method (Fiscus, 1997)
Longest Common Subsequences, Lattice (Evanini et al., 2010)

most difficult
Structured Outputs
transcription, translation and description

least difficult

Transcription
ROVER method (Fiscus, 1997)
Longest Common Subsequences, Lattice (Evanini et al., 2010)

Translation
BLEU (Pipineni et al., 2002);
Consensus Translation (Bangalore et al., 2001; Frederking and Nirenburg, 1994, Matusov et al., 2006, Rosti et al., 2007)

most difficult
Structured Outputs
transcription, translation and description

least difficult

**Transcription**
- ROVER method (Fiscus, 1997)
- Longest Common Subsequences, Lattice (Evanini et al., 2010)

**Translation**
- BLEU (Pipineni et al., 2002);
- Consensus Translation (Bangalore et al., 2001; Frederking and Nirenburg, 1994, Matusov et al., 2006, Rosti et al., 2007)

**Description**
- Information Fushion (Barzilay, 2003; Barzilay et al., 1999)

most difficult
Challenge #3:

aggregating difficult to articulate outputs
Belief Aggregation
with prediction markets

The Google Lunar X Prize to be won on/before 31 Dec 2012

Last prediction was: $4.60 / share
Today's Change: -

Event: Google Lunar X Prize (Rule 1.8 Applies)

Think this event **will** occur?

**Buy Shares**

Current best (lowest) price to buy shares is $5.00 / share. There are approx. 1 share available at this price.

Think this event **won't** occur?

**Sell Shares**

Current best (highest) price to sell shares is $0.30 / share. There are approx. 25 shares available at this price.

---

GOOGLE.LUNAR.X.DEC2012

Sep 16, 2007 - Aug 05, 2011

- Closing Price 46
- Vol 0

Source: www.intrade.com ©
TAKE-HOME

“classification and beyond”
TASK ROUTING

Motivation • Push Methods • Pull Methods
Correctness
Three Points of Intervention

task design
_during computation_

 task router
_before computation_

output aggregation
_after computation_
MOTIVATION
The most popular task routing method is WHTBT.
The most popular task routing method is WHTBT.

(which stands for “Whoever Happens To Be There”).
All human computers are not created equal.
Push versus Pull
methods of task routing
Push versus Pull

types of task routing

Push

Workers are passive receivers of tasks.

The system takes complete control over who is assigned which task.
<table>
<thead>
<tr>
<th>Push</th>
<th>Pull</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workers are passive receivers of tasks. The system takes complete control over who is assigned which task.</td>
<td>Workers are active seekers of tasks. The system supports a set of interfaces that enable workers to look for tasks to assign themselves.</td>
</tr>
</tbody>
</table>
The system takes complete control over who is assigned which task.
Allocation

complete knowledge of utility and cost
Allocation

(Shahaf and Horvitz, 2010)
Workers have known competencies.
Tasks have known demands.

weighted exact set-cover problem
Participants have known cost and utility (based on what they can cover).

Workers have known competencies. Tasks have known demands.

Weighted exact set-cover problem

Budgeted maximum coverage problem
Matching
complete or partial preferences

- man to woman (Gale and Shapley, 1962)
- medical residents to hospitals (Roth, 1984)
- students to schools (Teo, 2001)
- sailors to ships (Liebowitz, 2000)

incomplete information (Gusfield and Irving, 1989; Liebowitz, 2000)
Inference
incomplete knowledge of utility and cost
Inference
incomplete knowledge of utility and cost

Decision-Theoretic Model
e.g., Donmez et al., 2008
Discovery and Assignment Phase
Inference
incomplete knowledge of utility and cost

Decision-Theoretic Model
e.g., Donmez et al., 2008
Discovery and Assignment Phase

Online Learning
e.g., Donmez et al., 2009
Exploration-Exploitation Tradeoff
PULL METHODS
The system merely sets up the environment to allow workers to assign themselves (or each other) tasks.
Search locating tasks

(Chilton et al., 2010)
Visualization

locating particular input objects
Visualization
locating particular input objects

(Borden, 2006)
Visualization
locating particular input objects

(Borden, 2006)

source: ablegrape.wordpress.com
Task Recommendation
personalization
Task Recommendation

Content-Based
find similarities between worker profile and task characteristics.

Collaborative Filtering
make use of preference information about tasks (e.g., ratings) to infer similarities between workers.

Hybrid
a mix of content-based and collaborative filtering methods.
Content-Based
find similarities between worker profile and task characteristics.

Collaborative Filtering
make use of preference information about tasks (e.g., ratings) to infer similarities between workers.

Hybrid
a mix of content-based and collaborative filtering methods.

SuggestBot (Cosley et al., 2006)
Peer Routing
people’s knowledge of each other
Peer Routing

people's knowledge of each other

DARPA Red Balloon Challenge
Peer Routing
people’s knowledge of each other

DARPA Red Balloon Challenge

(Zhang et al., 2011)
“Wisdom of the individuals in the crowd”
CONCLUSION
Conclusion

Summary
What have we learned?

Closing
What are some opportunities for AI research?
What

Bird’s Eye View
of this tutorial

Who

How
Human Computation Algorithms

- definition
- control, operation and synthesis
- programming
- correctness and efficiency

Bird’s Eye View of this tutorial
Bird’s Eye View
of this tutorial

Human Computation Algorithms
• definition
• control, operation and synthesis
• programming
• correctness and efficiency

Output Aggregation
• classification
• ranking and clustering
• structured outputs
• beliefs

What

How

Who
Bird’s Eye View
of this tutorial

Human Computation Algorithms
- definition
- control, operation and synthesis
- programming
- correctness and efficiency

Output Aggregation
- classification
- ranking and clustering
- structured outputs
- beliefs

Designing for Human Computers
- who they are
- what are their wants and needs

The Art of Asking Questions
- task design
- game design

Task Routing
- push versus pull
- allocation / matching
- inference / online learning
CLOSING
The Role of AI in human computation
The Role of AI
in human computation

AI as requesters
learning to recognize objects, translate sentences, classifying music by querying human teachers.
The Role of AI
in human computation

**AI as requesters**
learning to recognize objects, translate sentences, classifying music by querying human teachers.

**AI as optimizers**
 improve the accuracy and efficiency of human computation algorithms.
The Role of AI in human computation

**AI as requesters**
learning to recognize objects, translate sentences, classifying music by querying human teachers.

**AI as optimizers**
improve the accuracy and efficiency of human computation algorithms.

**AI as enablers**
make human computers better, e.g., by organizing and displaying information to workers.
The Role of AI in human computation

**AI as requesters**
learning to recognize objects, translate sentences, classifying music by querying human teachers.

**AI as optimizers**
improve the accuracy and efficiency of human computation algorithms.

**AI as enablers**
make human computers better, e.g., by organizing and displaying information to workers.

**AI as workers**
perform tasks that they are better at than humans.
a conceptual framework
an annotated bibliography
a place to get ideas for research
a work in progress

free for you! Come pick one up during the break.

Other resources:
THANK YOU &
CATCH YOU @ COFFEE!