Probabilistic Machine Learning
(theory and practice)

Charles Sutton
Introduction to Research in Data Science
University of Edinburgh

New methodology

- New model types
- Inference algorithms
  (e.g., high dimensional, streaming)
- Approximate learning methods

New applications

- Analyzing computer programs
- Data mining
- Exploratory data analysis
- Home energy demand
- Computer security

Syntactic Idioms in Code

```
...
if (c != null) {
  try {
   if (c.moveToFirst()) {
     number = c.getString(
              c.getColumnIndex(
               phoneColumn));
   }
  } finally {
   c.close();
  }
}
...
```

(a)

```
IfStatement
  expression:
    c!=null
  then:Block
    TryStatement
      body:IfStatement
        expr:MethodInvocation
          expr:var%android.database.Cursor%
            name:c
          name:moveToFirst
        then:Block
          number = c.getString(c.getColumnIndex(phoneColumn));
      finally:Block
        ExpressionStatement
          MethodInvocation
            expr:var%android.database.Cursor%
              name:c
            name:close
```

E

```
try {
  if ($(Cursor).moveToFirst()) {
    $BODY$
  }
} finally {
  $(Cursor).close();
}
```

Allamanis and Sutton, FSE 2014

Example Idioms

From: Nonparametric Bayesian Tree Substitution Grammar

[Post and Gildea, 2009; Cohn et al, 2010]

```
channel=connection.
  createChannel();
```

(a)

```
Elements $name=$(Element).
  select($StringLit);
```

(b)

```
Transaction tx=ConnectionFactory.
  getDatabase().beginTx();
```

(c)

```
catch (Exception e){
  $(Transaction).failure();
}
```

(d)

```
SearchSourceBuilder builder=
  getQueryTranslator().build(
    $(ContentIndexQuery));
```

(e)

```
LocationManager $name =
  (LocationManager)getSystemService(
    Context.LOCATION_SERVICE);
```

(f)

```
Location.distanceBetween(
    $(Location).getLatitude(),
    $(Location).getLongitude(),
    $...);
```

(g)

```
try{
  $BODY$
}finally{
  $(RevWalk).release();
}
```

(h)

```
try{
  Node $name=$methodInvoc();
  $BODY$
}finally{
  $(Transaction).finish();
}
```

(i)

```
ConnectionFactory factory =
  new ConnectionFactory();
$methodInvoc();
Connection connection =
  factory.newConnection();
```

(j)

```
while ($(ModelNode) != null){
  if ($(ModelNode) == limit)
    break;
  $ifstatement
  $(ModelNode)=$(ModelNode)
      .getParentModelNode();
}
```

(k)

```
Document doc=Jsoup.connect(URL).
    userAgent("Mozilla").
    header("Accept","text/html").
    get();
```

(l)

Allamanis and Sutton, FSE 2014

Bayesian Melding

[Poole and Raftery, 2000]

Deterministic simulation $\tau = f(S)$.

e.g., S contains population at time 0 and growth rate

τ is population at time T

Two sources of information about τ

1. Information about S, which implies information about τ via f

$$p_S(S) \xrightarrow{f} p^*_\tau(\tau)$$

2. Direct measurements of τ

$$p_\tau(\tau)$$

How to combine? Geometric average.

$$\widetilde{p}_\tau(\tau) \propto p^*_\tau(\tau)^\alpha p_\tau(\tau)^{1-\alpha}$$

How does this yield a distribution on S?

$$\widetilde{p}_S(S) = c_\alpha p_S(S) \left(\frac{p_\tau(f(S))}{p^*_\tau(f(S))} \right)^{1-\alpha}$$

Latent Bayesian Melding

[Zhong, Goddard, Sutton, NIPS 2015]

Instead, model using a latent variable

$$p_\tau(\tau) = \int p_\xi(\xi) p(\tau|\xi) d\xi.$$

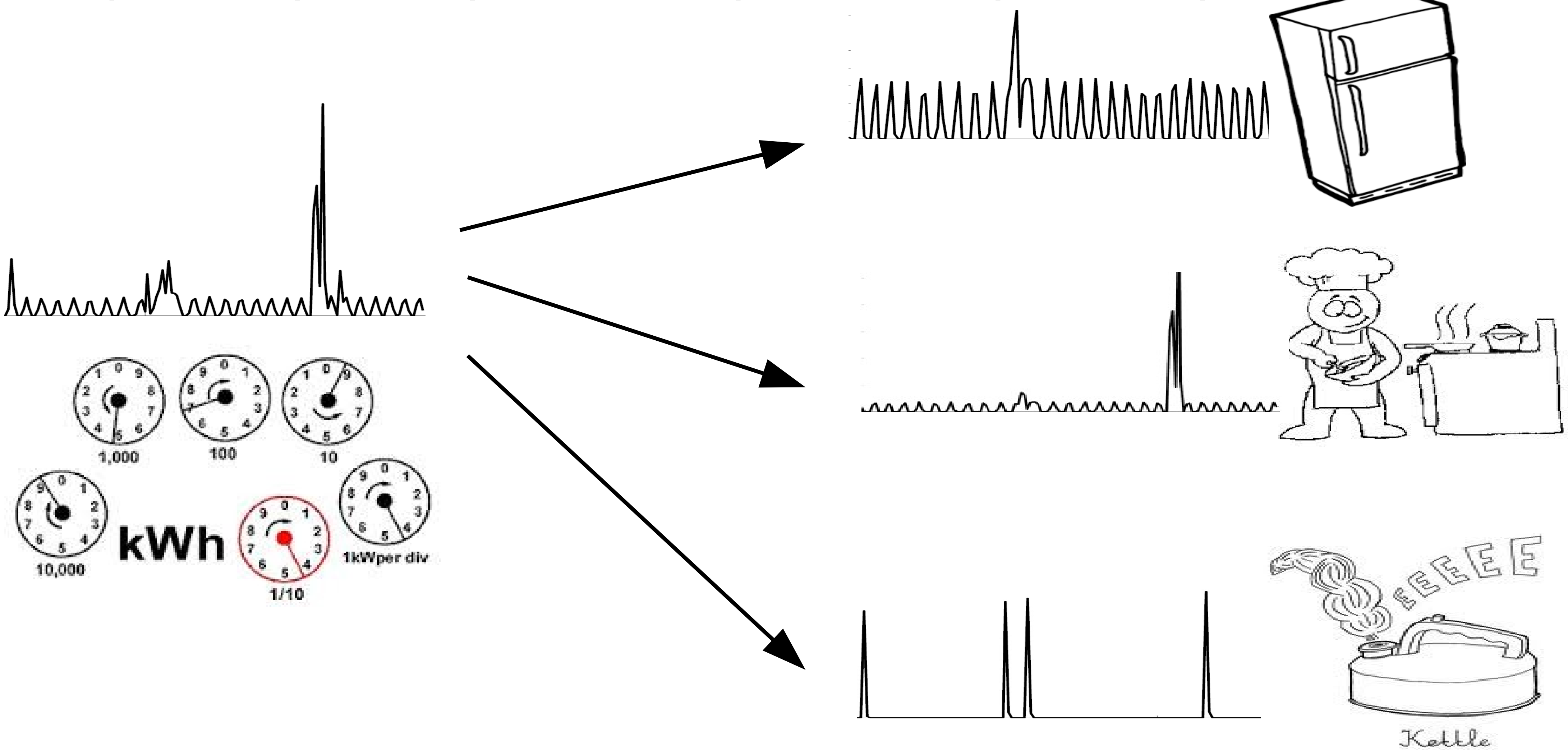
Ex of latent: Maybe different subpopulations have different growth rates.

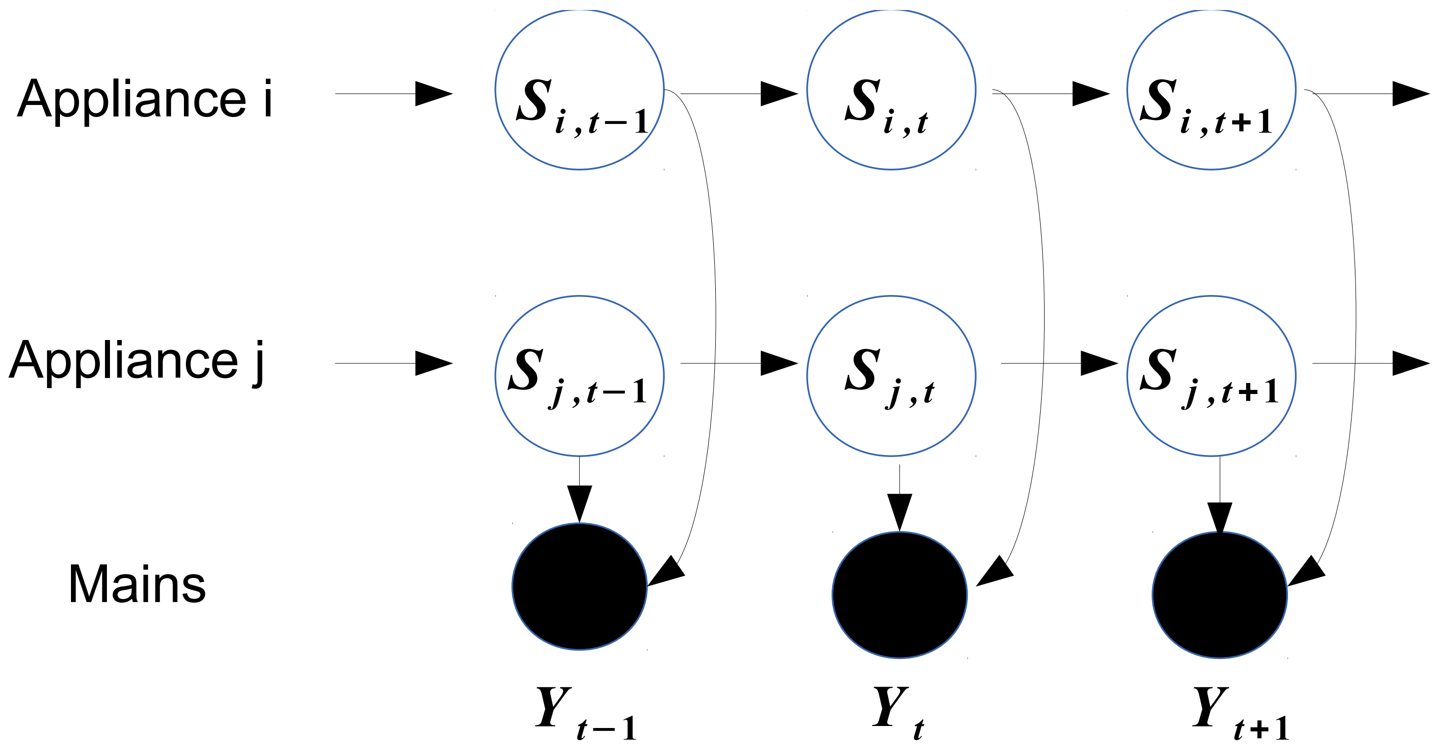
Following the standard BM approach intractable, so we take an approximation:

$$\widetilde{p}_S(S) \approx \max_\xi \widetilde{p}_{S,\xi}(S,\xi) = \max_\xi c_\alpha p_S(S) \left(\frac{p_\tau(f(S)|\xi)p(\xi)}{p^*_\tau(f(S))} \right)^{1-\alpha}$$

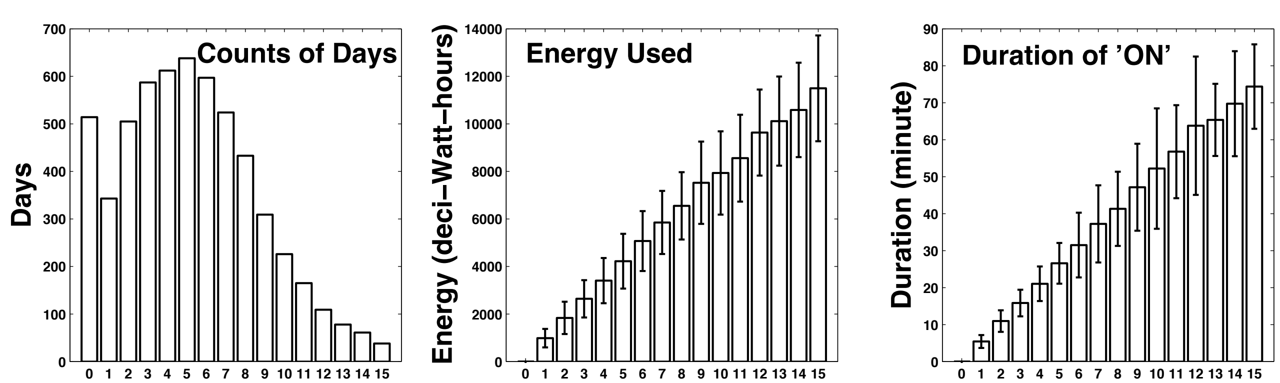
This yields an integer linear program that we can relax (for our application)

Mains (Mixture) | **Disaggregation (Separation)** | **Appliances (Sources)**

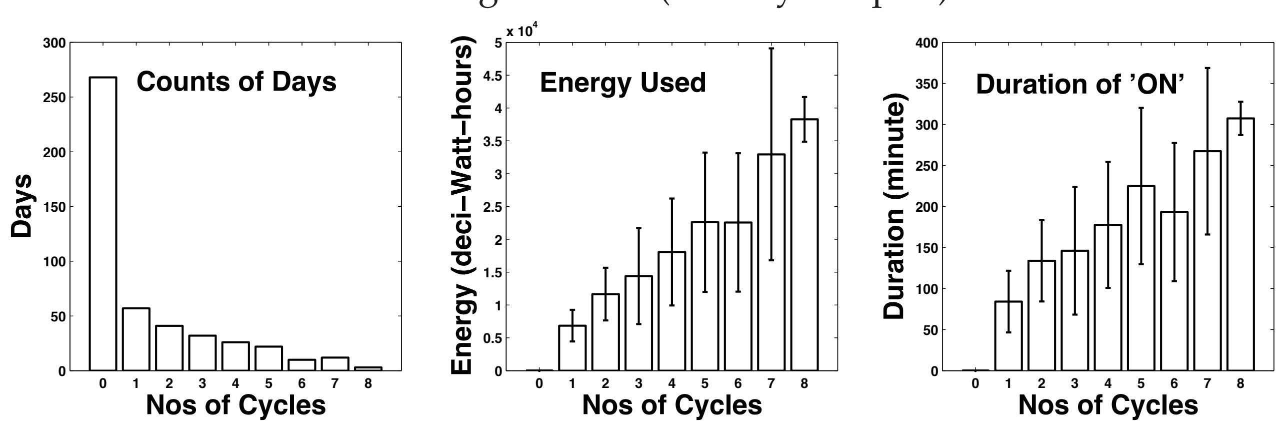




Kettle (5861 day samples)



Washing Machine (499 day samples)



Model 1

$p_S(S)$

f

count how many times appliance turns on in S

count how much energy appliance uses in S

Model 2

$$p_\tau(\tau) = \int p_\xi(\xi) p(\tau|\xi) d\xi.$$

- *Using machine learning to make programming better*
  - ML / NLP for programming languages
  - Combining program analysis with probabilistic machine learning
  - Find patterns in program executions: debugging
- *Using machine learning to make machine learning better*
  - Deep learning: Combining neural networks with prior knowledge
    - “interpretability bias”
  - Learning how to clean data
  - Interactive machine learning
  - Tools for monitoring models over time
  - Unsupervised and weakly supervised learning
- *Using machine learning to make the world better*
  - ML for computer security, NLP, sustainable energy…