Probabilistic Machine Learning (theory and practice)

Charles Sutton Introduction to Research in Data Science University of Edinburgh

New methodology



New applications

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

- Analyzing computer programs
- Practical data analysis
- Home energy demand
- Computer security

Predictive tasks

- Finding latent structure in data
- Predicting many interdependent variables

Hence: Probabilistic modelling

Hamiltonian Monte Carlo for Discrete Spaces



The Gaussian Integral Trick

$$\mathbf{S} \longrightarrow \mathbf{X}$$

$$p(\mathbf{s}) = \frac{1}{Z} \exp\left\{\mathbf{a}^T \mathbf{s} + \frac{1}{2} \mathbf{a}^T W \mathbf{a}\right\} \qquad p(\mathbf{x}|\mathbf{s}) = \frac{1}{Z} \exp\left\{\mathbf{a}^T \mathbf{s} + \frac{1}{2} \mathbf{a}^T W \mathbf{a}\right\}$$

$$p(\mathbf{x}, \mathbf{s}) \propto \exp\left\{-\frac{1}{2} \mathbf{x}^T (W + D)^{-1} \mathbf{x} + (\mathbf{a} - \frac{1}{2} \mathbf{d})^T \mathbf{s}\right\}$$

$$\mathbf{S} \qquad Si Sj$$

NIPS 2012]

Syntactic Idioms in Code

```
IfStatement
if (c != null) {
                                              expression:
                                                                                                   E
  try {
                                               c!=null
   if (c.moveToFirst()) {
                                              then:Block
     number = c.getString(
               c.getColumnIndex(
                                               TryStatement
                phoneColumn));
                                                  body:IfStatement
   }
                                                   expr:MethodInvocation
  } finally {
   c.close();
                                                     expr:var%android.database.Cursor%
  }
                                                     name:c
                                                    name:moveToFirst
. . .
                                                   then:Block
                  (a)
                                                   _number = c.getString(c.getColumnIndex(phoneColumn));
                                                  finally:Block
try {
                                                   ExpressionStatement
  if ($(Cursor).moveToFirst()) {
     $BODY$
                                                   MethodInvocation
  }
                                                      expr:var%android.database.Cursor%
} finally {
   $(Cursor).close();
                                                      name:c
}
                                                      name: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();

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

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

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

(d)

Location.distanceBetween(\$B \$(Location).getLatitude(), }fin \$(Location).getLongitude(), \$(\$...); }

(g)

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

(j)

(b)
SearchSourceBuilder builder=
getQueryTranslator().build(
 \$(ContentIndexQuery));

(e)

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

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

(h)

(k)

(c)

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

(f)

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

(i)

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

(1)

Allamanis and Sutton, FSE 2014

Possible project directions

- ML / NLP for programming languages
 - How programmers choose names
 - (Combining static program analysis with probabilistic modelling)
 - Learning graph structure: design patterns
 - Find patterns in program executions: debugging
- Visualization / Data analysis
 - "machine learning for the masses"
 - Clustering with prior knowledge: "Not that one"
 - Tools for monitoring models over time
 - Tools for error analysis
 - Combining interpretability with predictive accuracy