
A Theory of Aspects as Latent Topics

**Pierre Baldi, Cristina Lopes, Erik Linstead, Sushil Bajracharya
Donald Bren School of Information and Computer Science
University of California, Irvine**

{pfbaldi,lopes,elinstea,sbajrach}@ics.uci.edu





Overview

- Motivation
- Aspects as Latent Topics
 - Machine Learning for Concern Extraction
 - Latent Dirichlet Allocation
 - Data
 - Sourcerer
 - Vocabulary Selection
- Results
 - Scattering and Tangling in the Large
 - Scattering and Tangling in the Small
- Conclusions



Motivation

- AOP is still a controversial idea
- Hypotheses put forth by AOP have yet to be validated on the very large scale
 - **Cross-cutting concerns exist and are subject to scattering and tangling**
 - Excessive scattering and tangling are “bad” for software
 - Alternative composition mechanisms (eg. AspectJ) alleviate problems caused by cross-cutting concerns
- Advances in machine learning provide the necessary tools for such a validation
- Here we focus on empirical validation of first hypothesis
- Contributions
 - Unsupervised learning of cross-cutting concerns
 - An information-theoretic definition for scattering and tangling
 - Empirical validation across multiple scales



Learning Cross-Cutting Concerns

- Availability of Open-Source software facilitates large-scale empirical analysis of many software facets
- Recent advances in statistical text mining techniques offer new opportunities to mine Internet-scale software repositories
 - Unsupervised
 - Probabilistic
 - Proven to give better results than “traditional” methods
 - Scalable



Statistical Topic Models

- Statistical Topic Models represent documents as probability distributions over words and topics
 - Benefits of working in probabilistic framework
 - Robust – model documents directly
 - Finding patterns is intuitive and easily automated
- Active research area yielding exciting results
 - Traditional Text
 - Source Code (Linstead et al. ASE 2007, NIPS 2007)




Latent Dirichlet Allocation (LDA)

- Blei, Ng, Jordan (2003)
- Simple “Bag of Words” approach
- Models documents as mixtures of topics (multinomial)
- Topics are distributions over words (multinomial)
- Bayesian (Symmetric Dirichlet priors)
- Well analyzed in literature

Documents as “Bags of Words”

```
public class TextMiner {  
    private List<TrainCollection>  
    private Matrix bagOfWords;  
    public void nearestNeighbor(){  
        ...  
        bagOfWords.calcCosineDistance();  
        ...  
        Random r = new Random();  
    }  
}
```



text **words**
miner **random**
matrix **calc**
nearest **cosine**
neighbor **distance**
train
collection
bag



LDA – In a nutshell

- Given a document-word matrix
 - Probabilistically determine X most likely topics
 - For each topic determine Y most likely words
 - Do it without human intervention
 - Humans do not supply hints for topic list
 - Humans do not tune algorithm on the fly
 - No need for iterative refinement
- Output
 - Document-Topic Matrix
 - Topic-Word Matrix



Aspects as Latent Topics

- Unification of “topics” in text with “concerns” in software
 - **A CONCERN IS A LATENT TOPIC**
- Syntax and convention differentiates natural and programming languages, but:
 - At most basic level a source file is still a document
 - Tokens in source code still define a vocabulary
- Probability distributions of topics over files and files over topics allow for precise measurement of scattering and tangling, respectively

Measuring Scattering

- If the distribution of a topic, t , across modules $m_0 \dots m_n$ is given by $p^t = (p_0^t \dots p_n^t)$ then scattering can be measured by the entropy

$$H(p^t) = -\sum_k p_k^t \log(p_k^t)$$

- Can normalize by dividing by $\log(n)$
 - $H(p^t) = 0$ denotes a concern assigned to only one source file
 - $H(p^t) = 1$ denotes a concern uniformly distributed across source files
- **AN ASPECT IS A LATENT TOPIC WITH HIGH SCATTERING ENTROPY**

	t1	t2	t3	tn
d1	0	0	8	0
d2	1	0	8	5
d3	8	8	8	8
d4	3	0	8	1
d5	15	0	8	2
dn	12	0	8	4

Measuring Tangling

- If the distribution of a module, m , across concerns $t_0 \dots t_n$ is given by $q^m = (q^m_0 \dots q^m_r)$ then scattering can be measured by the entropy

$$H(q^m) = -\sum_k q^m_k \log(q^m_k)$$

- Can normalize by dividing by $\log(r)$
 - $H(q^m) = 0$ denotes a file assigned to only one concern
 - $H(q^m) = 1$ denotes a file uniformly distributed across concerns

	t1	t2	t3	tn
d1	0	0	8	0
d2	1	0	8	5
d3	8	8	8	8
d4	3	0	8	1
d5	15	0	8	2
dn	12	0	8	4



Data

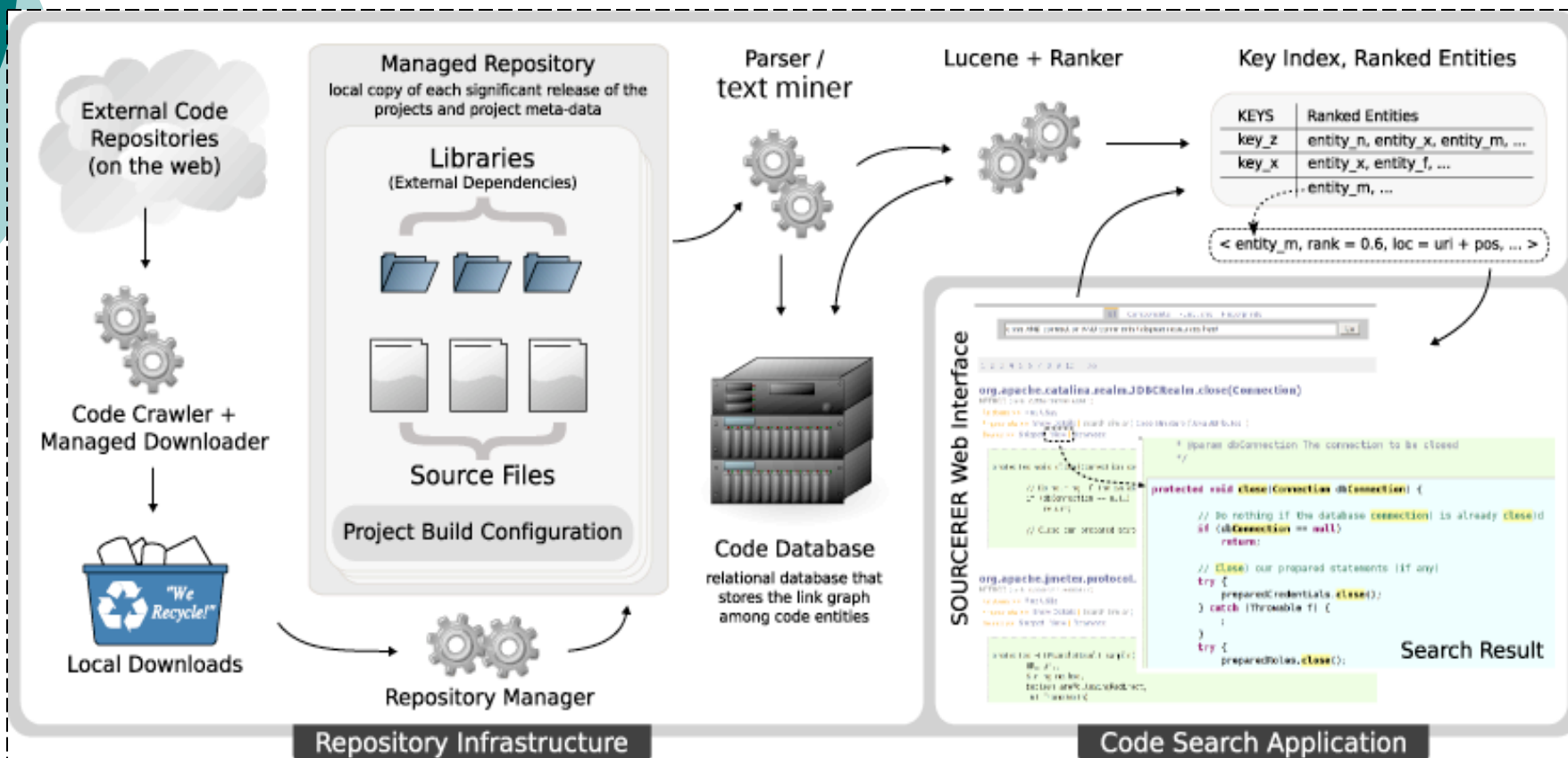
- We validate our technique at multiple scales
 - Internet-Scale
 - 4,632 open source projects constituting 38 million LOC, 366k files, and 426k classes
 - Leverage Sourcerer infrastructure
 - Individual Projects
 - JHotDraw
 - PDFBox
 - Jikes
 - JNode
 - CoffeeMud



Sourcerer

- UCI ICS project designed to:
 - Index publicly available source and provide fast search and mining
 - Leverage data to better understand code, facilitate reuse, provide tools for real-world software development
 - Explore new avenues for mining software
- Current Version
 - ~12k open source projects (4,632 with source code)
 - Focused on Java language as proof of concept
- Publicly Available
 - <http://sourcerer.ics.uci.edu>

Sourcerer Architecture





Vocabulary Selection

- Vocabulary size affects interpretability of topics extracted by LDA
 - Code as plain text yields noisy results

```
public class TextMiner {  
    private List trainCollection;  
    private Matrix bagOfWords;  
    public void nearestNeighbor(){  
        ...  
        bagOfWords.calcCosineDistance();  
        ...  
        Random r = new Random();  
    }  
}
```



Vocabulary Selection

- Vocabulary size affects interpretability of topics extracted by LDA
 - Code as plain text yields noisy results

```
public class TextMiner {  
    private List trainCollection;  
    private Matrix bagOfWords;  
    public void nearestNeighbor(){  
        ...  
        bagOfWords.calcCosineDistance();  
        ...  
        Random r = new Random();  
    }  
}
```




Vocabulary Selection

- Vocabulary size affects interpretability of topics extracted by LDA
 - Code as plain text yields noisy results

```
public class TextMiner {  
    private List trainCollection;  
    private Matrix bagOfWords;  
    public void nearestNeighbor() {  
        ...  
        bagOfWords.calcCosineDistance();  
        ...  
        Random r = new Random();  
    }  
}
```



Vocabulary Selection

- Vocabulary size affects interpretability of topics extracted by LDA
 - Code as plain text yields noisy results

```
public class TextMiner {  
    private List trainCollection;  
    private Matrix bagOfWords;  
    public void nearestNeighbor() {  
        ...  
        bagOfWords.calcCosineDistance();  
        ...  
        Random r = new Random();  
    }  
}
```



Vocabulary Selection

- Vocabulary size affects interpretability of topics extracted by LDA
 - Code as plain text yields noisy results

```
public class TextMiner {  
    private List trainCollection;  
    private Matrix bagOfWords;  
    public void nearestNeighbor() {  
        ...  
        bagOfWords.calcCosineDistance();  
        ...  
        Random r = new Random();  
    }  
}
```



Vocabulary Selection

- Vocabulary size affects interpretability of topics extracted by LDA
 - Code as plain text yields noisy results

```
public class TextMiner {  
    private List trainCollection;  
    private Matrix bagOfWords;  
    public void nearestNeighbor() {  
        ...  
        bagOfWords.calcCosineDistance();  
        ...  
        Random r = new Random();  
    }  
}
```

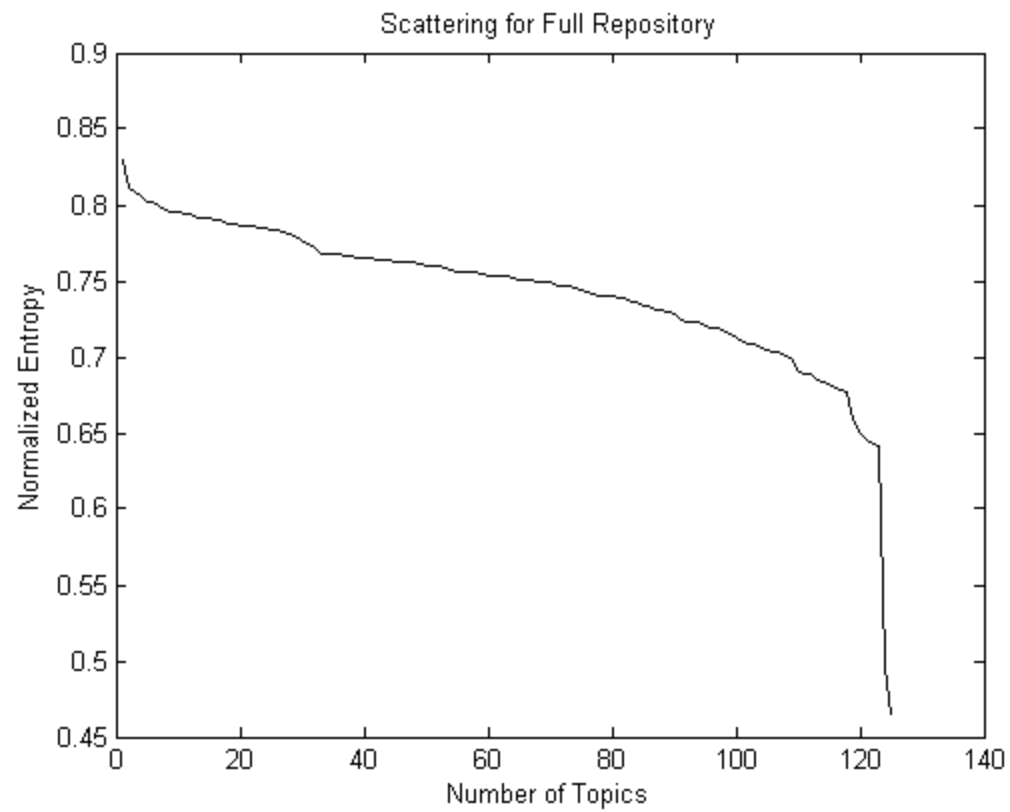


Scattering in the Large

- Many prototypical examples for AOP
- Cross-cutting found at multiple magnitudes

Concern	Extracted Topic	Entropy
String Processing	'string case length width substring'	.801
Exception Handling	'throwable trace stack print method'	.791
Concurrency	'thread run start stop wait'	.767
XML	'element document attribute schema child'	.749
Authentication	'user group role application permission'	.745
Web	'request servlet http response session'	.723
Database	'sql object fields persistence jdbc'	.677
Plotting	'category range domain axis paint'	.641

Scattering Visualization





Scattering in the Small: JHotDraw

Topic	Entropy
'instance test tear down vault'	0.813075061
'create factory collections map from'	0.722463637
'point move box index start'	0.71436202
'storable read write input output'	0.650160953
'list next has iterator add'	0.638290561
'polygon point internal chop count'	0.46080295
'size selected frame frames dimension'	0.43364049
'shape geom rectangular rectangle2 hashtable'	0.353301264
'drag drop target source listener'	0.352124151
'event component size transform mouse'	0.338653373

- Notable appearance of project-specific concerns
 - In general appear to have lower scattering entropy
 - Can be controlled in part by number of topics extracted by LDA
 - In specific cases may require developer expertise to determine valid concerns versus noise



Scattering in the Small: Jikes

Topic	Entropy
'next has element enumeration elements'	0.699351996
'buffer check empty char insert'	0.661522459
'print stream println writer total'	0.636898546
'hash map iterator next add'	0.636035451
'type array reference code resolved'	0.635043332
'cycles end time right begin'	0.486326254
'field type reflect value unchecked'	0.4684958
'short switch reference type read'	0.447104842
'sys lock unlock write socket'	0.428127362
'offset mask fits forward code'	0.346995542
'emit assembler gen reference laddr'	0.266546555



Scattering in the Small: JNode

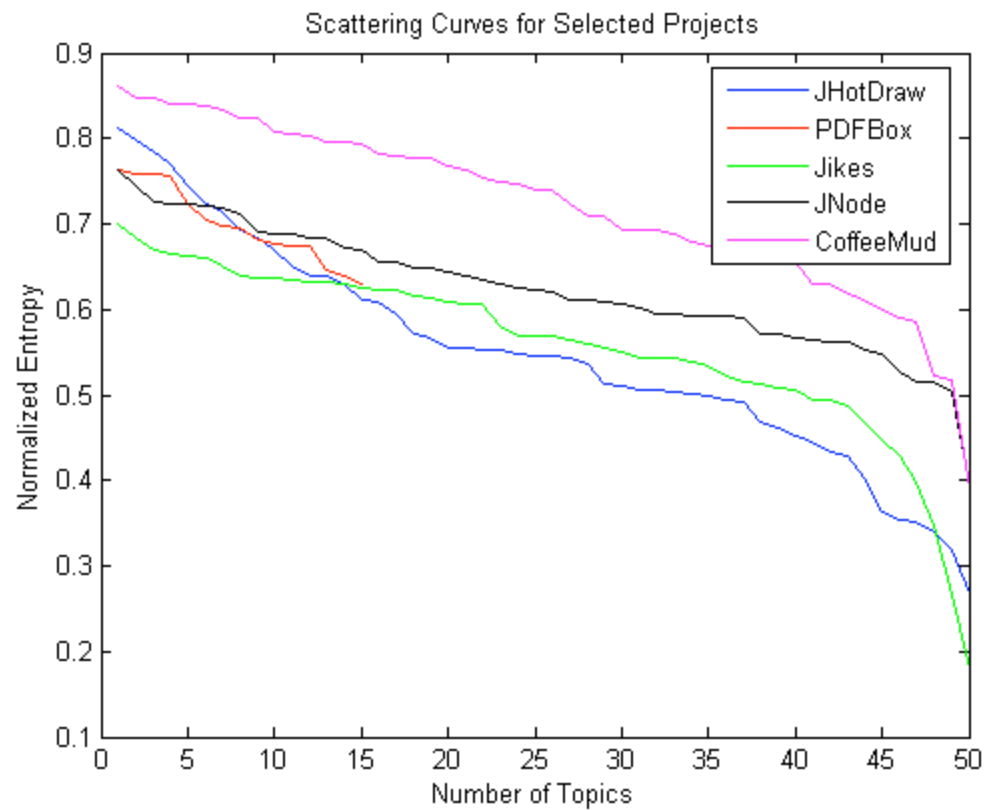
Topic	Entropy
'string length append substring tokenizer'	0.76224123
'map hash equals object value'	0.726874809
'byte array bytes arraycopy system'	0.723141514
'stream write output writer array'	0.723069203
'input read stream reader buffered'	0.718017023
'graphics color paint icon rectangle'	0.567084036
'image raster buffered create writable'	0.548839911
'time date calendar zone simple'	0.525970475
'zip entry jar plugin deflater'	0.515858882
'focus event window component listener'	0.502999404



Scattering in the Small: CoffeeMud

Topic	Entropy
'environmental mob msg location send'	0.861222835
'environmental name text vector string'	0.823602707
'vector element size add remove'	0.795135882
'mob hash environmental iterator next'	0.77667159
'string mob currency environmental shop'	0.600152681
'string channel imc send mud'	0.591218453
'string vector from xml buffer'	0.586218656
'string mob gen scr tell'	0.390775366

Scattering Visualization





Tangling in the Large

File	Entropy
org/openharmonise/rm/commands/CmdGenerateReport.java	0.8258
it/businesslogic/ireport/gui/ReportQueryDialog.java	0.7885
mail/core/org/columba/mail/imap/IMAPServer.java	0.7881
jRivetFramework/webBoltOns/ReportWriter.java	0.7869
org/lnicholls/galleon/apps/musicOrganizer/MusicOrganizer.java	0.7664
doctorj-5.0.0/org/incava/java/ASTNestedClassDeclaration.java	0.3379
nfop/fo/properties/FontSelectionStrategy.java	0.2275
net/sf/farrago/namespace/jdbc/MedJdbcColumnSet.java	0.2275
com/planet_ink/coffee_mud/Exits/Door.java	0.0
buoy/event/FocusLostEvent.java	0.0

- Full matrix available from supplementary materials page
 - 366,287 x 125
 - 72MB (compressed)



Tangling in the Small

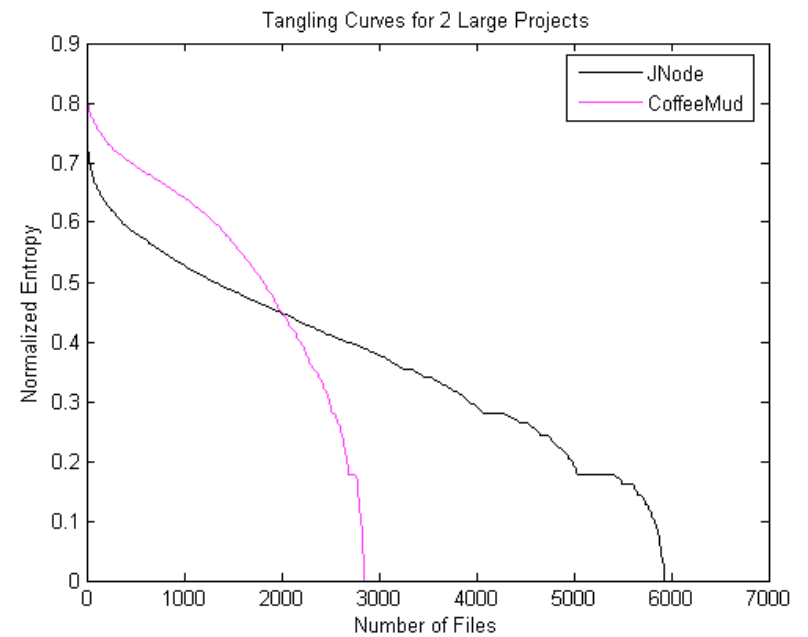
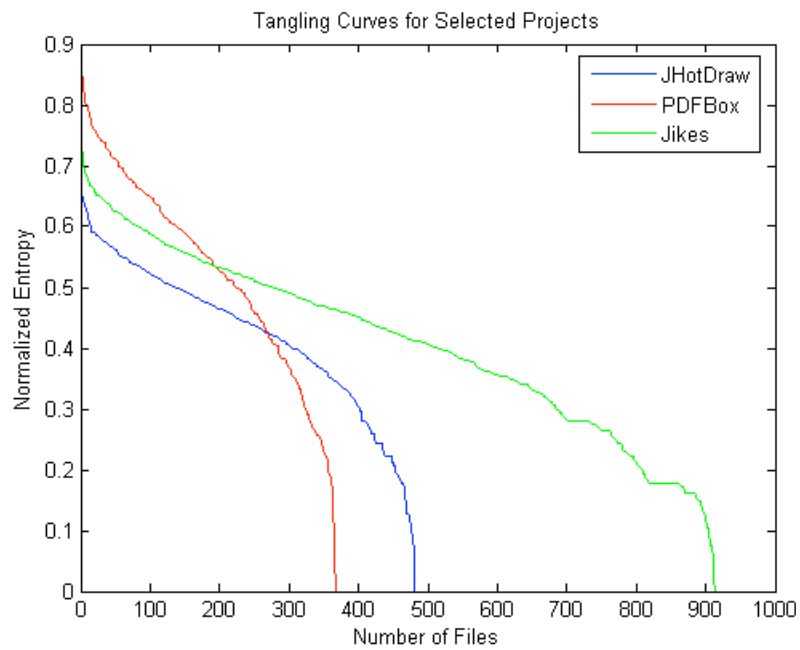
JHotDraw

File	Entropy
BouncingDrawing.java	0.6650
SingleFigureEnumeratorTest.java	0.6538
URLTool.java	0.6449
UndoRedoActivity.java	0.1000
CommandCheckBoxMenuItem.java	0.0892
JHotDrawException.java	0.0831

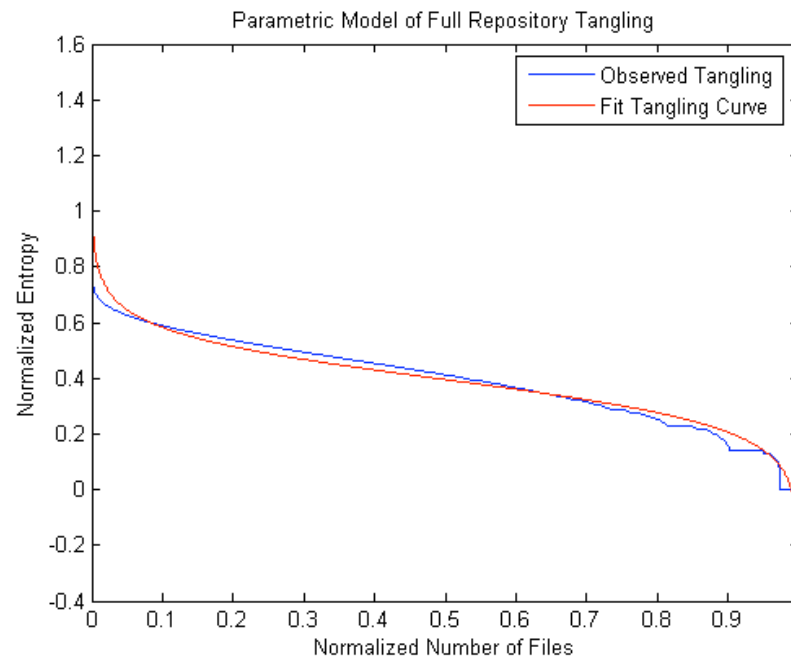
Jikes

File	Entropy
DebuggerThread.java	0.6932
TraceBuffer.java	0.6845
VM_Process.java	0.6736
VM_Listener.java	0.0693
PPC_Disassembler.java	0.0554
VM_Constants.java	0.0

Tangling Visualization



A Parametric Model of Tangling?



- Inverse sigmoidal behavior noted in tangling
- Fit simple 2 parameter model to data

$$f(x) = a * \ln((1/x)-1) + b$$

- R-Square of .947
- Standard deviation of .024



Comparison to Other Methods

- Validation for Internet-scale repository challenging
- Individual projects exist which make good baselines
- JHotDraw
 - Compared to fan-in/fan-out, identifier analysis, dynamic analysis, manual analysis, and mining code revisions
 - What aspects are identified?
 - To what degree are scattering and tangling observed?
 - General agreement with our LDA-based technique in all cases



Conclusions

- Statistical machine learning techniques make additional progress in Aspect Mining
- LDA effectively extracts concerns from arbitrarily large repositories
 - Unsupervised
 - No pre-conceived notion of what an Aspect is
 - A Concern is a latent topic in source code
- Statistical techniques allow for precise measurement of scattering and tangling using information theory
 - An Aspect is a concern with high scattering entropy
- Significant agreement with other aspect mining methods



BACKUP

Current/Future Work

- Validate Second AOP Hypothesis
 - Are scattering and tangling *truly* “bad” for real-world software?
- Apply LDA to Software Evolution
 - Concern trends over release histories

