USING CLASSIFIER CASCADES FOR SCALABLE E-MAIL CLASSIFICATION

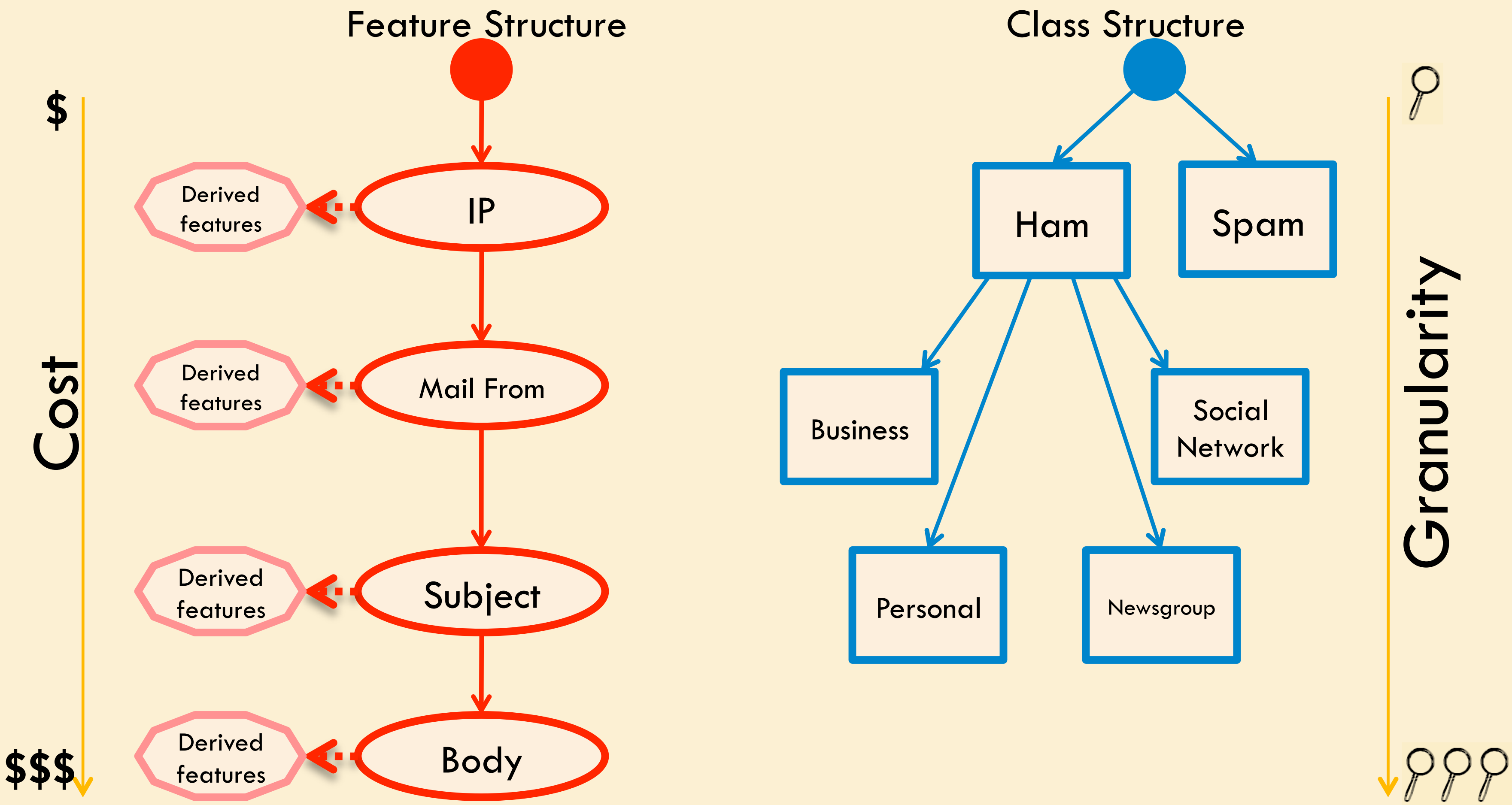
Jay Pujara	jay@cs.umd.edu
Hal Daumé III	me@hal3.name
Lise Getoor	getoor@cs.umd.edu

9/1/2011

Building a scalable e-mail system

- Goal: Maintain system throughput across conditions
- Varying conditions
  - Load varies
  - Resource availability varies
  - Task varies
- Challenge: Build a system that can adapt its operation to the conditions at hand

Problem structure informs scalable solution



Important facets of problem

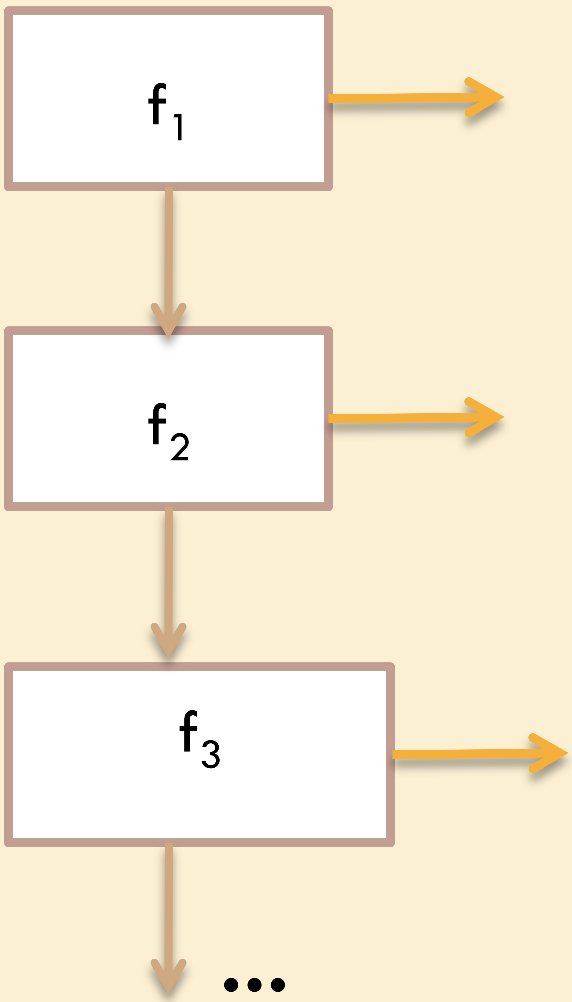
- Structure in input
  - Features may have an order or systemic dependency
  - Acquisition costs vary: cheap or expensive features
- Structure in output
  - Labels naturally have a hierarchy from coarse-to-fine
  - Different levels of hierarchy have different sensitivities to cost
- Exploit structure during classification
- Minimize costs, minimize error

Two overarching questions

- When should we acquire features to classify a message?
- How does this acquisition policy change across different classification tasks?

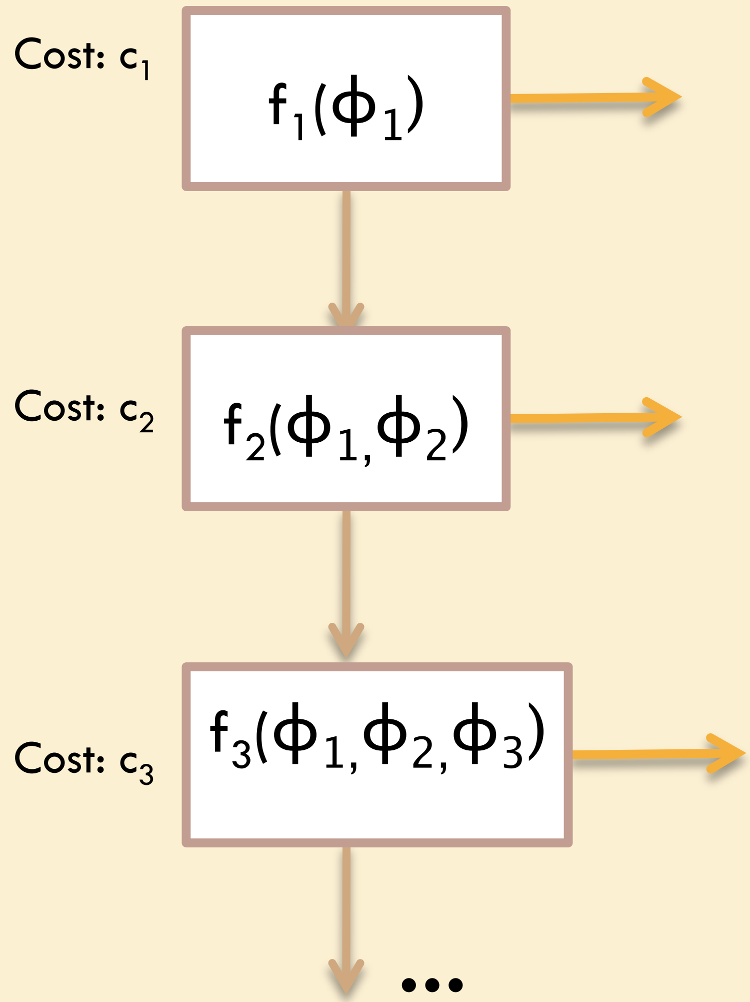
- Classifier Cascades can answer both questions!

Introducing Classifier Cascades



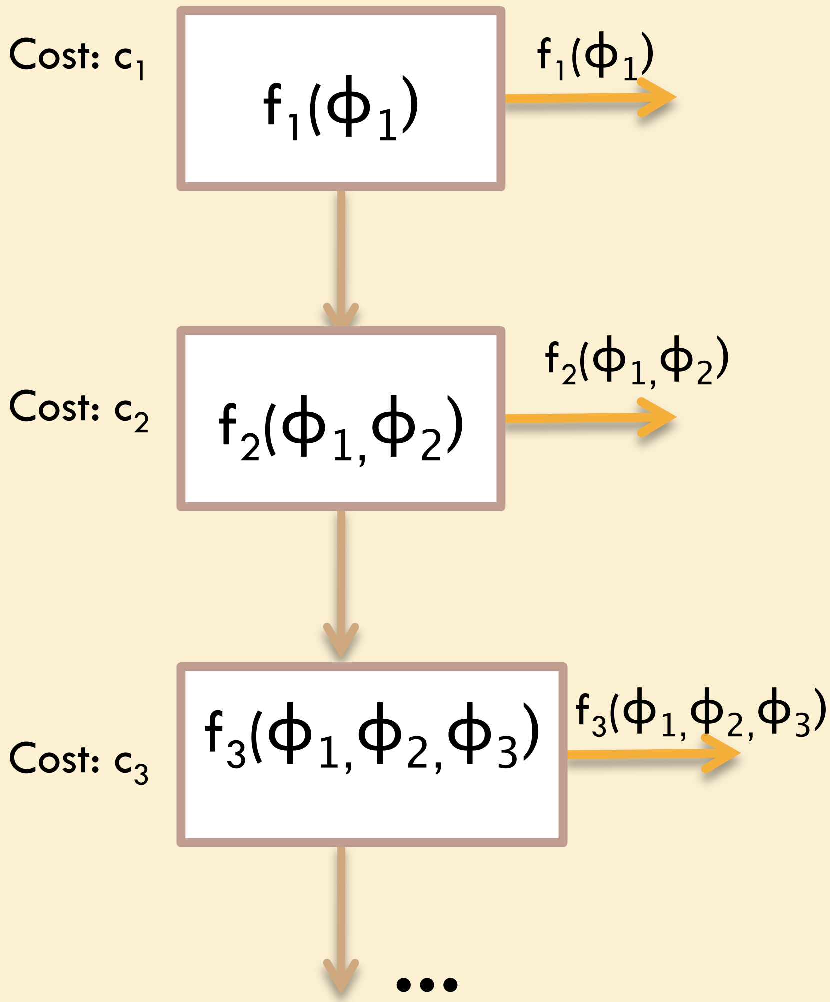
- Series of classifiers:
  $f_1, f_2, f_3 \ldots f_n$

Introducing Classifier Cascades



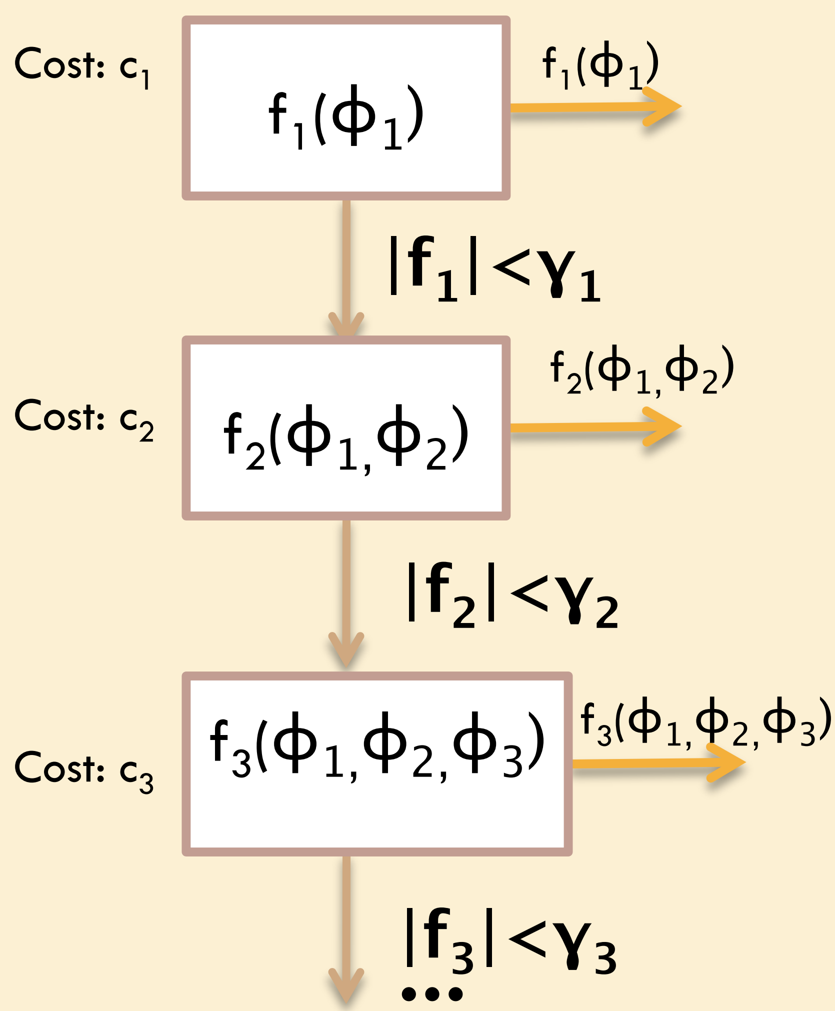
- Series of classifiers:
  $f_1, f_2, f_3 \ldots f_n$
- Each classifier operates on different, increasingly expensive sets of features (ϕ) with costs $c_1, c_2, c_3 \ldots c_n$

Introducing Classifier Cascades



- Series of classifiers: $f_1, f_2, f_3 \ldots f_n$
- Each classifier operates on different, increasingly expensive sets of features (ϕ) with costs $c_1, c_2, c_3 \ldots c_n$
- Classifier outputs a value in [-1,1], the margin or confidence of decision

Introducing Classifier Cascades



- Series of classifiers:
  $f_1, f_2, f_3 \ldots f_n$
- Each classifier operates on different, increasingly expensive sets of features (ϕ) with costs $c_1, c_2, c_3 \ldots c_n$
- Classifier outputs a value in [-1,1], the margin or confidence of decision
- γ parameters control the relationship of classifiers

Optimizing Classifier Cascades

- Loss function: $L(y, \mathcal{F}(\mathbf{x}))$ – errors in classification

- Minimize loss function, incorporating cost
  - Cost-constraint with budget (load-sensitive):

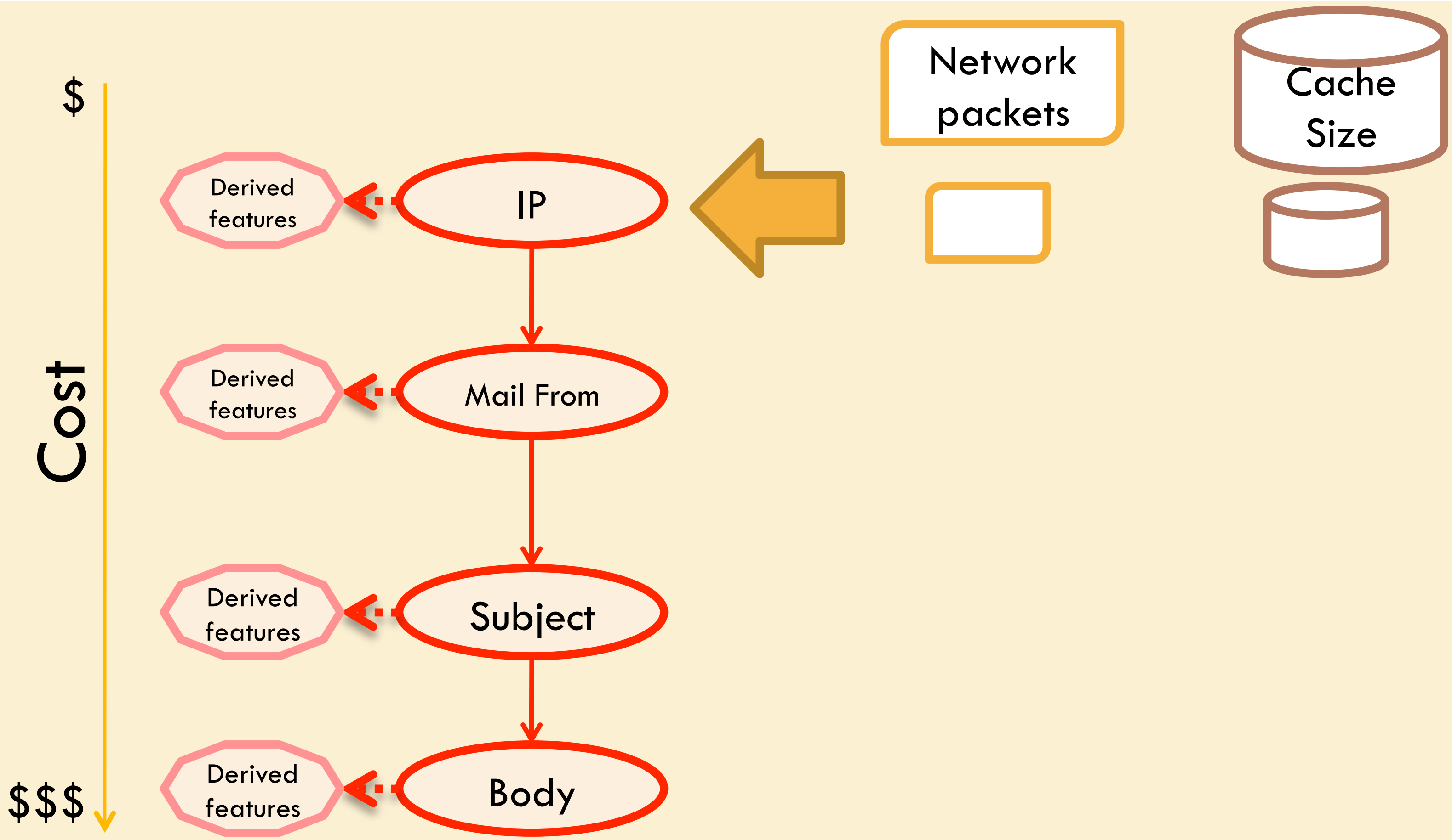
    $$\min \Sigma_{(\mathbf{x},y)\in D} L(y, \mathcal{F}(\mathbf{x})) \text{ s.t. } \mathcal{C}(\mathbf{x}) < B$$
  - Cost Sensitive loss function (granular):

    $$\min \Sigma_{(\mathbf{x},y)\in D} L(y, \mathcal{F}(\mathbf{x})) + \lambda \mathcal{C}(\mathbf{x})$$

- Use grid-search to find optimal **γ** parameters

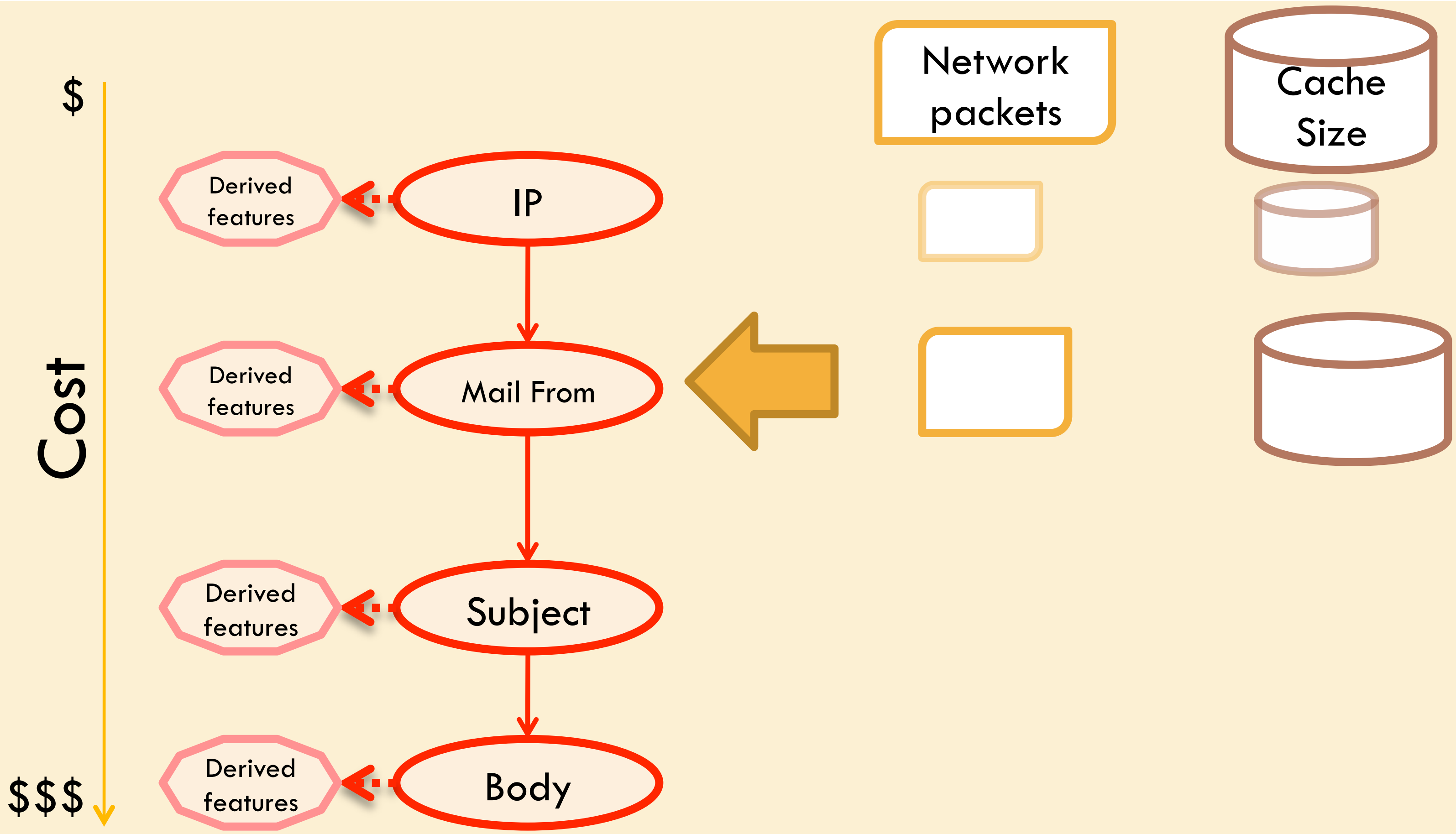
Load-Sensitive Classification

Features have costs & dependencies



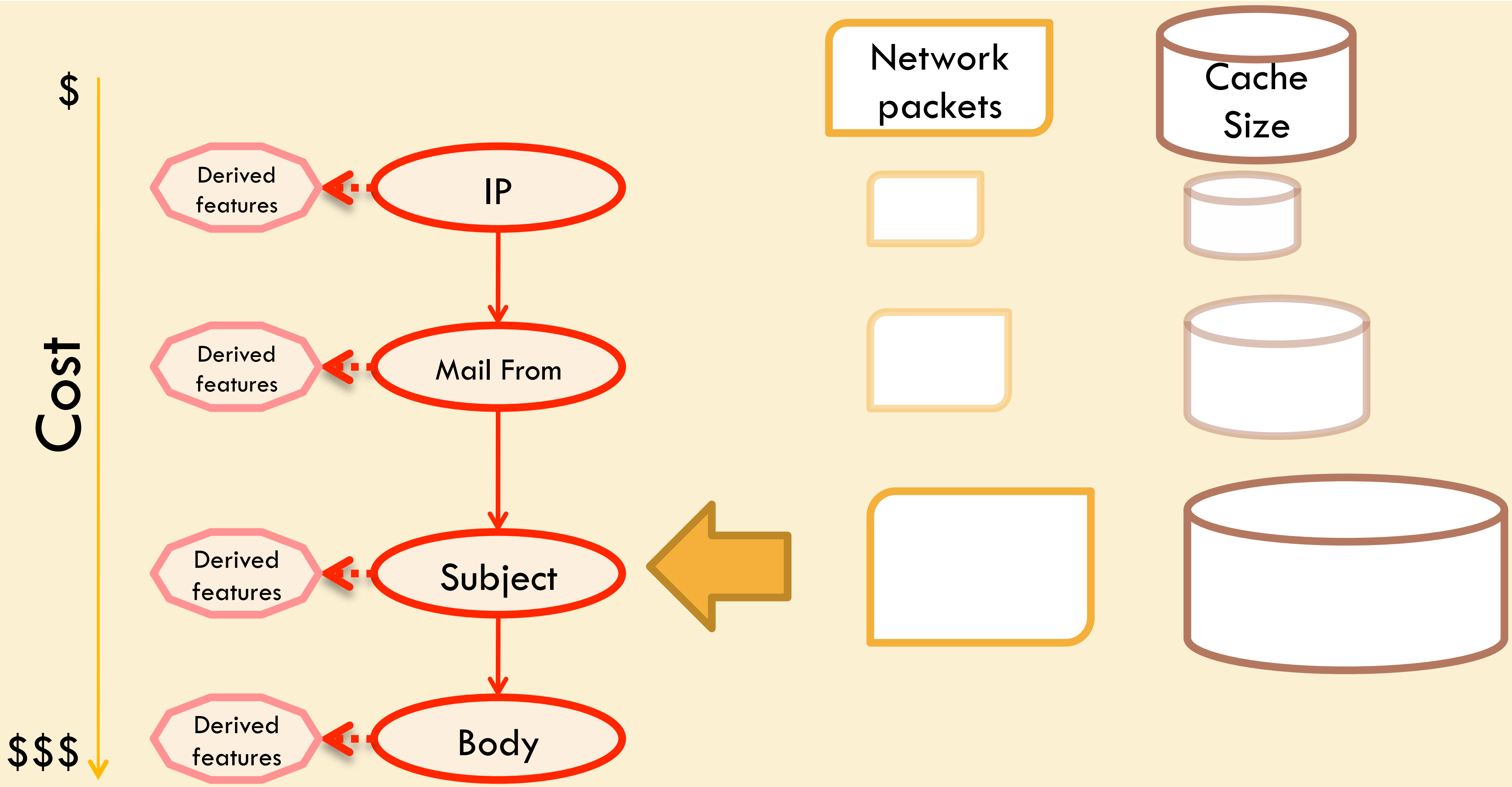
IP is known at socket connect time, is 4 bytes in size

Features have costs & dependencies



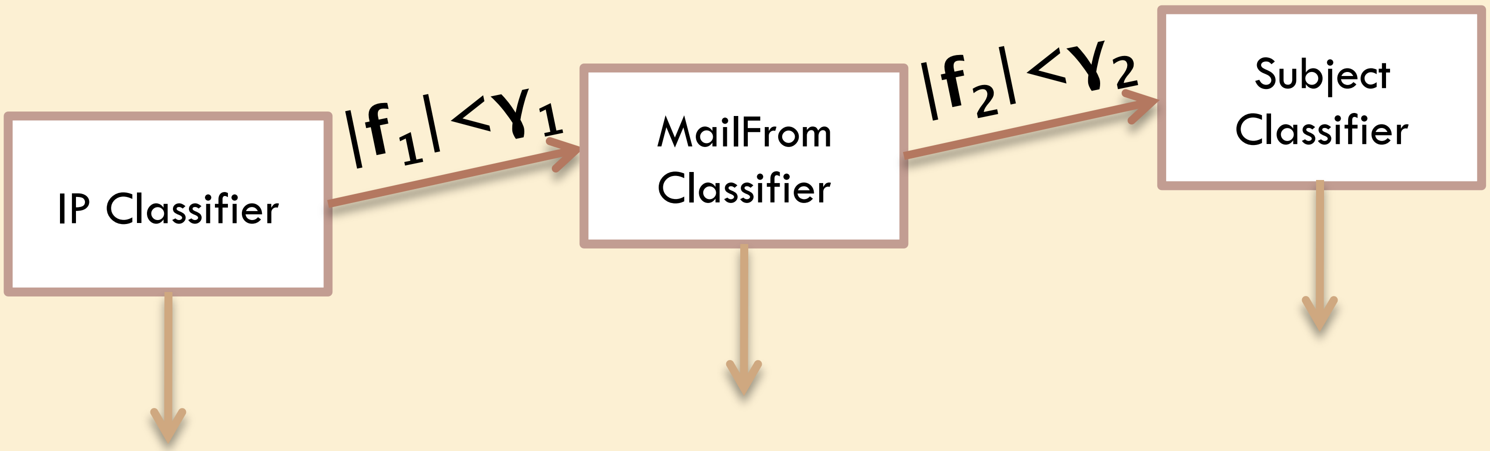
The Mail From is one of the first commands of an SMTP conversation
From addresses have a known format, but higher diversity

Features have costs & dependencies



The subject, one of the mail headers, occurs after a number of network exchanges. Since the subject is user-generated, it is very diverse and often lacks a defined format

Load-Sensitive Problem Setting



- Train IP, MailFrom, and Subject classifiers
- For a given budget, **B**, choose γ_1, γ_2 that minimize error within **B**
- Constraint: $C(x) < \mathbf{B}$

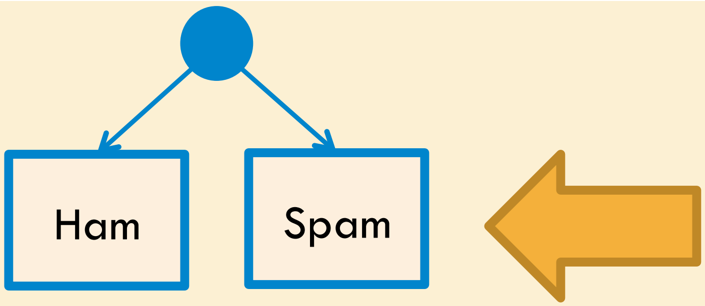
Load-Sensitive Challenges

- Overfitting model when choosing γ_1, γ_2
- Train-time costs underestimated versus test-time performance
- Use a regularization constant Δ
  - Sensitive to cost variance (σ)
  - Accounts for variability
- Revised constraint: $C(x) + \Delta\sigma < \mathbf{B}$

18

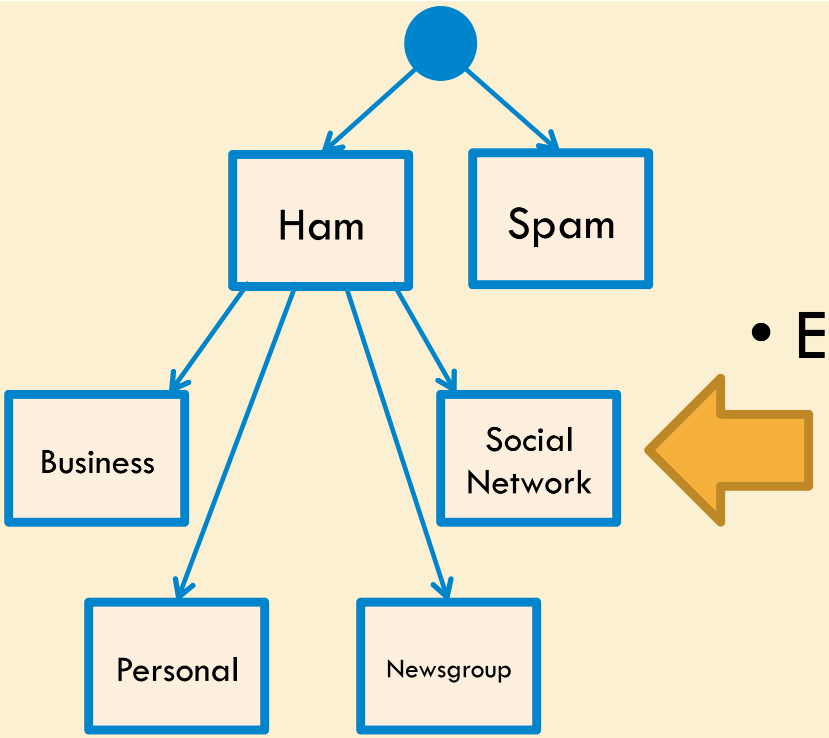
Granular Classification

E-mail Challenges: Spam Detection



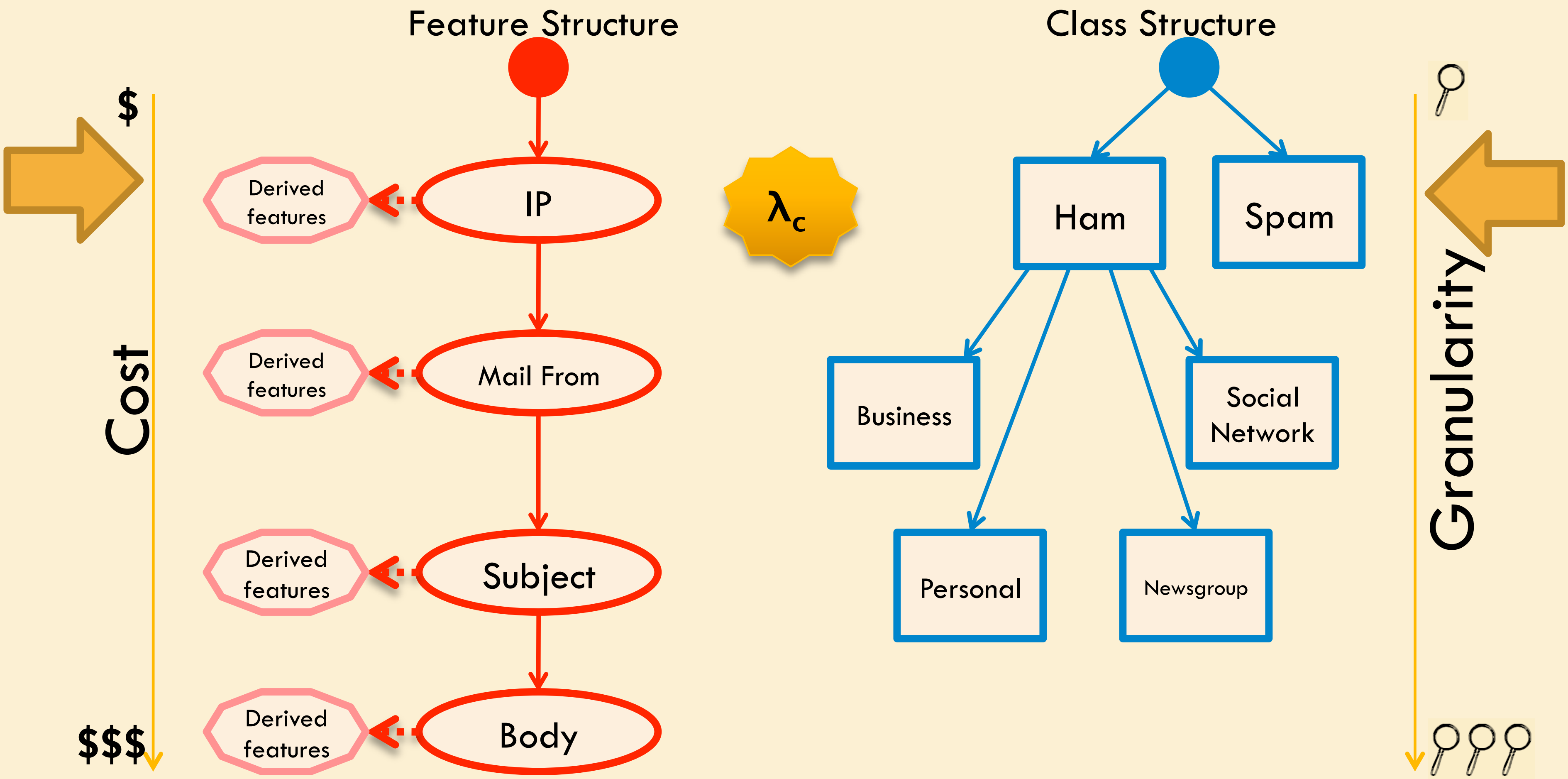
- Most mail is spam
- Billions of classifications
- Must be incredibly fast

E-mail Challenges: Categorizing Mail

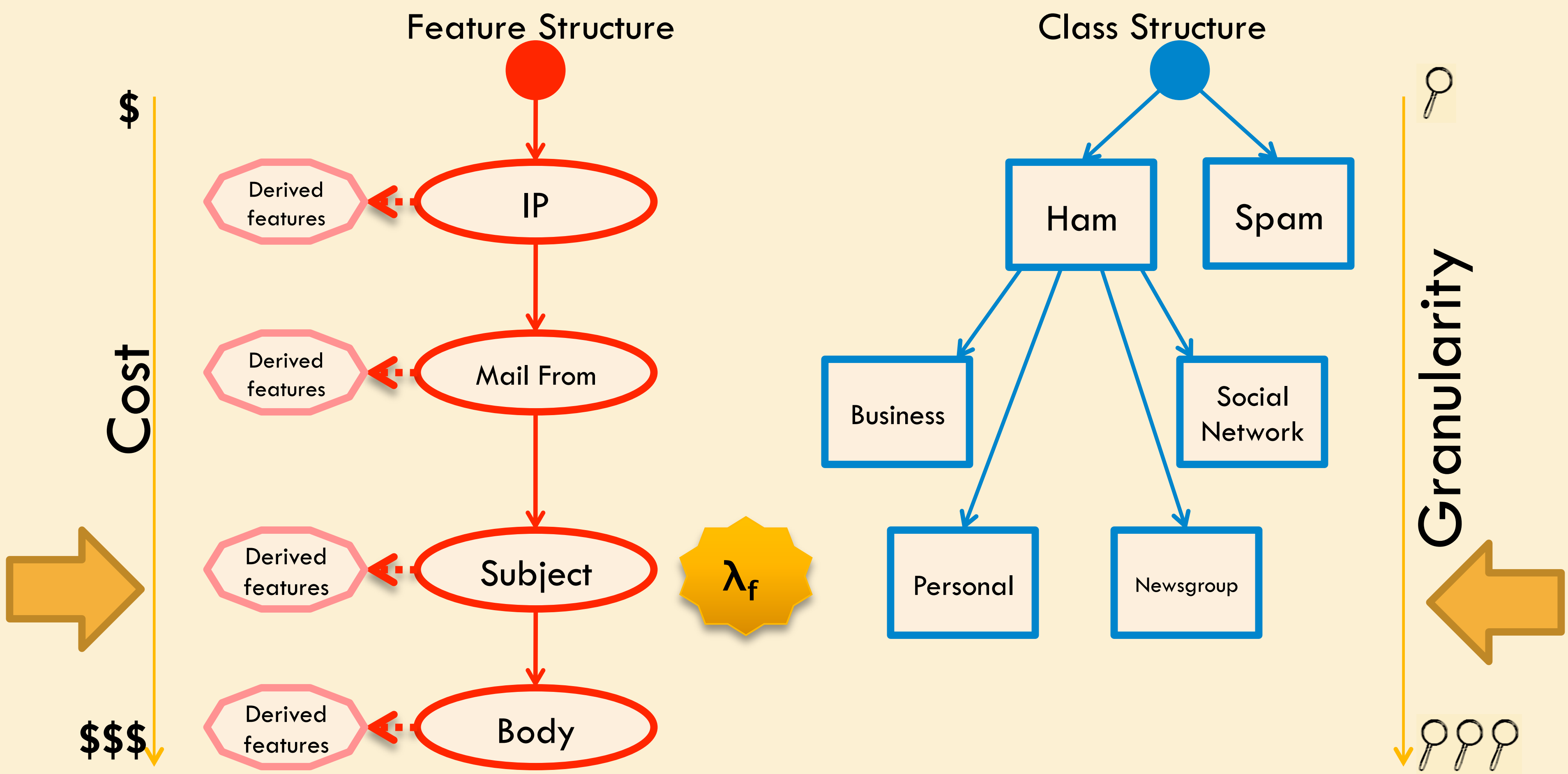


- E-mail does more, tasks such as:
  - Extract receipts, tracking info
  - Thread conversations
  - Filter into mailing lists
  - Inline social network response
- Computationally intensive processing
- Each task applies to one class

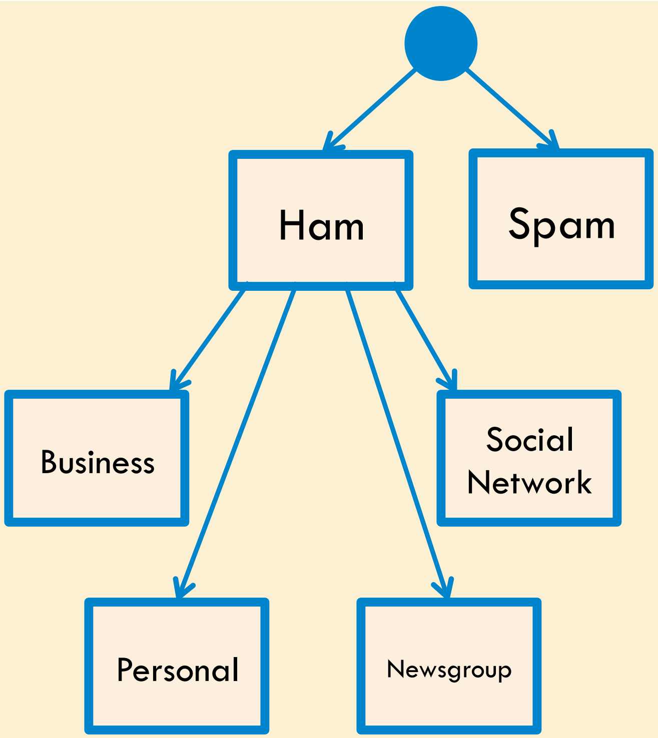
Coarse task is constrained by feature cost



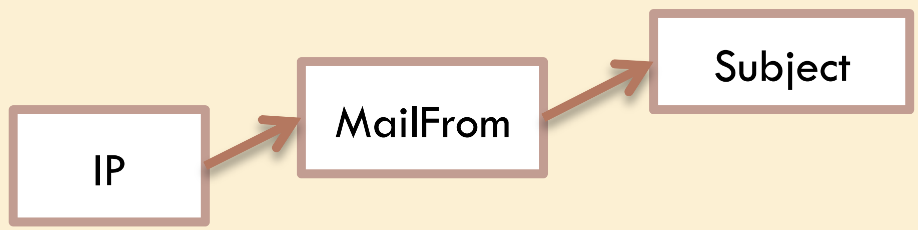
Fine task is constrained by misclassification cost



Granular Classification Problem Setting



$$L(y, h(x)) + \lambda_c C(x)$$



$$L(y, h(x)) + \lambda_f C(x)$$

- Two separate models for different tasks, with different classifiers and cascade parameters
- Choose γ_1, γ_2 for each cascade to balance accuracy and cost with different tradeoffs λ

Experimental Results

Experimental Setup: Overview

- Two tasks: load-sensitive & granular classification
- Two datasets: Yahoo! Mail corpus and TREC-2007
  - Load-sensitive uses both datasets, granular uses only Yahoo!
- Results are L1O, 10-fold CV with **bold** values significant ($p<.05$)
- Cascade stages use MEGAM MaxEnt classifier

Experimental Setup: Yahoo! Data

Class	Messages
Spam	531
Business	187
Social Network	223
Newsletter	174
Personal/Other	102

Feature	Cost
IP	.168
MailFrom	.322
Subject	.510

- Data from 1227 Yahoo! Mail messages from 8/2010
- Feature costs calculated from network + storage cost

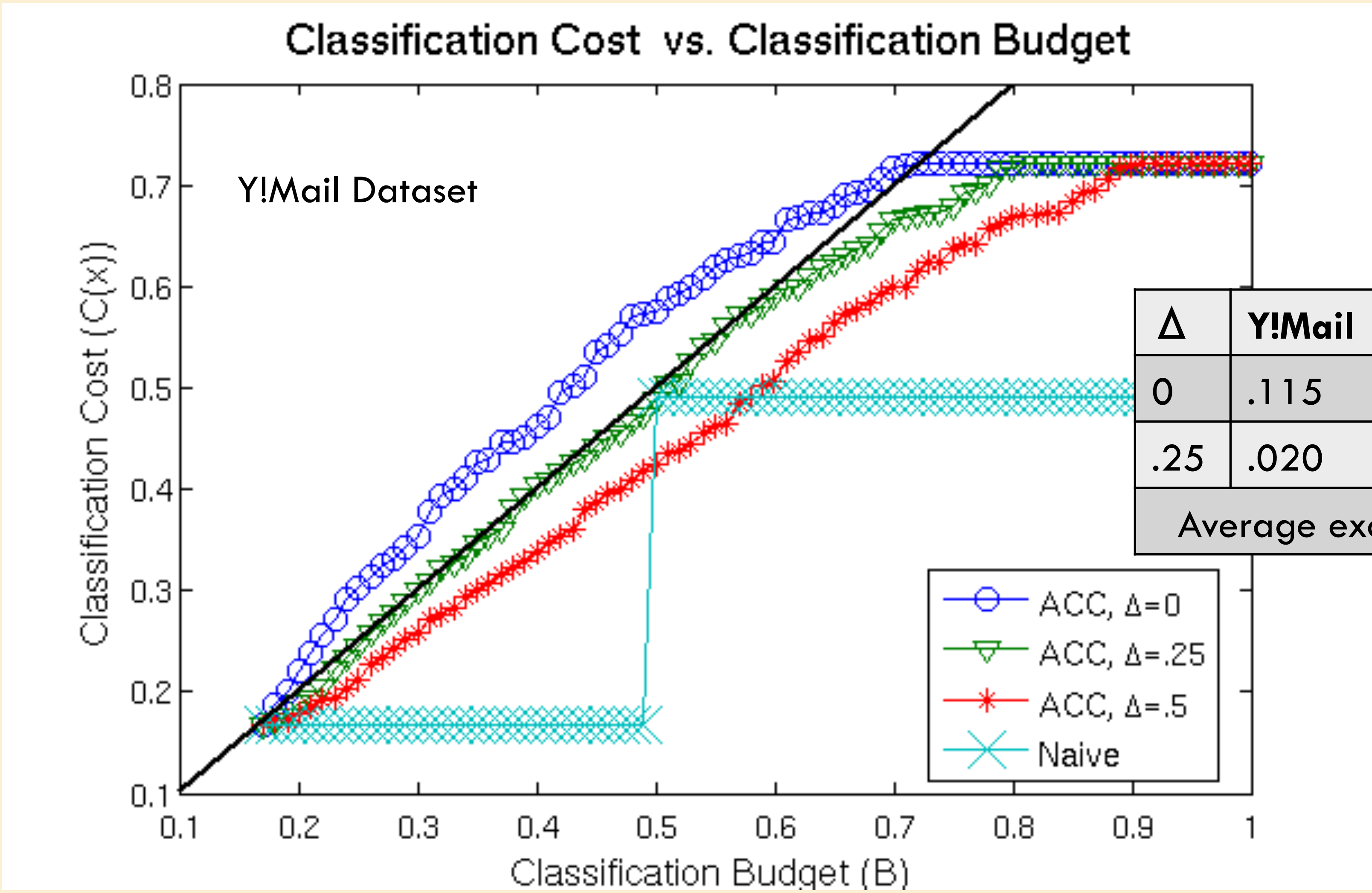
Experimental Setup: TREC data

Class	Messages
Spam	39055
Ham	8139

- Data from TREC-2007 Public Spam Corpus, 47194 messages
- Use same feature cost estimates

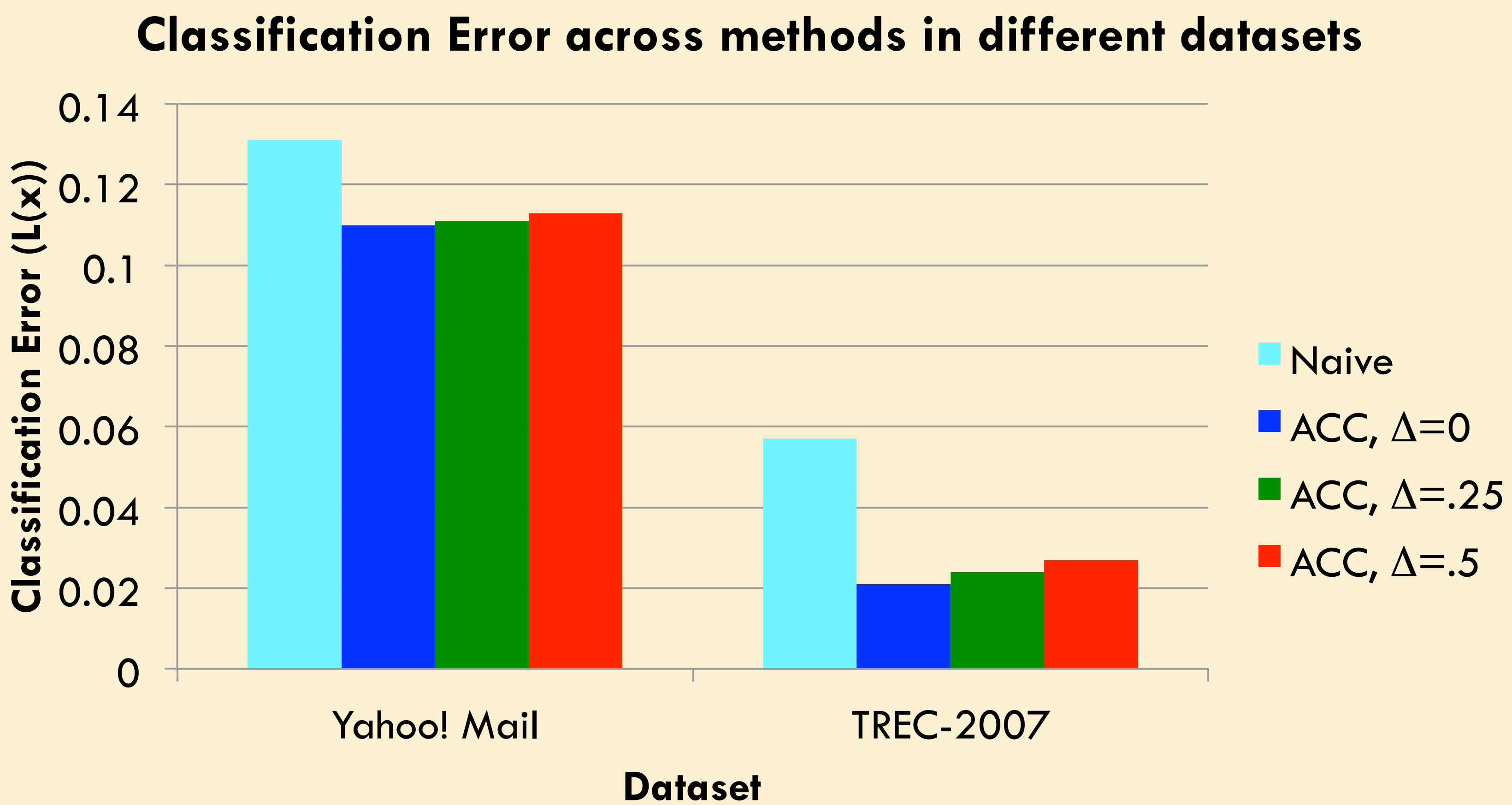
Results: Load-Sensitive Classification

Regularization prevents cost excesses



Δ	Y!Mail	TREC
0	.115	.059
.25	.020	0.00
Average excess cost		

Results: Load-Sensitive Classification
Significant error reduction

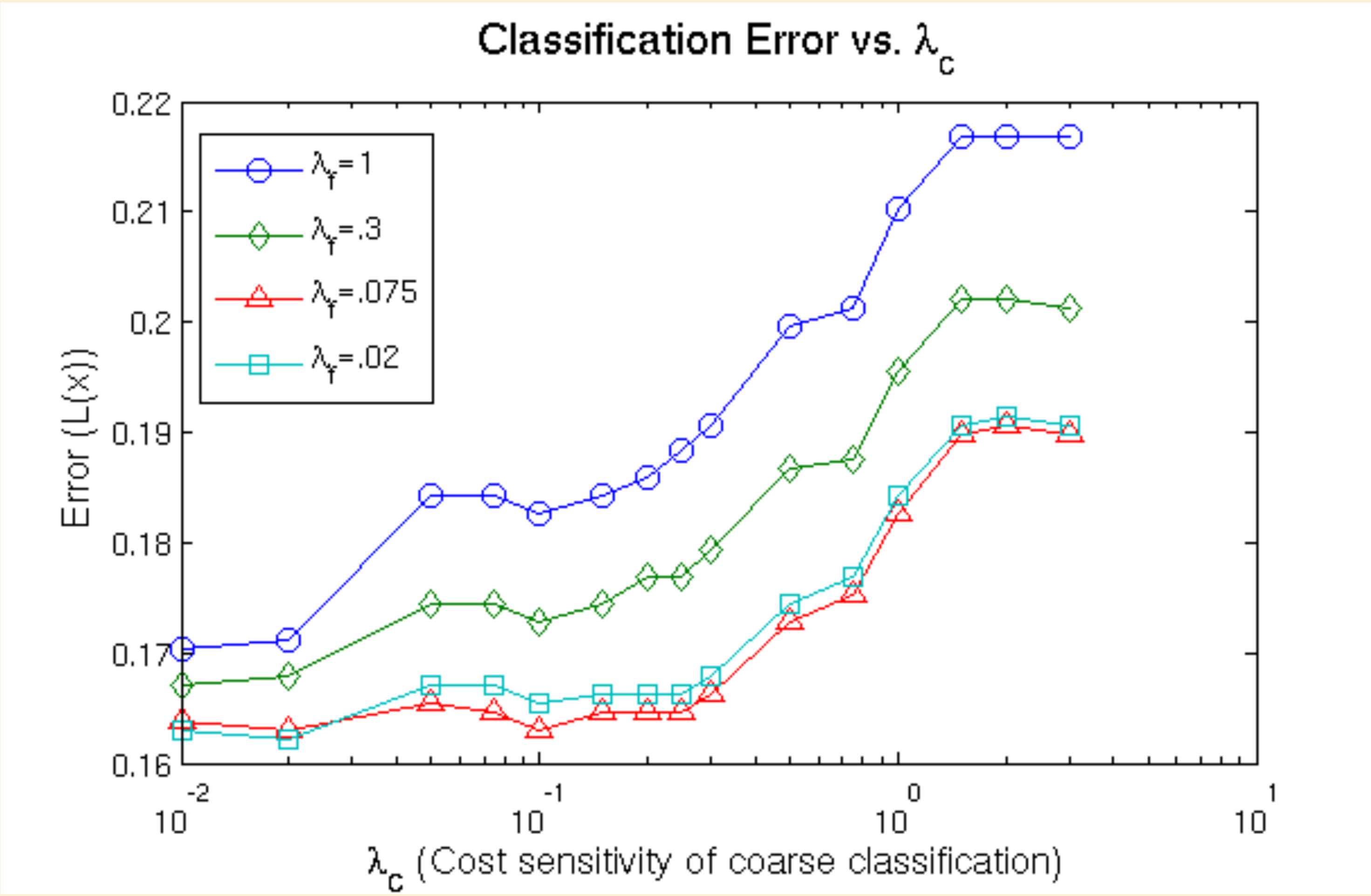


Results: Granular Classification

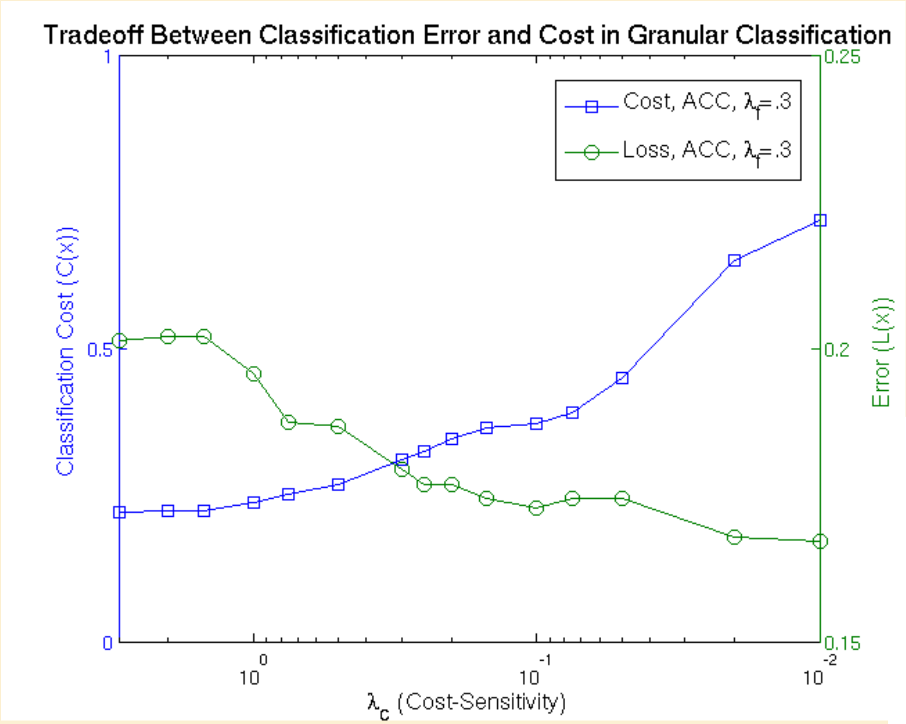
Feature Set	Feature Cost	Misclass Cost		
		Coarse	Fine	Overall
Fixed: IP	**.168**	.139	.181	.229
ACC: λ_c=1.5, λ_f=1	.187	.140	.156	**.217**
Fixed: IP+MailFrom	.490	.128	.142	.200
ACC: λ_c=.1, λ_f=.075	**.431**	.111	.100	**.163**
Fixed: IP+MailFrom+Subject	1.00	.106	.108	.162
ACC: λ_c=.02, λ_f=.02	**.691**	.108	.105	.162

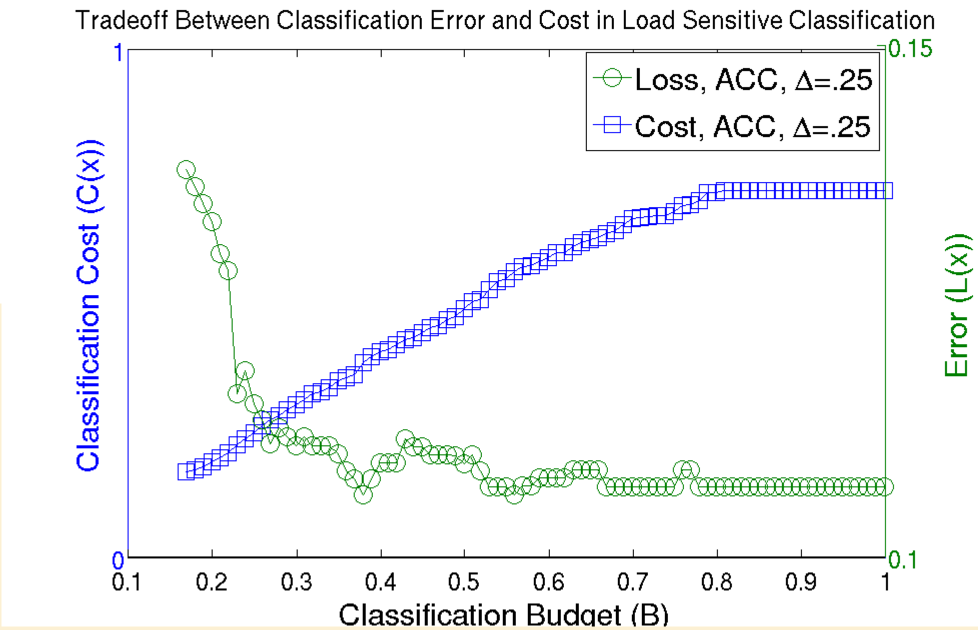
- Compare fixed feature acquisition policies to adaptive classifiers
- Significant gains in performance or cost (or both) depending on tradeoff

Dynamics of choosing λ_c and λ_f



Different approaches, same tradeoff





Conclusion

- Problem of scalable e-mail classification
- Introduce two settings
  - Load-sensitive Classification: known budget
  - Granular Classification: task sensitivity
- Use classifier cascades to achieve tradeoff between cost and accuracy
- Demonstrate results superior to baseline

Questions?

Research funded by Yahoo! Faculty Research Engagement Program