SOTON-WAIS @ CS2013

The shotgun approach to trying to find a technique that improves labels from the crowd

A TALE OF THREE TECHNIQUES

- How can we improve beyond majority voting with the provided workers?
  - Ideas:
    - Employ more workers
    - Play some statistical games
      - Find the unreliable workers and discount them
    - Play some more statistical games
      - Find the unreliable workers and discount them…
      - And at the same time try to *learn* classifiers from the data

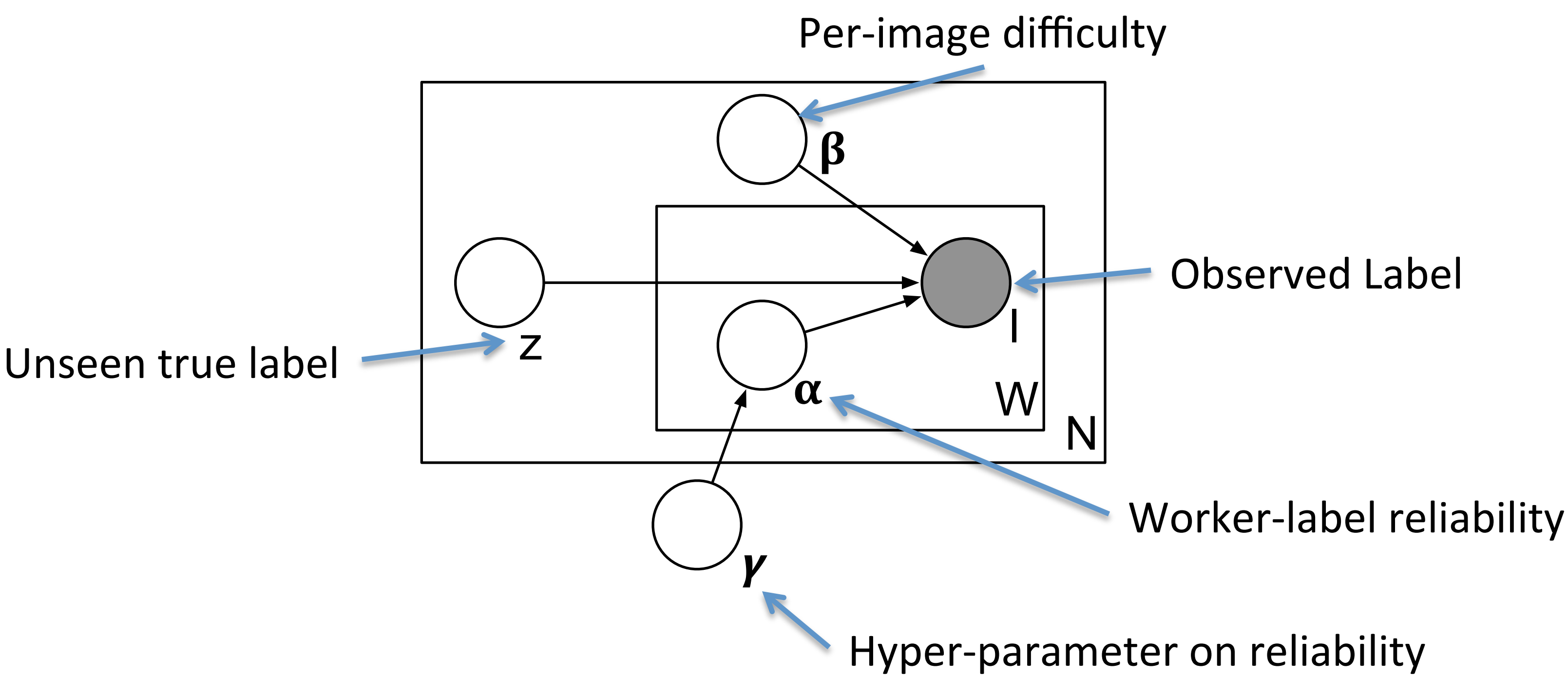
RUN 1: STATISTICAL GAMES

- There is a stack of research on using generative probabilistic models of workers to improve over majority voting.
  - Goes all the way back to a paper in 1977/78!

- Basic Idea:
  - Estimate worker reliability and thus better estimates of the true response
- More complex models incorporate item difficulty, etc.

RUN 1: STATISTICAL GAMES

- We picked an off-the-shelf model by Paul Mineiro @ Microsoft



RUN 2: CROWD & EXPERTS

- Idea: Generate additional labels, and use straight majority voting.
- Employ crowd workers to re-label the images that had more than 2 "NotSure" answers
  - Used the CrowdFlower platform
  - 824 additional responses from 421 images

RUN 2: CROWD & EXPERTS

- Get two fashion "experts" to label 1000 randomly selected images

RUN 2: CROWD & EXPERTS

- Get two fashion "experts" to label 1000 randomly selected images

FASHION EXPERTS

RUN 2: CROWD & EXPERTS

- Get two fashion "experts" to label 1000 randomly selected images

- Labelled images independently & then conferred on the ones which they disagreed

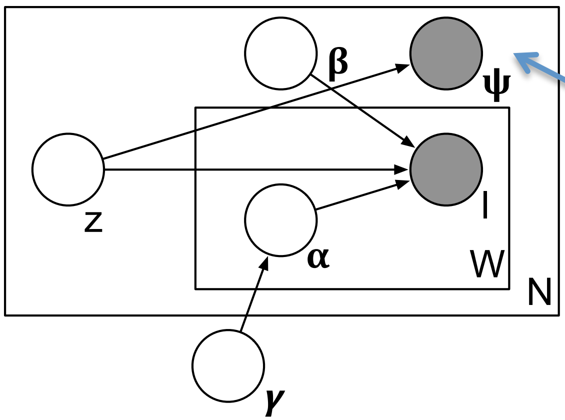
FASHION EXPERTS

RUN 3: CROWD, EXPERTS & STATISTICAL GAMES

- Use the run #1 PGM with the additional data from run #2
  - Use the expert labels to "clamp" the model during training.

RUN 4: CROWD, EXPERTS & MORE STATISTICAL GAMES WITH TEXT FEATURES

- Apply another PGM by Paul Mineiro which extends the previous one with features

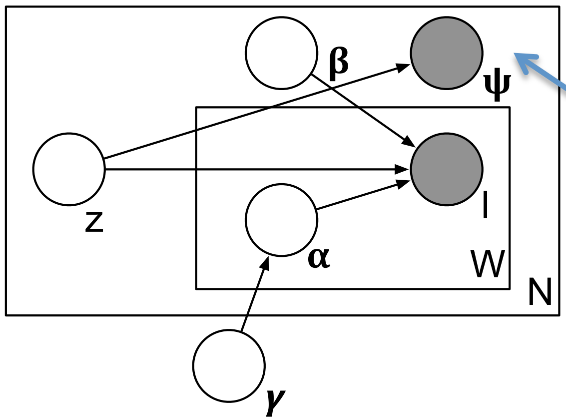


Observed features
(we used BoW from the titles, tags, descriptions, contexts and notes)

- In learning the model parameters, the features are used to learn a classifier, which in turn informs the model parameters for the next iteration

RUN 5: CROWD, EXPERTS & MORE STATISTICAL GAMES WITH TEXT & VISUAL FEATURES

- Same as run #4, but add visual features to the mix
  - 2x2-4x4 PHOW from dense SIFT quantised into 300 visual terms



Observed features
(BoW from the titles, tags, descriptions, contexts and notes + PHOW)

RESULTS AND OBSERVATIONS

RUN #	LABEL 1 F1 SCORE	LABEL 2 F1 SCORE
1	0.7352	0.7636
2	0.8377	0.7621
3	0.7198	0.7710
4	0.7097	0.7528
5	0.6427	0.6026

RESULTS AND OBSERVATIONS

RUN #	LABEL 1 F1 SCORE	LABEL 2 F1 SCORE
1	0.7352	0.7636
2	0.8377	0.7621
3	0.7198	0.7710
4	0.7097	0.7528
5	0.6427	0.6026

Additional data **really** helped with the first label, but not the second

RESULTS AND OBSERVATIONS

RUN #	LABEL 1 F1 SCORE	LABEL 2 F1 SCORE
1	0.7352	0.7636
2	0.8377	0.7621
3	0.7198	0.7710
4	0.7097	0.7528
5	0.6427	0.6026

The worker PGM didn't benefit from the additional data for label 1, but there was a minor improvement for label 2.

RESULTS AND OBSERVATIONS

RUN #	LABEL 1 F1 SCORE	LABEL 2 F1 SCORE
1	0.7352	0.7636
2	0.8377	0.7621
3	0.7198	0.7710
4	0.7097	0.7528
5	0.6427	0.6026

The joint modelling with text features didn't help, but didn't hurt to much (over run #3). Visual features didn't work so well though.

RESULTS AND OBSERVATIONS

RUN #	LABEL 1 F1 SCORE	LABEL 2 F1 SCORE
1	0.7352	0.7636
2	0.8377	0.7621
3	0.7198	0.7710
4	0.7097	0.7528
5	0.6427	0.6026

These are strangely similar… why?

In our PGMs we assumed this was a binary labelling problem, but it's really multi-class…

SOME THOUGHTS FOR DISCUSSION

- Were the questions asked of the workers too subjective?
  - Is asking "is this a fashion image" more subjective than asking if a certain fashion item is present in the image?
    - This might explain why our additional crowdsourcing had such a big effect on the first label, but virtually no effect on the second
  - How much do the example images shown to the workers bias their scoring?

SOME THOUGHTS FOR DISCUSSION

- Why don't the PGMs seem to fit well?
  - We'd at least expect the label 1 score for the third run to be near that of run 2.
  - Usual reasons given:
    - The PGM doesn't model the process well
      - Other published work shows these models to work though… what's special about our task?
    - The data is bad and no amount of statistical tricks can make it better
      - Difficult to prove/disprove, but if it is bad, why is it bad?

ANY QUESTIONS OR COMMENTS?