Factorization and Embeddings

Reminders/Comments

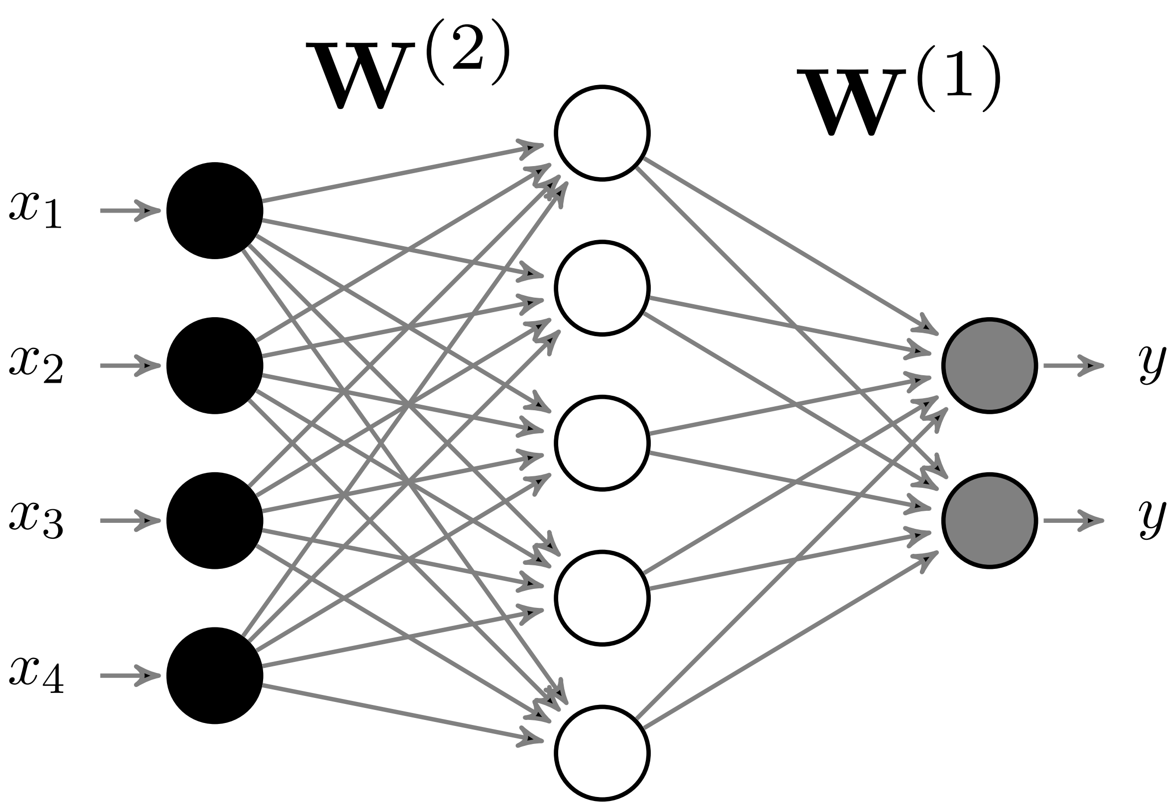
- Last assignment due this Friday week
- Mini-project draft due next Tuesday
- For three-way analysis, use Anova (extension on t-test)
- Do not submit data with mini-project
  - consider it like a paper submission: the document must be self contained, but code can be submitted in case the reviewers want to look at the code for some clarifications
- Each person will be randomly assigned one graduate student, that will review their project
  - for teams of two, this means you get two pieces of feedback

Review

- Done the material I will test you on, i.e., the foundations
  - probabilities
  - maximum likelihood and MAP
  - regression and generalized linear models
  - classification
  - (simple) representation learning approaches
  - empirical evaluation
- Remainder of the course will be reinforcing these topics, and providing more advanced material for interest (which in itself reinforces these foundational topics)

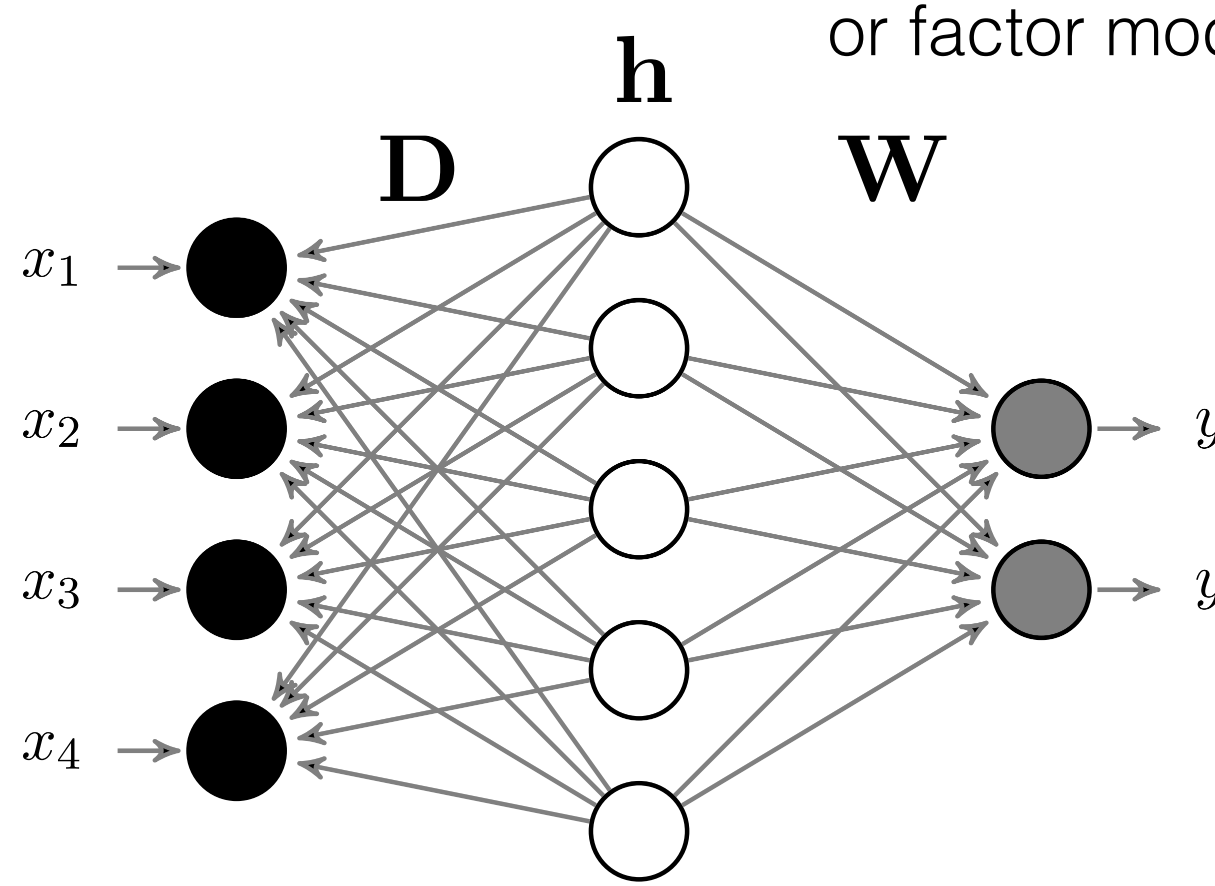
Two representation learning approaches

Neural network



Dictionary Learning models

or factor models



Pros/cons

- Neural networks

  ✓ demonstrably useful in practice

  ✓ theoretical representability results

  - can be difficult to optimize, due to non-convexity

  - properties of solutions not well understood

  - not natural for missing data

- Matrix factorization models

  ✓ widely used for unsupervised learning

  ✓ simple to optimize, with well understood solutions in many situations

  ✓ amenable to missing data

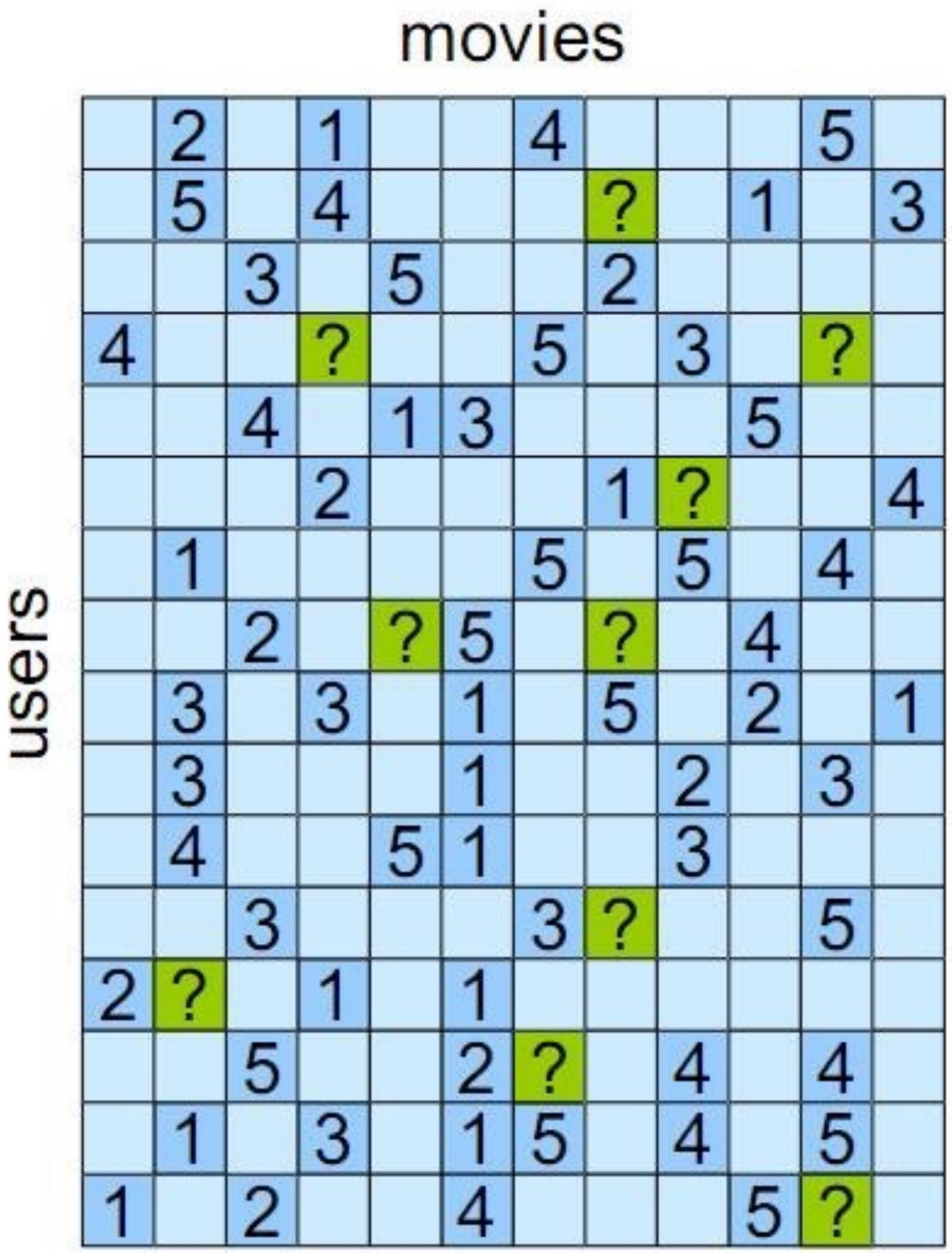
  - less well understood for supervised learning

  - much fewer demonstrations of utility

Whiteboard

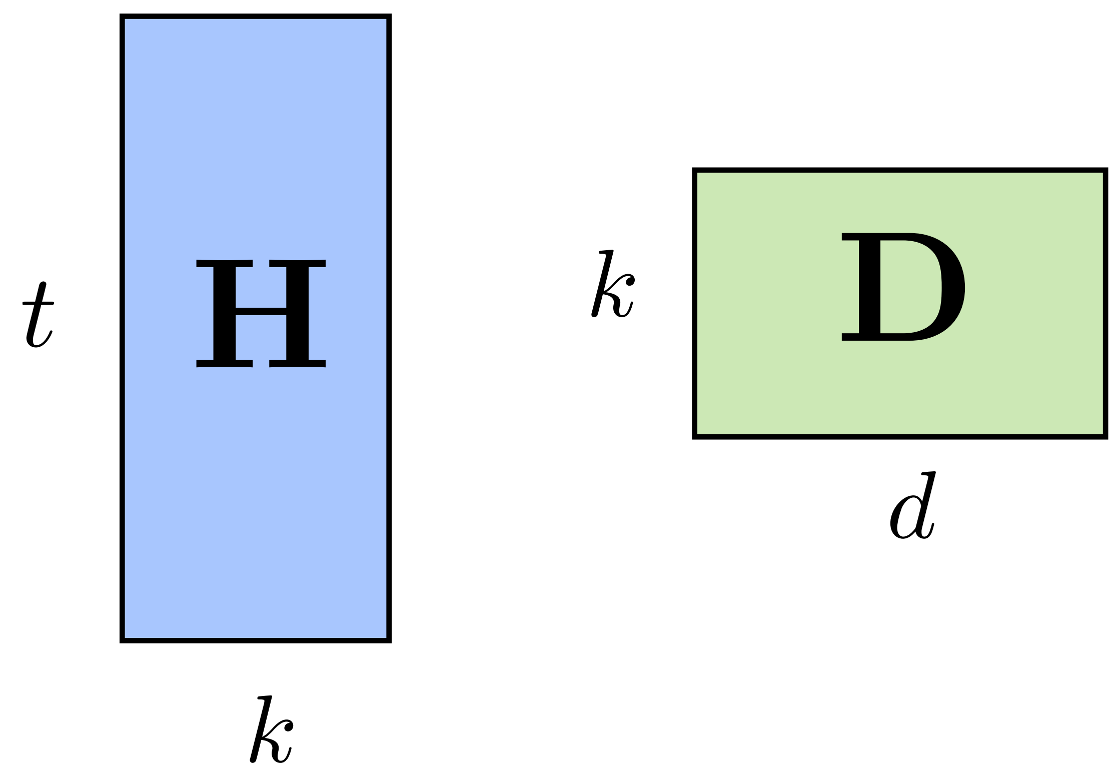
- PCA solution for basic matrix factorization
- Generalizing the loss
- Relationship to auto-encoders
- Missing variables
- Embedding vectors with co-occurrence data

Matrix completion



$\approx$

Subspace (low-rank) form

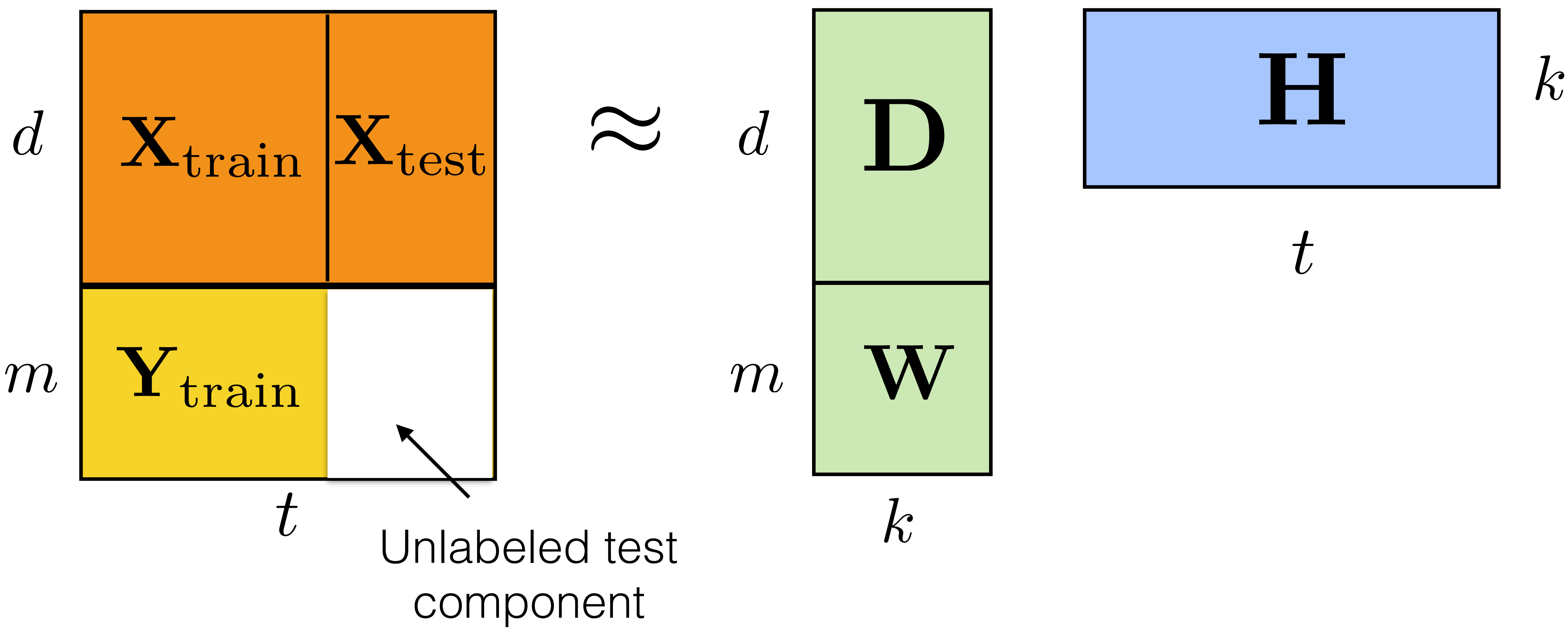


Transductive learning

- Transductive learning uses features for test set to improve prediction performance
- Induction (standard supervised learning): use training data to create general rule (prediction function) to apply to test cases
- Transduction: reason from both training and test cases to predict on test cases (i.e., learn model with both)
    - only training data has labels, but have access to unlabeled test cases
- Motivation: "When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one."
    - “easier” to label test set specifically, rather than generalize further

Transductive learning

Note: everything is transposed, because transductive papers typically write it this way. Exercise, write it in other the order



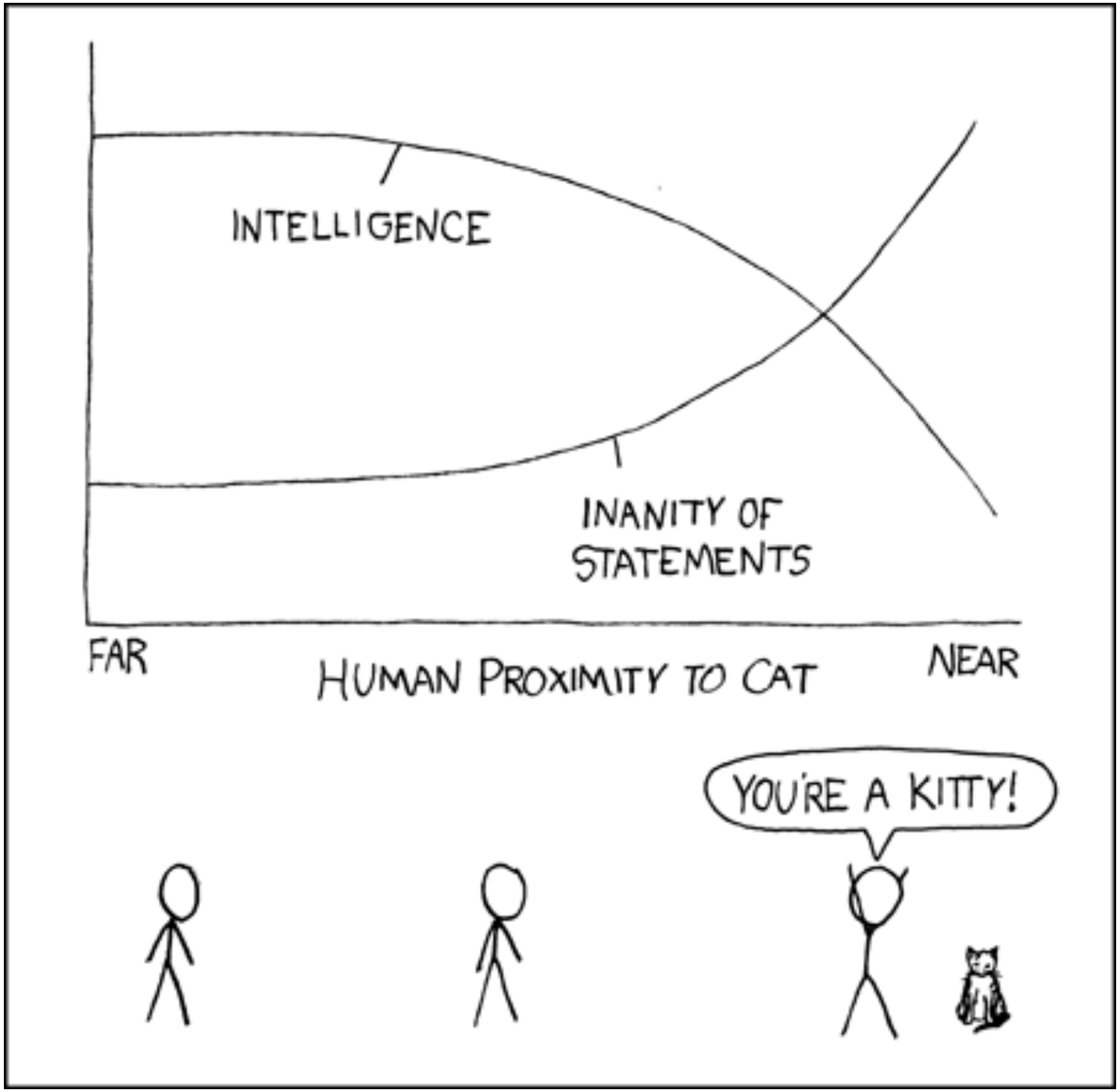
Hidden variables

- Different from missing variables, in the sense that we *could* have observed the missing information
  - e.g., if the person had just filled in the box on the form
- Hidden variables are never observed; rather they are useful for model description
  - e.g., hidden, latent representation
  - e.g., hidden state that drives dynamics
- Hidden variables make specification of distribution simpler
  - p(x) = \sum p(x | h = i) p(h = i) for Gaussians p(x | h = i)
  - p(x | h) is often much simpler to specify

Intuitive example

- Underlying “state” influencing what we observe; partial observability makes what we observe difficult to interpret

- Imagine we can never see that a kitten is present; but it clearly helps to explain the data



Hidden variable models

- Probabilistic PCA and factor analysis
  - common in psychology
- Mixture models
- Hidden Markov Models
  - commonly used for NLP and modeling dynamical systems