Cambridge Interview Technical Talk

David Duvenaud

February 2, 2010

Table of contents

Background

- Causal learning methods are often evaluated in terms of their ability to discover a true underlying directed acyclic graph (DAG) structure.
- However, in general the true structure is unknown and may not be a DAG structure.
- We therefore consider evaluating causal learning methods in terms of predicting the effects of interventions on unseen test data.

Background

- We define causal learning as the task of predicting $P(X|A)$, where A are the actions, and X are the observed variables.
- We show that there exist a variety of approaches to modeling $P(X|A)$, generalizing DAG-based methods.
- Our experiments on synthetic and biological data indicate that some non-DAG models perform as well or better than DAG-based methods.

Example of a Causal System

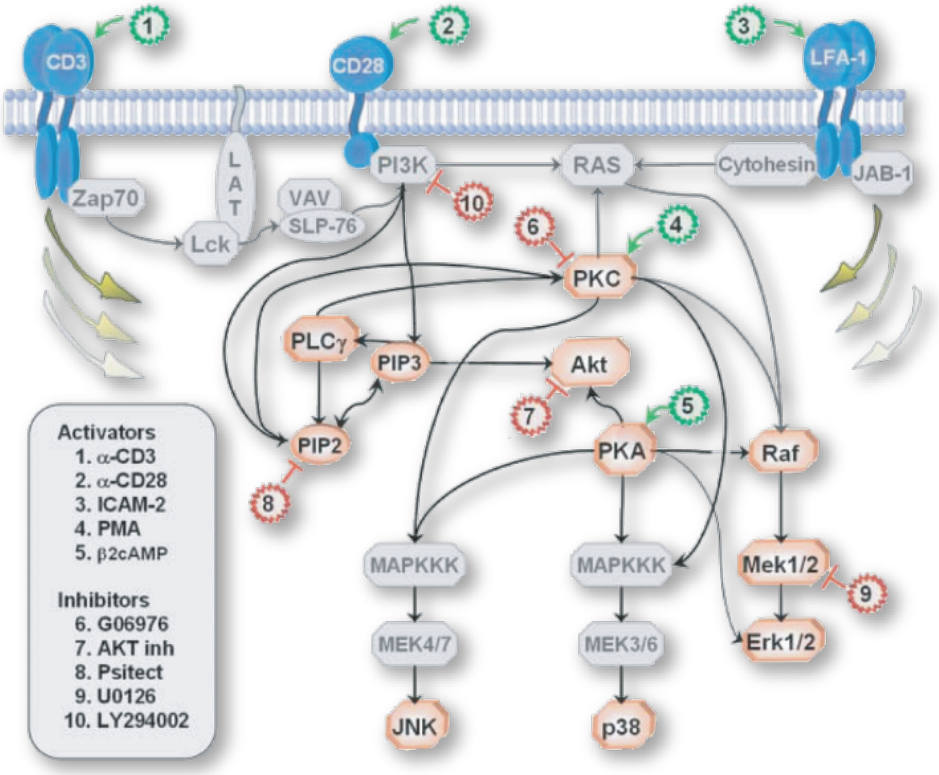


Figure: Biologist's view of the causal links between protein expression levels and gene expression.

Interventional Data

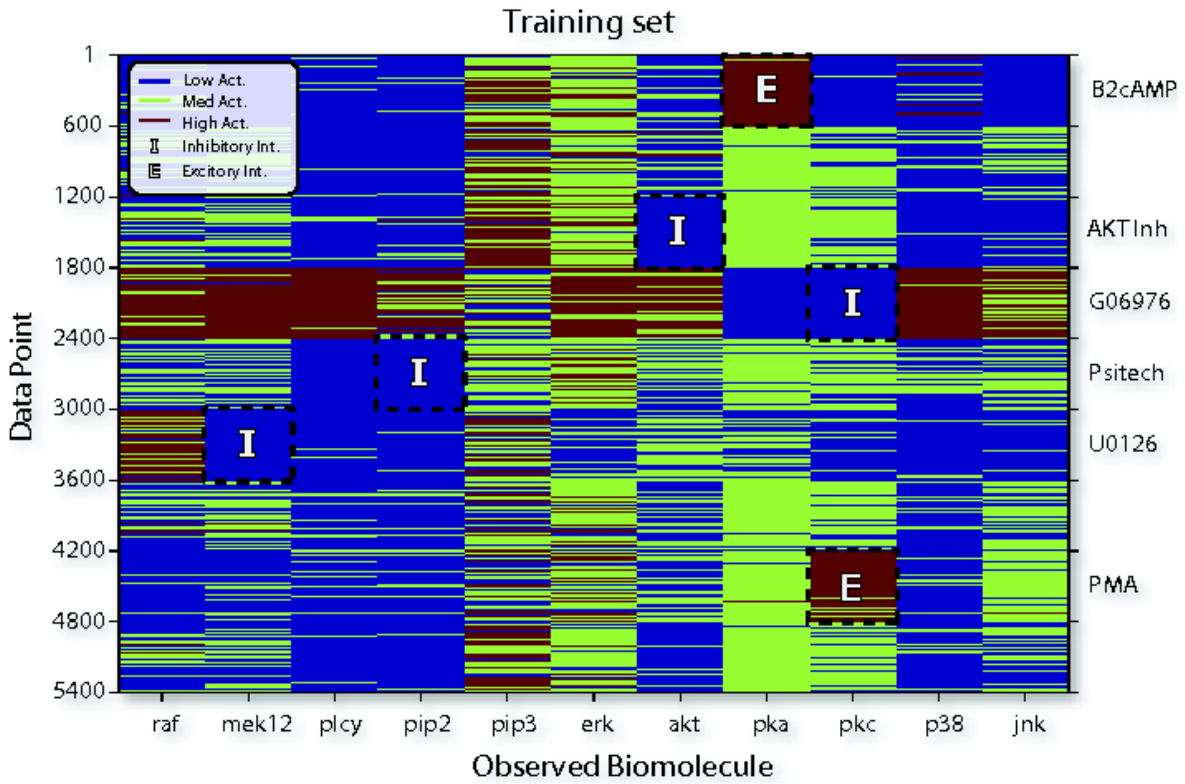


Figure: Interventional Data. Boxes marked 'I' and 'E' denote experiments where an intervention was made on the value of those variables.

Methods of Pooling Data Across Actions

1. **Ignore Actions**: We simply ignore A and build a generative model of $P(X)$.
2. **Independent Model for Each Action**: We fit a separate model $P(X|A)$ for each unique joint configuration of A. Cannot make a prediction for an unseen configuration of A.
3. **Conditional**: We build a model of $P(X|A)$, where we use some parametric model relating the A's and X's. This will allow us to borrow strength across action regimes, and to handle novel actions.
4. **Assume Perfect Interventions** : We assume perfect interventions, and find the MAP DAG representing the system. Inference is done by Pearls graph surgery method.

Off-the-Shelf Models

Generic methods of modeling $P(X|A)$:

- **MM**: Mixture of Multinomial Logistic Regressors.
- **UGM**: A pairwise CRF, conditional on A.
- **DAG**: Bayes Model Averaging over all possible DAGs over variables X and A.

Predictive Accuracy on Completely Unseen Actions

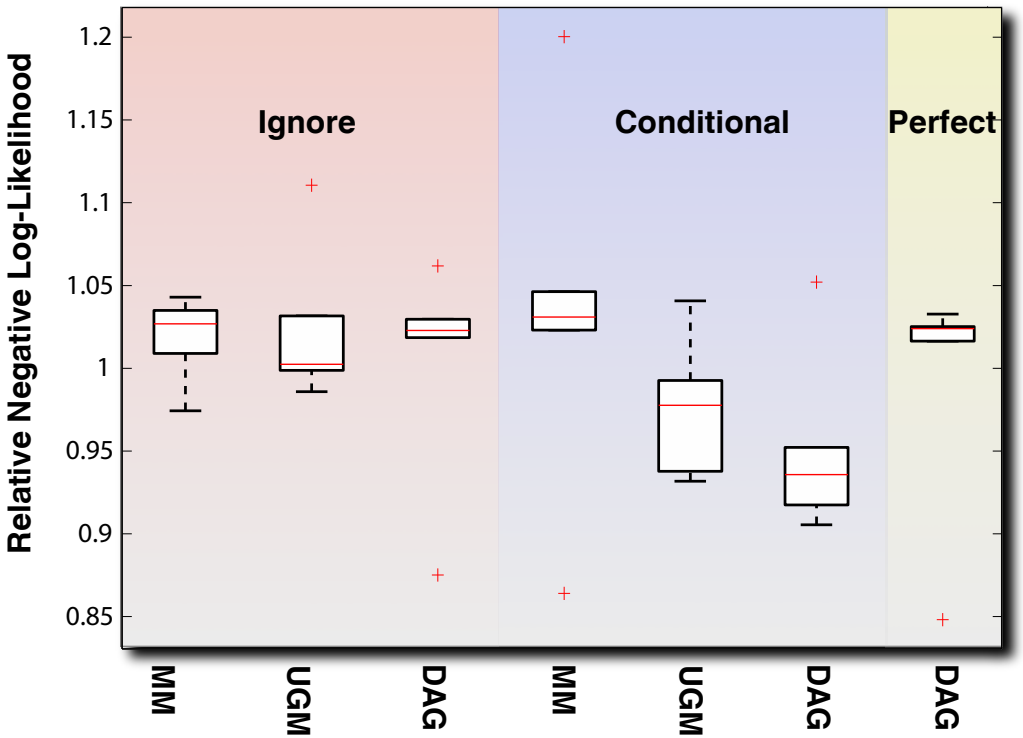


Figure: Modeling $P(X|A)$ under joint configurations of A that were never seen in the training data. Lower is better.

Conclusion

- Causal learning can be viewed as the supervised learning problem of modeling $P(X|A)$.
- In realistic situations, standard density estimators are competitive with Causal DAGs.
- This work is to appear in *JMLR Special Issue on Causality*, alongside Phil Dawid's *Beware of the DAG!*

Table of contents

Space Carving

- We introduce a new technique called *space carving* for initialization and incremental construction of deep architectures.
- Joint work with Benjamin Marlin and Kevin Murphy.

Models

We consider 3 different models:

- Multi-layer Perceptrons
- Auto-encoder Neural Networks
- Deep Belief Networks

All share a sigmoidal activation function for hidden units.

Hidden units as Hyperplanes

- We set the initial weights for a hidden unit by solving a linear classification problem where the data cases in a specified group form one class, and other data cases form the other class.
- The zero-crossing of this sigmoid defines a hyperplane splitting the groups, 'carving' off a region of input space.

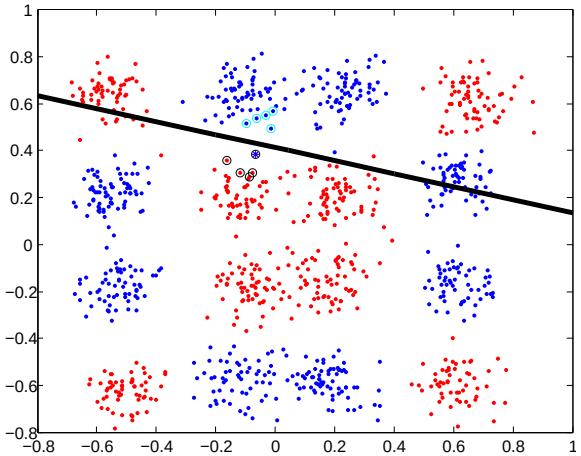
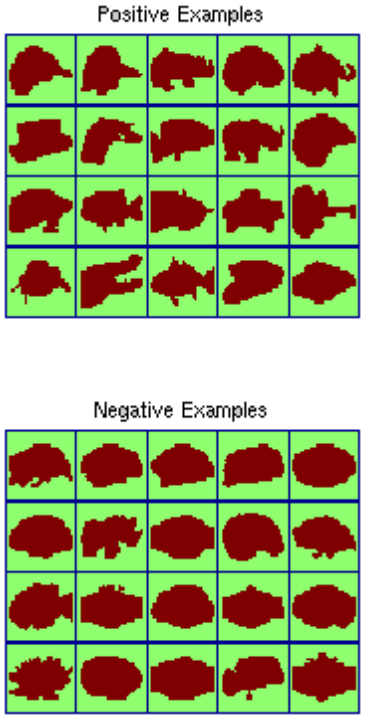


Figure: The black line shows the hyperplane separating the highlighted data points.

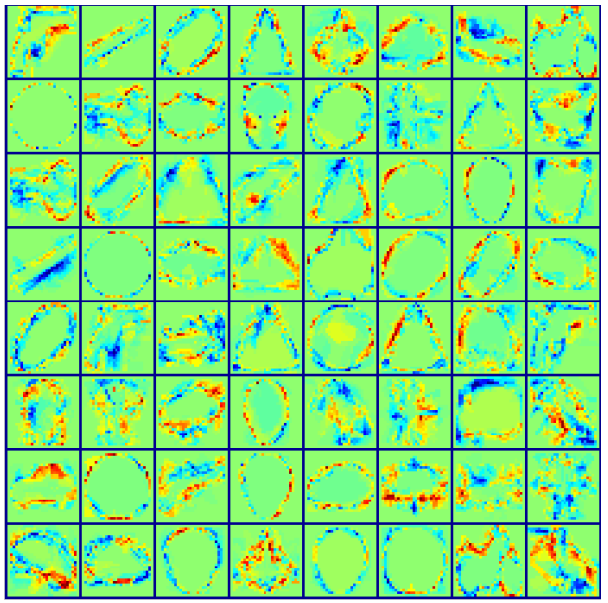
Uses for Space Carving

1. As a method for initializing the weights in single-hidden-layer MLP's, auto-encoders, and RBMs.
2. To initialize the training of each layer in a greedy layer-wise approach to learning deep models. (Pre-pretraining)
3. To incrementally construct deep models by iteratively initializing new hidden units at arbitrary depths to reduce loss.

Caltech Silhouettes Dataset



Data points to be separated by a hyperplane.



Weights representing separating hyperplanes. These weights are used to initialize the first layer of the network.

Results as a Pre-preinitialization step on Silhouettes

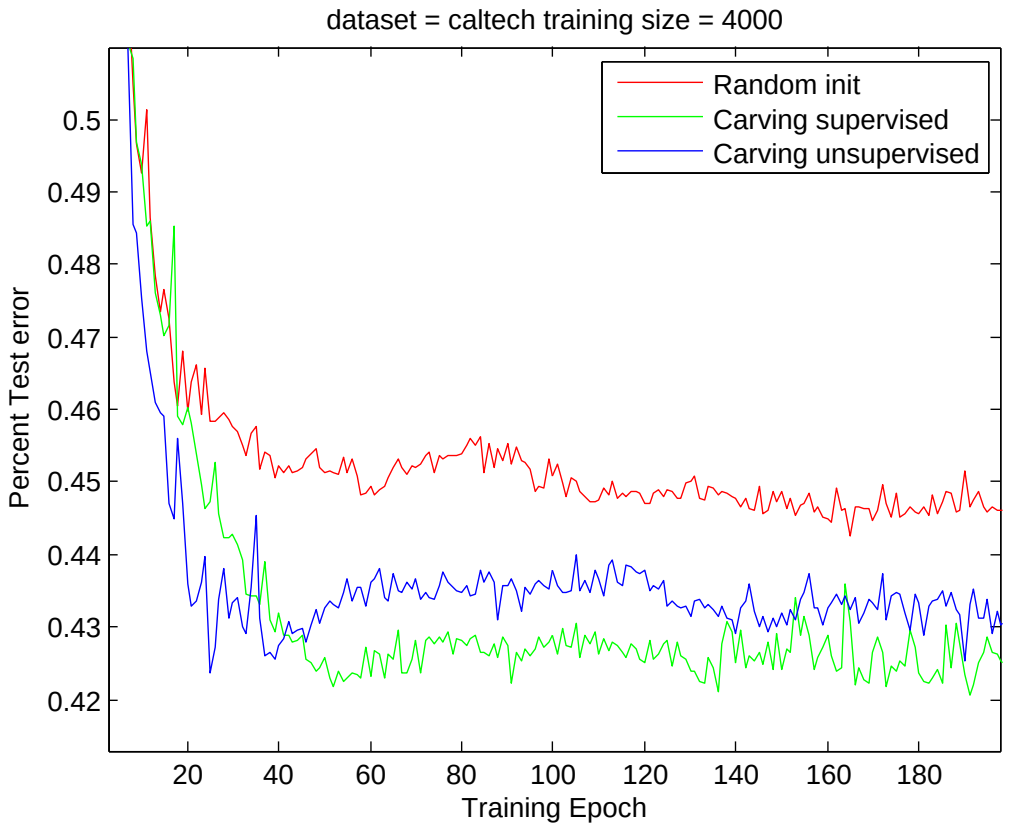


Figure: Performance on test set during training.

Results as a Pre-preinitialization step on MNIST

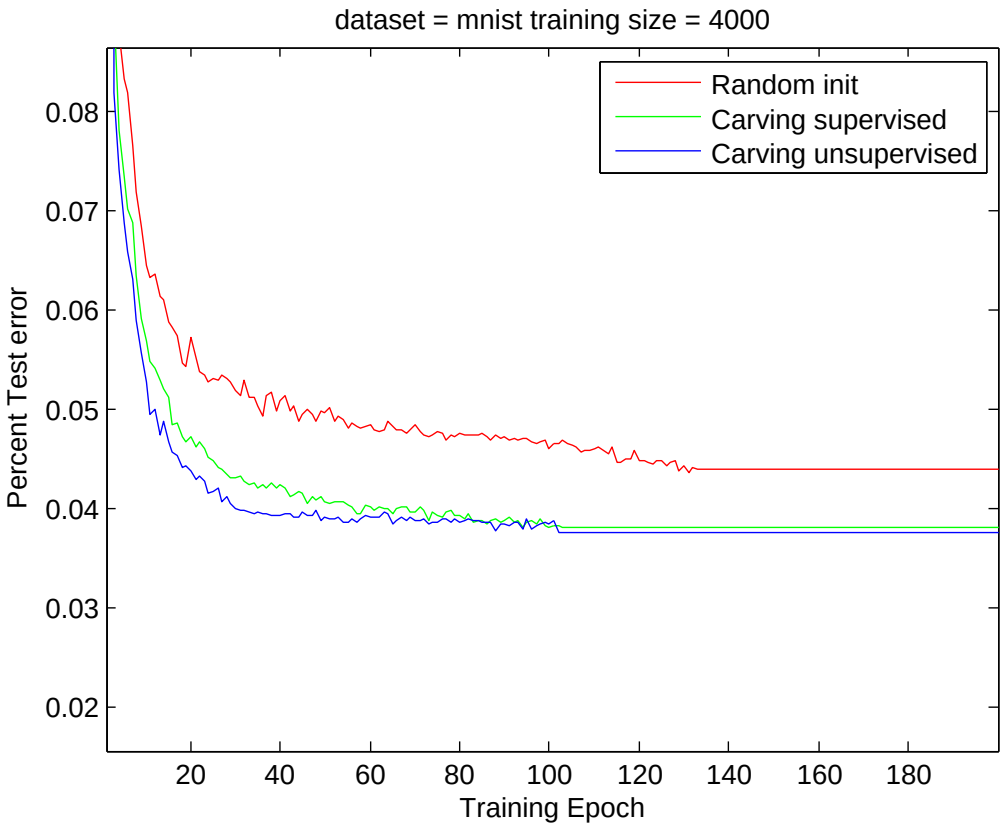


Figure: Performance on test set during training.

Results as a Pre-preinitialization Step

Test Error (percent)

Algorithm	Silhouettes	MNIST(4000)
Random init	44.6	4.41
Unsup. carving	43.0	**3.77**
Supervised carving	**42.9**	3.81

Growing and Shrinking a Net

- Space carving can be used to propose new nodes at any layer in an MLP.
- If we use Group L1 to prune nodes, we can grow/shrink the net as we learn.

Growing and Shrinking a Net

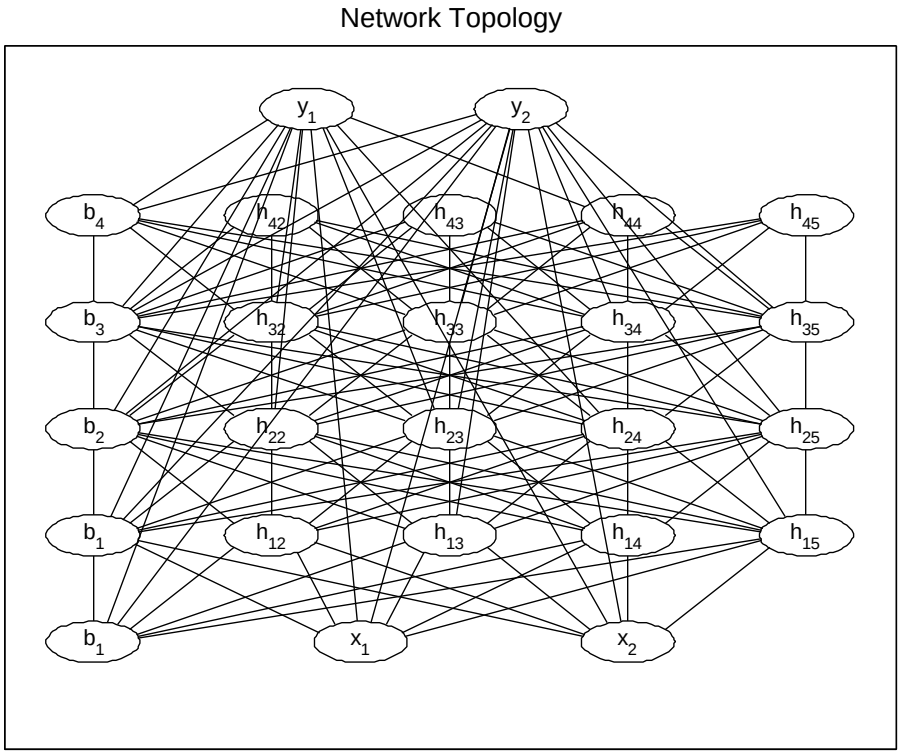


Figure: Initial deep, wide net.

Growing and Shrinking a Net

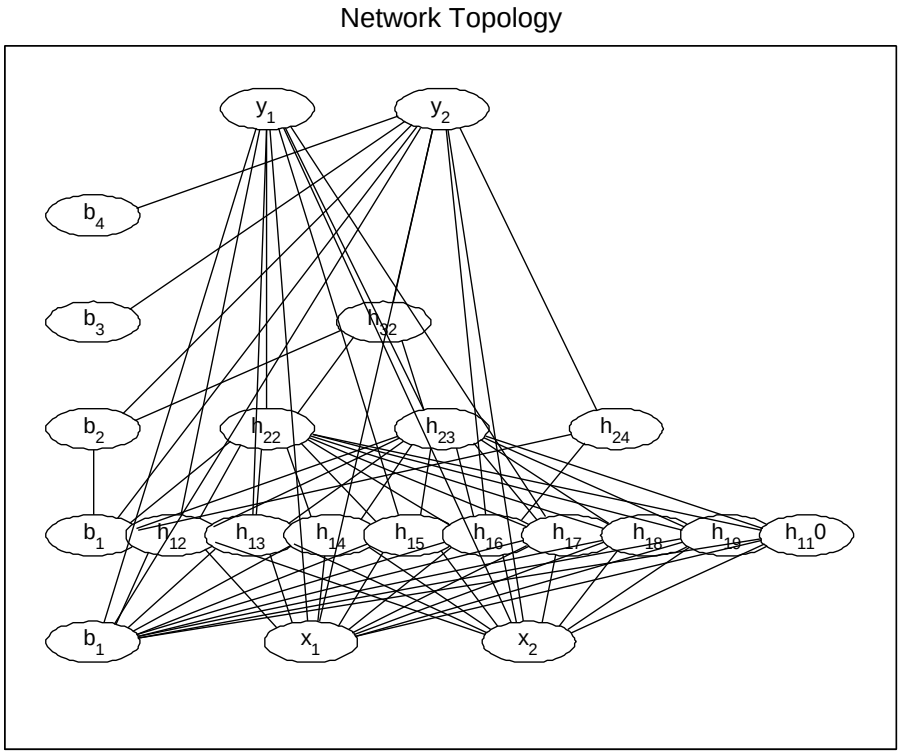


Figure: Net after adding hidden nodes and pruning.

Future Directions

- Effect seems stronger for smaller data sizes.
- Related work: Why does Unsupervised Pre-training Help Deep Learning?, Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, Samy Bengio, *Journal of Machine Learning Research* (2010).
- This is just one potential heuristic for pre-initializing weights into good basins of attraction.

Table of contents

Motivation

- Most image data is weakly labeled, having only bounding boxes or caption data available.
- Feature detectors can be run at multiple scales, and evidence from one scale can help classification at other scales.

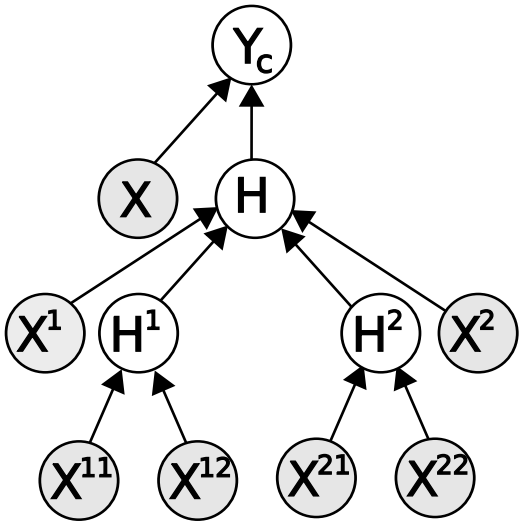


Figure: Standard vision models, using pixel level and image-level features to predict the label.

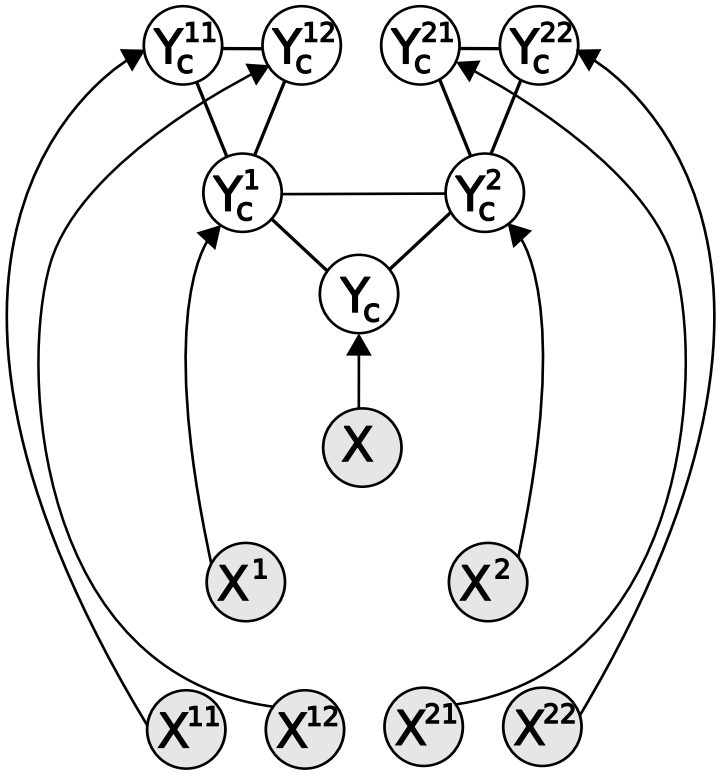


Figure: A hierarchical model with image features providing evidence at every scale.

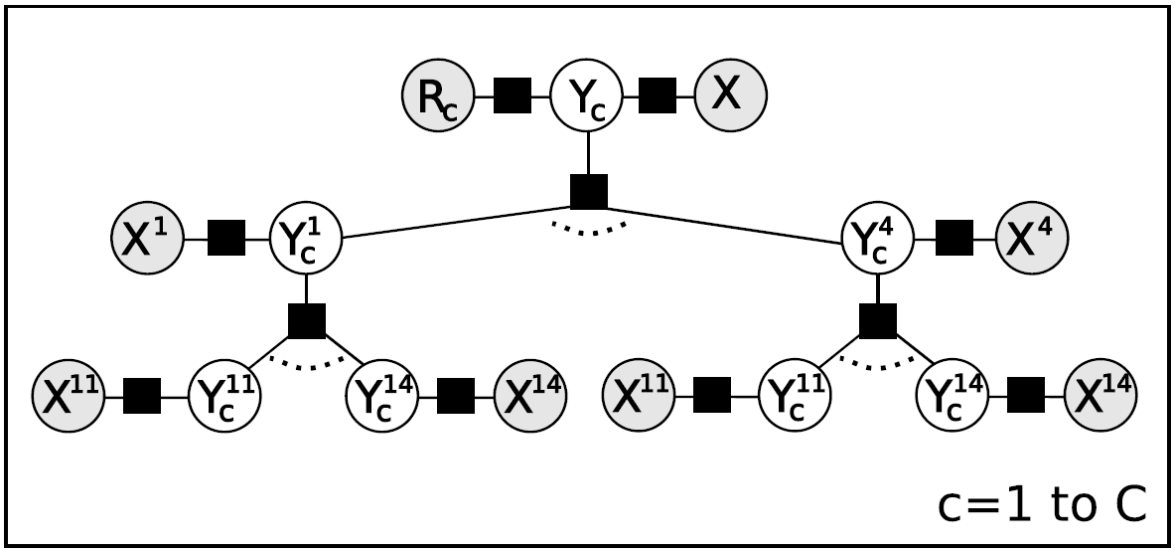


Figure: A hierarchical factor graph. Yc is the label for the whole image. Each factor is a noisy-OR node, in which inference is linear in the number of children. [Pearl 88]

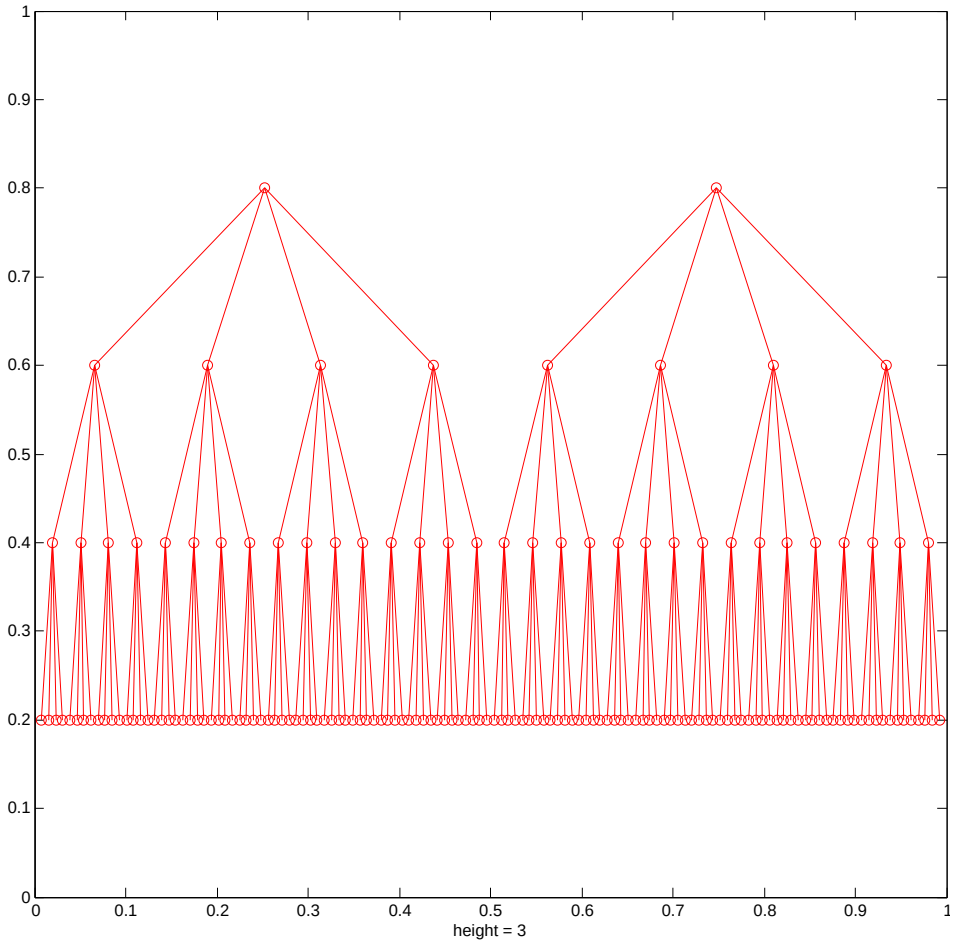


Figure: Recursive noisy-or graph. Each class has its own tree. Trees are connected at the bottom by soft mutual exclusion links.

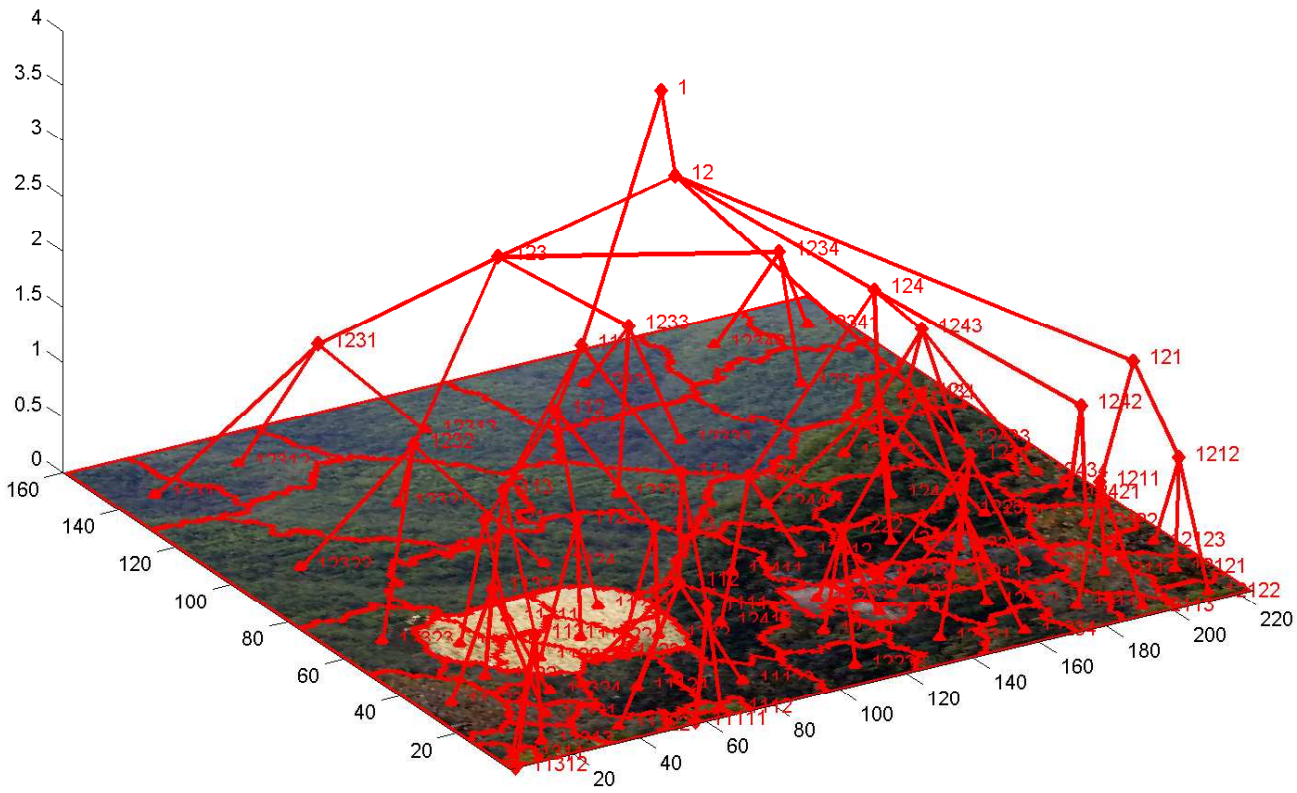


Figure: Recursive segmentation on an image.

Learning

We have a logistic regression at each node:

$$\phi_f\left(y_c^{(r)}, x^{(r)}\right) = y_c^{(r)}(x^{(r)})^T W_c^l \tag{1}$$

Which can be learned using EM and with Loopy BP. Gradients are just the difference in marginals between BP with the observed nodes clamped, and unclamped BP:

$$\frac{\partial E[\mathcal{L}]}{\partial W_c^l} = \sum_{n=1}^{N} \sum_{(i)\in N_l} \left(E_{q(y_{cn}^{(i)}|\mathbf{y}_n^{obs},\mathbf{x}_n,\mathbf{r}_n)}[y_{cn}^{(i)}\mathbf{x}_n^{(i)}] - E_{p(y'^{(i)}_c|\mathbf{x}_n,\mathbf{r}_n)}[y'^{(i)}_c\mathbf{x}_n^{(i)}] \right)$$

$$= \sum_{n=1}^{N} \sum_{(i)\in N_l} \left(q(y_{cn}^{(i)}|\mathbf{y}_n^{obs},\mathbf{x}_n,\mathbf{r}_n) - p(y'^{(i)}_c|\mathbf{x}_n,\mathbf{r}_n) \right) \mathbf{x}_n^{(i)} \tag{2}$$

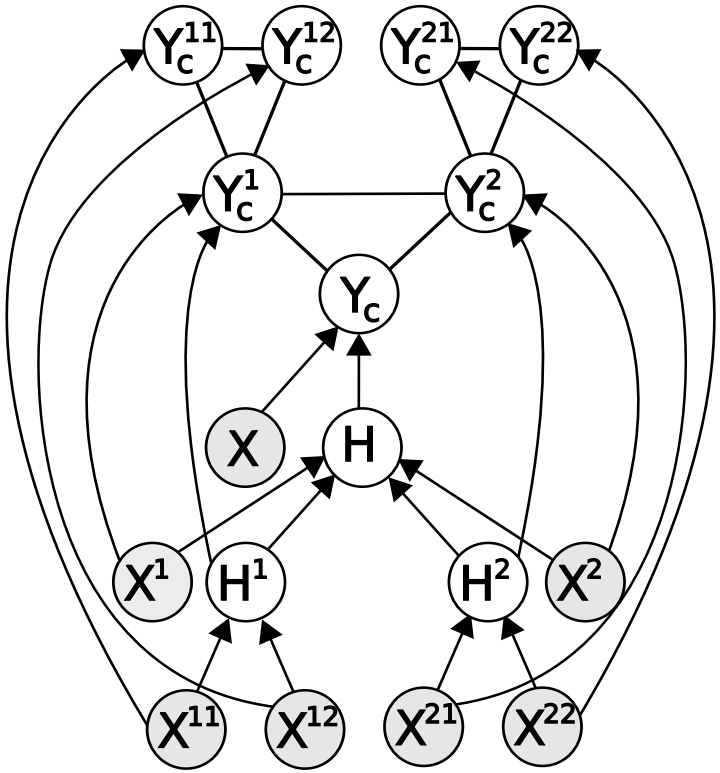


Figure: One possible direction: Including deep features, with hidden nodes passing up information from lower layers.

Conclusion

1. We use a per-image hierarchical recursive segmentation structure.
2. The logical structure of this segmentation allows us to combine evidence at multiple scales, to use weak labels, such as bounding boxes or captions.
3. Inference is done using Loopy BP, and learning is done using EM.
4. Joint work with Ben Marlin, Kevin Murphy, and probably Tom Dean at Google this summer.