Probabilistic Label Relation Graphs with Ising Models

Nan Ding
Google Inc.
340 Main Street
Venice, CA 90291
dingnan@google.com

Jia Deng
University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109
jiadeng@umich.edu

Kevin P. Murphy
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
kpmurphy@google.com

Hartmut Neven
Google Inc.
340 Main Street
Venice, CA 90291
neven@google.com

ABSTRACT

We consider classification problems in which the label space has structure. A common example is hierarchical label spaces, corresponding to the case where one label subsumes another (e.g., animal subsumes dog). But labels can also be mutually exclusive (e.g., dog vs cat) or unrelated (e.g., furry, carnivore). In our prior work, we introduced the notion of a HEX graph, which is a way of encoding hierarchy and exclusion relations between labels into a conditional random field (CRF). We combined the CRF with a deep neural network (DNN), resulting in state of the art results when applied to visual object classification problems where the training labels were drawn from different levels of the ImageNet hierarchy (e.g., an image might be labeled with the basic level category "dog", rather than the more specific label "husky"). In this paper, we extend the HEX model to allow for soft or probabilistic relations between labels, which is useful when there is uncertainty about the relationship between two labels (e.g., a penguin is "sort of" a subclass of birds, but not to the same degree as a robin or sparrow). We call our new model pHEX, for probabilistic HEX. We show that the pHEX graph can be converted to an Ising model, which allows us to use existing off-the-shelf inference methods (in contrast to the HEX method, which needed specialized inference algorithms). Experimental results show significant improvements in a number of large-scale object classification tasks.

1. INTRODUCTION

Classification is a fundamental problem in machine learning. In this paper, we consider how to extend the standard approach to exploit structure in the label space. For example, consider the problem of classifying images of animals. The labels may be names of animal types, like dog, puppy, and cat, as well as attribute labels like yellow, furry, hasstripes, etc. Many of these labels are not semantically independent of each other. For example, a puppy is also a dog, which is a hierarchical or subsumption relation; an animal cannot be both a dog and a cat, an exclusive relation; but an animal can be yellow and furry, which is a non-relation.

In [8], we proposed an approach called Hierarchy and EXclusion (HEX) graphs for compactly representing such constraints between the labels; in particular, we used a probabilistic graphical model with deterministic or hard constraints between the binary label nodes. We showed how these hard constraints cut down the feasible set of labels from 2^n (where n is the number of labels) to something much smaller, allowing for efficient exact inference. For example, if all labels are mutually exclusive, the HEX graph is a clique, and there are only $n+1$ valid label configurations. This graphical model can be combined with any standard discriminative classifier (such as deep neural networks), resulting in a conditional random field (CRF) model with label constraints. We showed that modeling the relationships between the labels can result in much improved performance.

In this paper, we extend our previous work by allowing for "soft" relationships between the labels. We call this the pHEX model. This is necessary because many relationships between labels are not absolute. For example, a lion may be mostly yellow, but it could also be another color. Also, a dog looks more like a cat than a car, so the label *dog* is more "exclusive" of *car* than of *cat*.

Unfortunately, these soft constraints do not allow us to reduce the feasible set of labels, so exact inference is no longer possible in general. However, we show how we can represent pHEX models as Ising models, for which many standard inference tools are available. We show experimentally that approximate inference (using loopy belief propagation) in a pHEX model outperforms exact inference in a HEX model on a variety of classification tasks. Furthermore, not only are pHEX models more accurate, but they are easier to use, since they do not require custom inference algorithms.

In summary, the main contributions of this paper are as follows: we propose a new model called pHEX for representing soft or probabilistic relationships between labels, we propose a way to perform efficient inference in such models by converting them to Ising models, and we show experimentally that the method outperforms our previous approach based on HEX models (which in turn was shown to

outperform standard multiclass and multilabel classification methods).

2. RELATED WORK

There has been a lot of prior work on exploiting structure in the label space; we only have space to mention a few key papers here. Conditional random fields [10, 22] and structural SVMs [21] are often used in structured prediction problems. In addition, in transfer learning [18, 17], zero-shot learning [11, 15], and attribute-based recognition [1, 25, 19], consistency between visual predictions and semantic relations are often enforced.

More closely related to this paper is work that exploits hierarchical structure (e.g., [26, 12, 24, 14]), exclusive relations [4], or both of them [5]. Recently [8] proposed the HEX graph approach, which subsumes a lot of prior work by modeling hierarchical and exclusive relations using graphical models. We discuss this in more detail in Section 3, since it forms the foundation for the current paper.

3. THE HEX MODEL

In a nutshell, HEX graphs are probabilistic graphical models with directed and undirected edges over a number of binary variables. Each binary variable represents a label and takes value from $\{-1, 1\}$. Each edge or no-edge between any two labels represents one of three label relations: exclusion, hierarchy and non-relation. The combination of all pairwise label relations allows the HEX graph to characterize the legal and illegal state space of labels, as we explain below.

3.1 HEX Relations

The three types of label relations in the HEX graph are defined as follows:

Exclusion.

When two nodes are connected by an *undirected edge* (Figure 1), this is called an exclusive relation. It means that the two labels cannot be both equal to 1. For example, an animal cannot be both a *cat* and a *dog*. So *cat* and *dog* are mutually exclusive. The legal state space for exclusion is:

$$S^e \triangleq \{(-1,-1),(-1,1),(1,-1)\}. \tag{1}$$

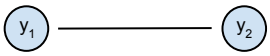

Figure 1: Exclusive relation in HEX graph: both y_1 and y_2 equal to 1 is prohibited.

Hierarchy.

When two nodes are connected by a *directed edge* from y_1 to y_2 (Figure 2), this is called a subsumption (hierarchical) relation. It means that if y_2 is 1 then y_1 must be 1 as well. For example, a *puppy* is always a *dog*. So *dog* subsumes *puppy*. The corresponding legal state space for subsumption is:

$$S^s \triangleq \{(-1,-1),(1,-1),(1,1)\}. \tag{2}$$

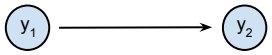

Figure 2: Subsumption (hierarchical) relation in HEX graph: if $y_2 = 1$, then y_1 has to be 1.

No relation.

When two nodes are *not connected* by any edge (Figure 3), we say there is no relation between them. This means that the two labels are independent to each other. For example, *carnivore* and *yellow* are independent properties of animals. In this case, the legal state space of the two variables contains all 4 possible configurations:

$$S^o \triangleq \{(-1,-1),(-1,1),(1,-1),(1,1)\}. \tag{3}$$

Figure 3: Non-relation in HEX graph: y_1, y_2 are mutually independent.

3.2 HEX graph as a graphical model

To mathematically formulate the HEX model, assume we have a set of n possible labels, represented as the bit vector $\mathbf{y} = \{y_1, \ldots, y_n\}$, where $y_i \in \{-1, +1\}$. Also, assume we have an input feature vector $\mathbf{x} = \{x_1, \ldots, x_d\}$, and some discriminative model which maps this to the score vector $\mathbf{z} = \{z_1, \ldots, z_n\}$, where z_i is the "local evidence" for label y_i. (The mapping from $\mathbf{x}$ to $\mathbf{z}$ is arbitrary; in this paper, we assume it is represented by a deep neural network parameterized by $\mathbf{w}$, which we will denote by $\mathbf{z} = DNN(\mathbf{x}; \mathbf{w})$; see Section 6 for details on how to train these parameters.) Give this, we can define the model as follows:

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{i=1}^{n} \psi(y_i, z_i) \prod_{(i,j)\in G} \phi(y_i, y_j), \tag{4}$$

where $\psi(y_i, z_i) = 1/(1+\exp(-2y_i z_i))$ is the logistic function, and $\phi(y_i, y_j)$ is the (edge-specific) potential function, defined below.

We define three kinds of edge potentials, one for each kind of relation. (We use the notation ϕ_a to represent an "absolute" or deterministic potential, to distinguish it from the soft or probabilistic potentials we use later, denoted by ϕ_p.)

- Exclusion

$$\phi_a^e(y_1, y_2) = \begin{cases} 1 & (y_1, y_2) \in S^e \\ 0 & (y_1, y_2) = (1,1); \end{cases} \tag{5}$$

- Hierarchy

$$\phi_a^s(y_1, y_2) = \begin{cases} 1 & (y_1, y_2) \in S^s \\ 0 & (y_1, y_2) = (-1,1); \end{cases} \tag{6}$$

- No relation

$$\phi_a^o(y_1, y_2) = 1 \ \ \forall (y_1, y_2). \tag{7}$$

It is worth noting that the HEX graph naturally generalizes two classical classification models: binary multi-label logistic regression corresponds to a HEX graph with no edges; and the multiclass softmax model is a clique graph where all nodes are connected to each other by undirected edges.

3.3 Inference in HEX models

HEX models have a few important properties which are crucial to enable efficient exact inference. Let us define the product of all the potential functions:

$$\Phi_a^G(\mathbf{y}) = \prod_{(i,j)\in G} \phi_a(y_i, y_j) \in \{0, 1\}. \tag{8}$$

We say that two HEX graphs G and G' are equivalent if and only if they have the same legal state space. In other words,

$$\Phi_a^G(\mathbf{y}) = \Phi_a^{G'}(\mathbf{y}) \ \ \forall \mathbf{y}\,.$$

This allows us to sparsify and densify the HEX graph to its equivalence graph without changing the distribution $p(\mathbf{y} \mid \mathbf{x})$.

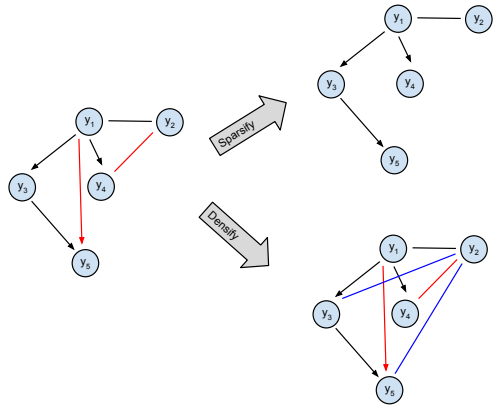

Figure 4: An example of equivalent HEX graphs.

Figure 4 gives an example of three equivalent HEX graphs. In this example, exclusion of label y_1 and y_2 implies the exclusion of y_4 and y_2 because y_1 subsumes y_2, hence we can drop the explicit undirected red $y_2 - y_4$ edge. Similarly, subsumption of y_1 to y_3 and y_3 to y_5 implies the subsumption of y_1 to y_5. Therefore, the left HEX graph can be sparsified to the top right graph without changing $\Phi(\mathbf{y})$. Similarly, we can also densify the left graph to obtain the bottom right graph by dropping the implicit directed blue edges.

Sparsification results in a sparse graph, which often reduces the treewidth of the resulting junction tree, resulting in faster inference. On the other hand, densification adds more constraints, thus reducing the size of the state space for each clique in the junction tree, again speeding up infernece. For example, if n labels are connected to each other with an exclusive clique, then the the legal state space of the n labels is only of size $n+1$ compared to the entire state space of size $O(2^n)$. For more details, see [8].

4. PROBABILISTIC HEX MODELS

In this section, we introduce an extension of the HEX model to allow for soft or probabilistic relationships between labels. The basic idea is to relax the hard constraints, by replacing the value 0 (corresponding to illegal combinations) in the definitions of the potential functions with a value $0 \leq q \leq 1$, representing how strongly we wish to enforce the constraints. (This is somewhat analogous to the approach used in Markov logic networks [16], which relax the hard constraints used in first order logic.) The magnitude of this parameter can in principle be learned from data, although in this paper, we just perform a 1d grid search over a range of values.

The main disadvantage of this relaxation is that we lose the ability to perform tractable exact inference. However, we show that we can formulate pHEX models as Ising models, which opens up the door to using standard tractable approximate inference methods.

4.1 Probabilistic HEX Relations

For clarity, we now explicitly specify the form of the two new factors we introduce. We use the generic parameter q to represent the strength of this relation, although this could easily be made edge/ label dependent.

Probabilistic exclusion.

The potential function of the two variables y_1, y_2 under probabilistic exclusion is defined as:

$$\phi_p^e(y_1, y_2; q) = \begin{cases} 1 & (y_1, y_2) \in S^e \\ q & (y_1, y_2) = (1, 1), \end{cases} \tag{9}$$

where $0 \leq q \leq 1$. When $q = 1$, Equation (9) reduces to the non-relation in Equation (7), where y_i and y_j are independent. When $q = 0$, Equation (9) reduces to the hard exclusion relation Equation (5), where $(y_1, y_2) = (1, 1)$ is strictly prohibited. When $0 < q < 1$, all configurations are legal, but $(y_1, y_2) = (1, 1)$ is with less probability.

Probabilistic hierarchy.

For subsumption, we define

$$\phi_p^s(y_1, y_2; q) = \begin{cases} 1 & (y_1, y_2) \in S^s \\ q & (y_1, y_2) = (-1, 1), \end{cases} \tag{10}$$

where $0 \leq q \leq 1$. When $q = 1$, Equation (10) reduces to the unconstraned relation Equation (7), where y_i and y_j are independent. When $q = 0$, Equation (10) reduces to the hard subsumption relation Equation (6), where $(y_1, y_2) = (-1, 1)$ is strictly prohibited. When $0 < q < 1$, all configurations are legal, but $(y_1, y_2) = (-1, 1)$ is with less probability.

Probabilistic exclusions and subsumptions can be seen as a probabilistic mixture of absolute exclusions, subsumptions, and non-relations, where

$$\phi_p^e(y_1, y_2; q) = q\phi_a^o(y_1, y_2) + (1 - q)\phi_a^e(y_1, y_2),$$
$$\phi_p^s(y_1, y_2; q) = q\phi_a^o(y_1, y_2) + (1 - q)\phi_a^s(y_1, y_2).$$

Therefore, the combination of probabilistic label relations generalizes the absolute label relations in the HEX graph.

4.2 Converting pHEX models to Ising models

We now show how to convert a pHEX model to an Ising model.

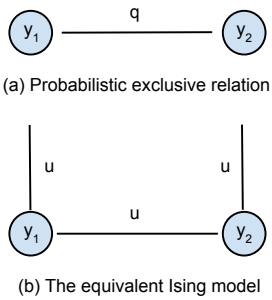

Figure 5: (a) Probabilistic exclusive relations in a pHEX graph with $\phi(1,1) = q$; (b) the coefficients on the nodes and the edge of the equivalent Ising model, where $q = \exp(-4u)$.

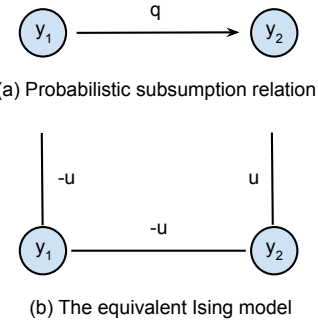

Figure 6: (a) Probabilistic subsumption relations in a pHEX graph with $\phi(-1,1) = q$; (b) the coefficients on the nodes and the edge of the equivalent Ising model, where $q = \exp(-4u)$.

The Ising model was first proposed in statistical mechanics to study ferromagnetism [2]. Mathematically, it is essentially an undirected graphical model which defines the joint distribution of configurations of n binary random variables $\mathbf{y}$ in graph G by a Boltzmann distribution,

$$p_\beta(\mathbf{y}) = \frac{1}{Z_\beta}\exp(-\beta E(\mathbf{y})), \tag{11}$$

where β is a temperature variable that will be omitted later by fixing it to 1. Z_β is the normalization constant. $E(\mathbf{y})$ is the energy function of the configuration $\mathbf{y}$, which takes into account local energy potentials $h_i y_i$ as well as pairwise energy potential $J_{ij} y_i y_j$,

$$E(\mathbf{y}) = \sum_{(i,j)\in G} J_{ij} y_i y_j + \sum_{i=1}^{n} h_i y_i. \tag{12}$$

To convert a pHEX graph to an Ising model, we first show how to convert the factor functions $\phi(y_1, y_2)$ for the pairwise probabilistic relations to the equivalent pairwise energy functions $E(y_1, y_2)$ of an Ising model.

Consider an Ising model of two variables in Figure 5(b), where $u \geq 0$ are the weights on the local potentials and the pairwise potential. The resulting pairwise energy function of this Ising model is,

$$\begin{aligned} E_p^e(y_1, y_2; u) &= uy_1y_2 + uy_1 + uy_2 \\ &= \begin{cases} -u & (y_1, y_2) \in S^e \\ 3u & (y_1, y_2) = (1, 1). \end{cases} \end{aligned} \tag{13}$$

Clearly, Equation (13) looks very similar to Equation (9) of the probabilistic exclusion. In fact, by letting $q = \exp(-4u)$ and $\phi_p^e(y_1, y_2; q) \propto \exp(-E_p^e(y_1, y_2; u))$, we can show they are equivalent up to a constant factor. In particular, let (y_1, y_2) be a legal label pair, and (y_1', y_2') be an illegal pair. We have

$$\phi_p(y_1, y_2)/\phi_p(y_1', y_2') = 1/q = e^{4u}, \tag{14}$$

$$\exp(-E(y_1, y_2) + E(y_1', y_2')) = e^{u+3u} = e^{4u}. \tag{15}$$

A larger u means a stronger exclusion between the two labels. When $u \to +\infty$, Equation (13) reduces to the hard exclusive relation; conversely when $u = 0$, Equation (13) reduces to the absolute non-relation.

Similarly, the equivalent Ising model of the probabilistic subsumption is shown in Figure 6(b), where,

$$\begin{aligned} E_p^s(y_1, y_2; u) &= -uy_1y_2 - uy_1 + uy_2 \\ &= \begin{cases} -u & (y_1, y_2) \in S^s \\ 3u & (y_1, y_2) = (-1, 1). \end{cases} \end{aligned} \tag{16}$$

We set $\phi_p^s(y_1, y_2; q) \propto \exp(-E_p^s(y_1, y_2; u))$ and $q = \exp(-4u)$. Again a larger u means a stronger subsumption relation. When $u \to +\infty$, Equation (16) reduces to the absolute subsumption relation; when $u = 0$, Equation (16) reduces to the absolute overlapping relation.

The product of the pairwise factor functions $\phi_p(y_i, y_j; q_{ij})$ can now be written in terms of the sum of pairwise energy functions $E(y_i, y_j; u_{ij})$ to characterize the label relations of the pHEX graph G. In particular,

$$\begin{aligned} &\Phi_p^G(\mathbf{y}) = \prod_{(i,j)\in G} \phi_p(y_i, y_j, q_{ij}) \\ &\propto \exp(-E(\mathbf{y})) = \exp\left(-\sum_{(i,j)\in G} E(y_i, y_j; u_{ij})\right) \\ &= \exp\left(-\sum_{(i,j)\in ex.} E_p^e(y_i, y_j; u_{ij}) - \sum_{(k,l)\in sub.} E_p^s(y_k, y_l; u_{kl})\right) \\ &= \exp\left(-\sum_{(i,j)\in G} J_{ij} y_i y_j - \sum_{i=1}^{n} h_i y_i\right), \end{aligned}$$

where

$$J_{ij} = \begin{cases} u_{ij}, & (i,j) \in ex. \\ -u_{ij}, & (i,j) \vee (j,i) \in sub. \end{cases} \tag{17}$$

$$h_i = \sum_{\{j|(i,j)\in ex.\}} u_{ij} - \sum_{\{k|(k,i)\in sub.\}} u_{ki} + \sum_{\{l|(i,l)\in sub.\}} u_{il}. \tag{18}$$

Here we denote $ex.$ to be the set containing all exclusive relations and $sub.$ the set containing all subsumption relations. Note that all the pairs $(i,j) \in ex.$ satisfy $i < j$, and pairs $(i,j) \in sub.$ means i subsumes j.

To incorporate local evidence into the model, we can rewrite Equation (4) as follows:

$$p(\mathbf{y} \mid \mathbf{z}) \propto \exp\left(\sum_{i=1}^{n} \log \psi(y_i, z_i) - \sum_{(i,j)\in G} E(y_i, y_j; u_{ij})\right). \tag{19}$$

We can rewrite the exponent as follow:

$$\begin{aligned}
&\sum_{i=1}^{n} \log \psi(y_i, z_i) - \sum_{(i,j)\in G} E(y_i, y_j; u_{ij}) \\
=&\sum_{i=1}^{n} y_i z_i - \sum_{(i,j)\in G} J_{ij} y_i y_j - \sum_{i=1}^{n} h_i y_i \\
=&-\sum_{(i,j)\in G} J_{ij} y_i y_j - \sum_{i=1}^{n} (h_i - z_i) y_i,
\end{aligned} \tag{20}$$

where J_{ij} and h_i are from Equation (17) and Equation (18). Note that we omitted a constant from the logistic function $\log \psi$ in the equations because they can be absorbed in the normalization constant Z. By defining $h'_i = h_i - z_i$, we can "absorb" the local evidence into the Ising model, and use standard inference methods.

5. INFERENCE IN PHEX MODELS

At test time, we need to compute the marginal distribution per label, $p(y_i|\,\mathbf{z})$. In multi-label classification problems, a label y_i is predicted to be true if $p(y_i|\,\mathbf{z}) \geq 0.5$. In multi-class classification problems, the label

$$y^* = \underset{i=1}{\overset{n}{\operatorname{argmax}}}\, p(y_i|\,\mathbf{z})$$

is predicted to be the true label. At training time, we need $p(y_i|\,\mathbf{z})$ as well as the term $p(y_i|y_j = 1, \mathbf{z})$, where some of the true observed labels (eg for node j) are set to their desired target states, as we explain in Section 6. In this section, we explain how to comptue these quantities (see Algorithm 1 for the pseudo code).

Algorithm 1: Testing Phase

Input : pHEX graph G, edge strength q, neural network parameters $\mathbf{w}$, input features $\{\mathbf{x}^{(b)}\}$
Output: $p(y_i|\,\mathbf{x}^{(b)})$ for all labels i and examples b
Compute J_{ij}, h_i using Equation (17), (18);
for $b = 1, 2, \ldots$ **do**
  $\mathbf{z}^{(b)} = DNN(\mathbf{x}^{(b)}; \mathbf{w})$;
  Update local potentials of pHEX graph with $h'_i = h_i - z_i^{(b)}$ for all i;
  Run LBP to obtain $p(y_i|\,\mathbf{z}^{(b)})$ for all i;
  Output $p(y_i|\,\mathbf{z}^{(b)})$ for prediction.
end

Exact inference in pHEX models is usually intractable, when the graphs are loopy, and the legal states are not sparse. Since $p(\mathbf{y} \mid \mathbf{z})$ is an Ising model, we can apply any off-the-shelf inference method, including mean-field inference (MF), loopy belief propagation (LBP), and Markov Chain Monte Carlo (MCMC) methods [3]. In practice, we find that the standard LBP algorithm works consistently well, so we use it as our main inference algorithm in our experiments. We give the details below.

5.1 Belief propagation in pHEX models

We define the belief on each label y_i to be $b_i(-1)$ and $b_i(1)$, and the message from y_i to its neighbour y_j to be $m_{i\to j}(-1)$ and $m_{i\to j}(1)$. Then the algorithm iterates through all beliefs and messages with updates,

$$\begin{aligned}
b_i(1) &\propto \exp(-h'_i) \prod_{j\in N(i)} m_{j\to i}(1), \\
b_i(-1) &\propto \exp(h'_i) \prod_{j\in N(i)} m_{j\to i}(-1), \\
m_{j\to i}(1) &\propto \exp(-J_{ij}) \frac{b_i(1)}{m_{i\to j}(1)} + \exp(J_{ij}) \frac{b_i(-1)}{m_{i\to j}(-1)}, \\
m_{j\to i}(-1) &\propto \exp(-J_{ij}) \frac{b_i(-1)}{m_{i\to j}(-1)} + \exp(J_{ij}) \frac{b_i(1)}{m_{i\to j}(1)},
\end{aligned}$$

where $N(i)$ denotes the neighbours of i.

To maintain numerical stability, we normalize b_i and $m_{j\to i}$ throughout the inference and keep updating on the logarithm domain to avoid overfloating. After all beliefs have converged or a maximum number of iterations has reached, we estimate the marginal probabilities of y_i by,

$$p(y_i = 1|\,\mathbf{z}) = b_i(1).$$

The inference of $p(y_i|y_j = 1, \mathbf{z})$ is almost the same as the one of $p(y_i|\,\mathbf{z})$, except we set $b_j(1) = 1$ and $b_j(-1) = 0$ to represent the fact that node j is clamped to state 1. (We can extend this procedure if we have multiple clamped nodes.)

5.2 Modified BP when we have mutually exclusive labels

Sometimes we have some hard constraints as well as soft constraints. For example, we often have k mutually exclusive binary variables, which is equivalent to a single multinomial variable with $k+1$ possible states. By replacing the clique on the k binary nodes with a single multinomial node, we reduce the number of loops, improving inference speed and accuracy [13, 23]. Although an undirected graphical model with multinomial nodes is strictly speaking not an Ising model, a slight variant on the standard LBP algorithm can still be applied for efficient approximate inference.

For simplicity, we only illustrate the inference algorithm for pHEX graphs with one multinomial label node, since this will be used in later experiments. Further generalization to pHEX graphs with multiple cliques is straightforward and follows similar procedures.

We first group the k nodes that belong to the exclusive clique together as a clique node and denote it to be $c = \{c_1, \ldots, c_k\}$. The standard nodes and messages updates rules are the same as the standard LBP algorithm. The belief of the clique node c is updated as,

$$b_c(i) \propto \exp(-h'_i) \prod_{j\in N(c)} m_{j\to c}(i)$$

for state $i \in \{1, \ldots, k\}$ in which $y_{c_i} = 1$, and

$$b_c(0) \propto \prod_{j\in N(c)} m_{j\to c}(0)$$

for state 0 in which $y_{c_i} = -1$ for all $i \in \{1, \ldots, k\}$. Here $N(c) = \cup_{i=1}^{k} N(c_i)$ is the neighbour set of the clique node.

The message from a standard node j to the clique node c is,

$$m_{j\to c}(i) \propto \exp(\sum_{s=1}^{k} J_{jc_s} - 2J_{jc_i})\frac{b_j(1)}{m_{c\to j}(1)} + \exp(-\sum_{s=1}^{k} J_{jc_s} + 2J_{jc_i})\frac{b_j(-1)}{m_{c\to j}(-1)}$$

for state i, and

$$m_{j\to c}(0) \propto \exp(\sum_{s=1}^{k} J_{jc_s})\frac{b_j(1)}{m_{c\to j}(1)} + \exp(-\sum_{s=1}^{k} J_{jc_s})\frac{b_j(-1)}{m_{c\to j}(-1)}$$

for state 0. The message from the clique node to a standard node j is,

$$m_{c\to j}(1) \propto \sum_{i=1}^{k}\exp(\sum_{s=1}^{k} J_{jc_s} - 2J_{jc_i})\frac{b_c(i)}{m_{j\to c}(i)} + \exp(\sum_{s=1}^{k} J_{jc_s})\frac{b_c(0)}{m_{j\to c}(0)},$$

$$m_{c\to j}(-1) \propto \sum_{i=1}^{k}\exp(-\sum_{s=1}^{k} J_{jc_s} + 2J_{jc_i})\frac{b_c(i)}{m_{j\to c}(i)} + \exp(-\sum_{s=1}^{k} J_{jc_s})\frac{b_c(0)}{m_{j\to c}(0)}.$$

Same as the standard LBP algorithm, we normalize b_c, $m_{j\to c}$ and $m_{c\to j}$ and update them on the logarithm domain throughout the inference procedure. After the algorithm converges, the marginal probability of a node c_k in clique c is,

$$p(y_{c_k} = 1|\,\mathbf{z}) = b_c(k).$$

6. LEARNING

An important property of the (p)HEX model is that not all the target labels need to be specified during training. For example, consider a data set of images. It is more common for a user to use basic level category names, such as "dog", than very specific names such as "husky" or "beagle". Furthemore, a user may not label everything in an image. So the absence of a label is not evidence of its absence.

To model this, we allow some of the labels to be unobserved or hidden during training. For example, if we clamp the "husky" label to true, and leave all other label nodes unclamped, the hard constraints will force the "dog" label to turn on, indicating that this instance is an example of both the husky class and the dog class. However, if we clamp the "dog" label to true, we will not turn on "husky" or "beagle", since the relation is asymmetric. We can also clamp labels to the off state, if we know that the corresponding class is definitely absent. For example, turning on "dog" will turn off "cat" if they are mutually exclusive. (In the pHEX case, the "illegal" states are down weighted, rather than given zero probability.)

Let the input scores for the b'th training instance be $\mathbf{z}^b$, and let the subset of target labels be $\mathbf{t}^b = (t_1^b, \ldots, t_m^b)$, where we have assumed that m labels are observed in every instance for notational simplicity. A natural loss function is the negative log likelihood of the observed labels given the inputs:

$$L = \sum_{b=1}^{N}\sum_{j=1}^{m} -\log p(y_{t_j}^b = 1|\,\mathbf{z}^b) = \sum_{b=1}^{N}\sum_{j=1}^{m} -\log \sum_{\{\mathbf{y}\,|\,y_{t_j}^b=1\}} p(\mathbf{y}\,|\,\mathbf{z}^b). \quad (21)$$

Although the q_{ij} coefficients of the pHEX graph can be trained, in this paper we set $q_{ij} = q$ by cross validation, and focus on optimizing the loss wrt the model parameters $\mathbf{w}$, controlling the mapping from $\mathbf{x}$ to $\mathbf{z}$. In particular, we implement the pHEX graph as a layer on top of a deep feed-forward neural network (based on DistBelief framework [6]), as shown in Figure 7.

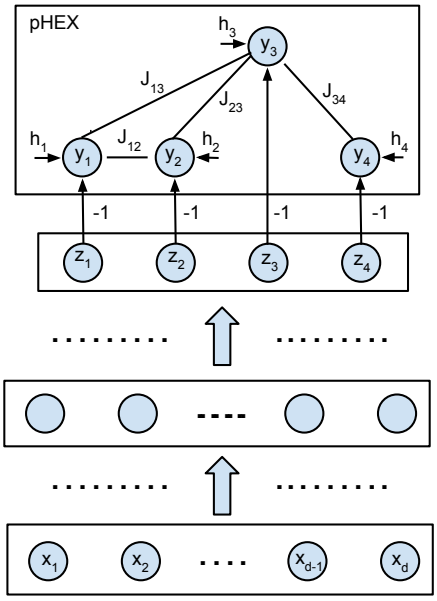

Figure 7: The pHEX graph as a top layer of a deep feed-forward neural network.

6.1 Backpropagating the error to the local classifers

To fit the local (label specific) classifiers, we need to derive the gradient of the loss wrt the input scores z_i. We show how to do this below, considering a single term b in the above sum.

The derivative of $\log p(y_{t_j} = 1|\,\mathbf{z})$ over some z_i is,

$$\frac{\partial \log p(y_{t_j} = 1|\,\mathbf{z})}{\partial z_i} = \frac{1}{p(y_{t_j} = 1|\,\mathbf{z})}\frac{\partial p(y_{t_j} = 1|\,\mathbf{z})}{\partial z_i} = \frac{1}{p(y_{t_j} = 1|\,\mathbf{z})}\sum_{\{\mathbf{y}\,|\,y_{t_j}=1\}}\frac{\partial p(\mathbf{y}\,|\,\mathbf{z})}{\partial z_i}$$

$$= \frac{1}{p(y_{t_j}=1|\,\mathbf{z})} \sum_{\{\mathbf{y}\,|y_{t_j}=1\}} p(\mathbf{y}\,|\,\mathbf{z}) \frac{\partial \log p(\mathbf{y}\,|\,\mathbf{z})}{\partial z_i}$$

$$= \sum_{\{\mathbf{y}\,|y_{t_j}=1\}} p(\mathbf{y}\,|y_{t_j}=1,\mathbf{z}) y_i - \sum_{\mathbf{y}} p(\mathbf{y}\,|\,\mathbf{z}) y_i$$

$$= \mathbb{E}_{p(y_i|y_{t_j}=1,\mathbf{z})}[y_i] - \mathbb{E}_{p(y_i|\,\mathbf{z})}[y_i]. \quad (22)$$

Therefore, in order to evaluate the gradient of the loss function L in Equation (21), we need to compute the conditional distributions $p(y_i|y_{t_j}=1,\mathbf{z})$ and marginal distributions $p(y_i|\,\mathbf{z})$ for all i. These correspond to the well-known "clamped" and "unclamped" phases of MRF / CRF learning.

6.2 Details of the learning algorithm

At the beginning of training, the pHEX graph of the candidate labels is converted to its equivalent Ising model with coefficients J_{ij} and h_i using Equation (17) and Equation (18) which are stored inside the pHEX layer and used throughout the training. During training, each training example's inputs $\mathbf{x} = \{x_1, \ldots, x_d\}$ forward propagates through the feed-forward network and outputs $\mathbf{z}$ as the inputs of the pHEX graph. The size of $\mathbf{z}$ is the same as the size of $\mathbf{y}$. Each y_i takes $-z_i$ as the input and combines it with the pre-stored h_i according to Equation (20) to obtain the local potential of the resulting Ising model for inference.

The output of the pHEX layer is different in training and testing phases. During the training phase, we use stochastic gradient descent to minimize the loss function Equation (21). To this end, we first perform an LBP inference to estimate $p(y_i|y_{t_j}=1,\mathbf{z})$ and $p(y_i|\,\mathbf{z})$, and compute the derivative $\partial L/\partial\,\mathbf{z}$ based on Equation (22). Then we back-propagate the derivative through the z_i's to the whole network. Therefore, the output of the pHEX layer during training is $\partial L/\partial z_i$ to each z_i. The training algorithm of the entire system is summarized in Algorithm 2.

Algorithm 2: Training Phase

Input : pHEX graph G, edge strength q, labeled data $\{\mathbf{x}^{(b)}, \mathbf{t}^{(b)}\}$
Output: Neural network parameters $\mathbf{w}$
Initialize $\mathbf{w} = \mathbf{w}^{(0)}$;
Compute J_{ij}, h_i using Equation (17), (18);
for $b = 1, 2, \ldots$ **do**
  $\mathbf{z}^{(b)} = DNN(\mathbf{x}^{(b)}; \mathbf{w})$;
  Update local potentials of pHEX graph with $h'_i = h_i - z_i^{(b)}$ for all i;
  Run LBP to obtain $p(y_i|\,\mathbf{z}^{(b)})$ for all i;
  Run LBP to obtain $p(y_i|y_{t_k}, \mathbf{z}^{(b)})$ for all k and i;
  Evaluate $\partial L/\partial\,\mathbf{z}$ using Equation (22) and output to $\mathbf{z}^{(b)}$;
  Back propagate the gradient from $\mathbf{z}^{(b)}$ to $\mathbf{x}^{(b)}$ and update $\mathbf{w}$;
end

7. EXPERIMENTS

In [8], we showed that HEX graphs significantly outperform standard softmax and (multi-label) logistic regression models, so in this paper, we will just compare pHEX to HEX. We conduct three experiments.

The first experiment is the standard ImageNet image classification problem [7]. We add hierarchical relations between the labels based on the publicly available WordNet hierarchy. Since WordNet does not have exclusive relations, we assume that any two labels are exclusive if they are not in subsumption relation. Figure 8 is an example of the subgraph of "fish". To make the problem more interesting, we assume that the training labels are drawn from different levels of the hierarchy as in [8]. In this paper, we show that pHEX improves on HEX, especially when leaf labels are rarely present in the training set.

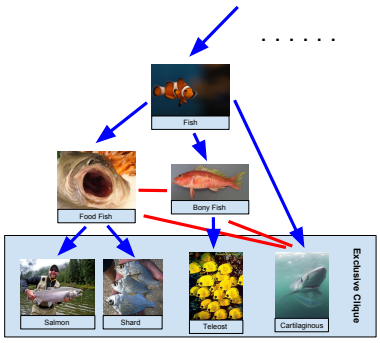

Figure 8: An illustration of the (p)HEX graph based on the WordNet hierarchy in the ImageNet experiments. The blue directed edges denote the subsumption relations; and the red undirected edges denote the exclusive relations. An absolute exclusive clique is placed in the final pHEX graph.

The second experiment is a zero-shot learning task, in which we must predict unseen classes at test time, leveraging known relations between the class labels and attributes of the class. We use the Animals with Attributes dataset [11]. Following [8], we first assume that all object classes are mutually exclusive. We then add subsumption relations from a predicate (or attribute) to an object if the binary predicate of the object is 1, and add exclusive relations between predicate and objects if the binary predicate of the object is 0. See the illustration in Figure 9. In this paper, we relax the hard constraints and show that pHEX can work significantly better than HEX. Finally, the third experiment is another zero-shot learning task, this time on the PASCAL VOC/ Yahoo images with attributes dataset [9]. Again, we show that pHEX can significantly outperform HEX.*

7.1 Experimental setup

In our experiments, we vary the parameter u across the ranges shown in Table 1 to measure the importance of the strength of the relationships between the labels.

Note that, since all three tasks are evaluated on test labels which are mutually exclusive, we add a hard mutually exclusive clique into the pHEX graphs (See Figure 8 and 9). In particular, for the ImageNet dataset, we add an exclusive clique on the 1000 leaf labels; in the Animal with

* Note: All three datasets used in the experiments are publicly available. We have put the HEX graphs and pHEX graphs that we used at https://sites.google.com/site/ssnding/phex/. We will release the pHEX inference code after paper acceptance.

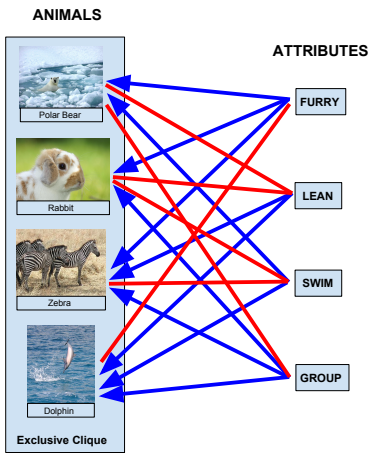

Figure 9: An illustration of the (p)HEX graph in the Animal with Attributes experiments.

Table 1: The Ising coefficients u as well as the corresponding strengths of the label relations q used in pHEX graphs in the experiments, where $q = \exp(-4u)$.

u	0.1	0.3	0.5	0.7	1.0	1.5
q	0.67	0.30	0.14	0.06	0.02	0.002

Attributes dataset, we add an exclusive clique on the 50 animal classes; and in the VOC/Yahoo dataset, we add an exclusion clique on the 32 object classes. After adding the exclusion clique, we remove the replicated soft relations from the pHEX graph.

7.2 ImageNet classification experiments

In this section, we use the ILSVRC2012 dataset [7], which consists of 1.2M training images from 1000 object classes. These 1000 classes are mutually exclusive leaf nodes of a semantic hierarchy based on WordNet that has 860 internal nodes. As in [8], we evaluate the recognition performance in the multiclass classification at the leaf level, but allow the training examples to be labeled at different semantic levels. Since ILSVRC2012 has no training examples at internal nodes, we create training examples for internal nodes by relabelling $\{50\%, 90\%, 95\%, 99\%\}$ of the leaf examples to their immediate parents based on the WordNet Hierarchy. Since the ground truth for test set is not released for ILSVRC2012, we use 10% of the released validation set as our validation set while use the other 90% as our test set to compare classification accuracy.

The underlying feed-forward network that we use is based on a deep convolutional neural network GoogLeNet [20]. Since GoogLeNet consists of a giant network, to speedup our training procedure, we used the pHEX graphs as an add-on on the classification model based on the HEX graphs: we first pretrain the model using a HEX graph as a top layer until converging and store the check points; then we continue training from the check points by replacing the HEX graph with different pHEX graphs (in Section 7.1) on the top layer and obtain the results of the pHEX graphs.

Figure 10 shows the Top-1 (top row) and Top-5 (bottom row) accuracies across classes as a function of u, for the relabeling experiments. We see that in general, pHEX outperforms HEX albert only slightly for 50% relabeling. (Note that although the absolute difference is small, a 1% difference is considered statistically significant due to the size of this dataset.) The improvements appear to be much more significant for 90%, 95% and 99% relabeling, where the Top-1 accuracies improve over 2%, 3% and 8% respectively.

At first, it might seem odd that relaxing the hard constraints imposed by the hierarchy can help, since the hierarchy provided by WordNet is supposed to be correct. However, [8] observed that too few training examples labeled at leaf nodes (especially at 99% relabeling) may confuse the leaf models, especially at the beginning of the training. As the algorithm runs longer, it becomes harder to recover from a bad local minimum because the constraints in the HEX graph can be quite strong. By contrast, in the pHEX graph, the weaker relations between internal nodes and leaf nodes make the resulting posterior distribution smoother, so it is easier to overcome bad local minima for the pHEX graph in later iterations.

It is also interesting to see that the optimal value of u appears to depend on the relabeling percentage. When a larger portion of training examples are relabeled, e.g. 99% relabeling, the optimal relation coefficient are the smallest ($u = 0.1$). This indicates that weaker label relations are preferred when there is more uncertainty in the leaf labels.

On the other hand, when u is large, the label relations become quite certain and the the pHEX graph becomes closer to the HEX graph. In the case of $u = 1.5$ ($q \simeq 0.002$), the performance of pHEX graph can become inferior to HEX graph, due to the inferior inference algorithm in the pHEX graph.

7.3 Zero shot learning experiments

We use two datasets to illustrate zero shot learning. The first is the Animals with Attributes dataset [11], which includes images from 50 animal classes. For each animal class, it provides binary predicates for 85 attributes. We evaluate the zero-shot setting where training is performed using only examples from 40 animal classes (with 24295 images) and testing is on classifying the 10 unseen classes (with 6180 images). Our experimental results are based on 5-fold cross validation. The underlying network is a single-layer network whose inputs come from the recently released DECAF features.

The second dataset is the aPascal-aYahoo dataset [9], which consists of a 12695 image subset of the PASCAL VOC 2008 dataset and 2644 images that were collected using the Yahoo image search engine. The PASCAL part serves as training data and has 20 object classes. The Yahoo part serves as test data and contains 12 different object classes. Each image has been annotated with 64 binary attributes that characterize shape, material and the presence of important parts of the visible object. The underlying network is again a single-layer network whose inputs come from the precomputed color, texture, edge orientation and HoG features that the authors of [9] extracted from the objects bounding boxes (as provided by the PASCAL VOC annotation) and released as part of the dataset. Once again we use 5-fold cross validation.

Figure 11 shows the mean accuracy per class (along with

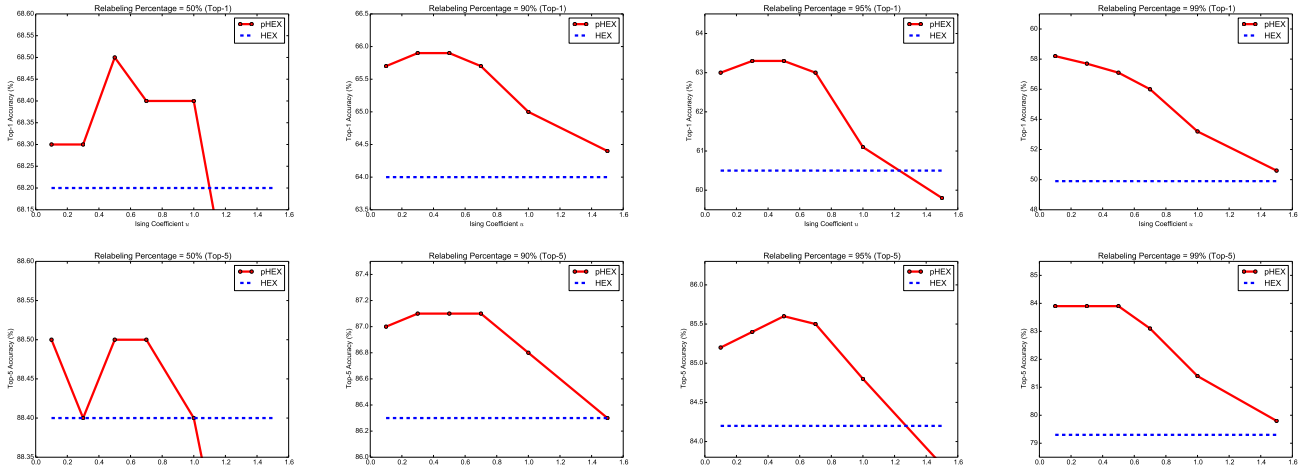

Figure 10: Top-1 (top) and Top-5 accuracies (bottom) vs relation strength u for the ImageNet classification experiment. The results of the pHEX graphs are in the red solid curves, and the results of the HEX graphs are in the blue dashed horizontal lines. From left to right: relabeling 50%, 90%, 95%, 99%.

standard errors) vs u, along with standard errors. We see that pHEX is generally significantly outperforming HEX. In particular, when $u \in [0.1, 1.5]$ for Animals with Attributes and $u \in [0.3, 1.0]$ for VOC/Yahoo, the difference is statistically significant at the 5% level according to a paired t-test. The accuracies of the pHEX graph get closer to the ones of the HEX graph as u becomes larger and the pHEX graph approaches to the HEX graph.

7.4 Speed comparison of HEX vs pHEX

The inference cost of the pHEX graph is proportional to the number of probabilistic relations in the graph. In the ImageNet experiment, the number of edges is about 6000, which is 3 times larger than the number of nodes in graph (which is 1860). However, the total cost of inference is negligible compared to the cost of the underlying deep neural network. In the two zero-shot learning experiments, inference in the pHEX graph takes about 7 to 11 times more compared to the HEX graph; however, the benefits are also correspondingly greater. Furthermore, many other algorithms have been devised for Ising models which we could try in the future.

8. CONCLUSIONS

In this paper, we studied object classification with probabilistic label relations. In particular, we proposed the pHEX graph, which naturally generalizes the HEX graph. The pHEX graph is equivalent to an undirected Ising model, which allows for efficient approximate inference methods. We embed the pHEX graph on top of a deep neural network, and show that it outperforms the HEX graph on a number of classification tasks which require exploiting label relations.

There are several possible future directions of this work. One idea is to learn the Ising coefficients of the pHEX graph together with the underlying neural network parameters. Another is to combine the pHEX graph into a larger framework which exploits spatial relations between the objects. A third possibility is to try to make the relation strengths be context dependent.

9. ACKNOWLEDGMENTS

We acknowledge Changyou Chen, Youhan Fang, and Kai Wang for helpful discussions and comments.

10. REFERENCES

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826. IEEE, 2013.

[2] K. Binder. Ising model, 2001. Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] X. Chen, X.-T. Yuan, Q. Chen, S. Yan, and T.-S. Chua. Multi-label visual classification with label exclusive context. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 834–841. IEEE, 2011.

[5] B. Dalvi, E. Minkov, P. P. Talukdar, and W. W. Cohen. Automatic gloss finding for a knowledge base using ontological constraints. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 369–378. ACM, 2015.

[6] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, P. A. Tucker, K. Yang, and A. Y. Ng. Large scale distributed deep networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NIPS*, pages 1232–1240, 2012.

[7] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. Imagenet large scale visual recognition challenge 2012, 2012. www.image-net.org/challenges/LSVRC/2012.

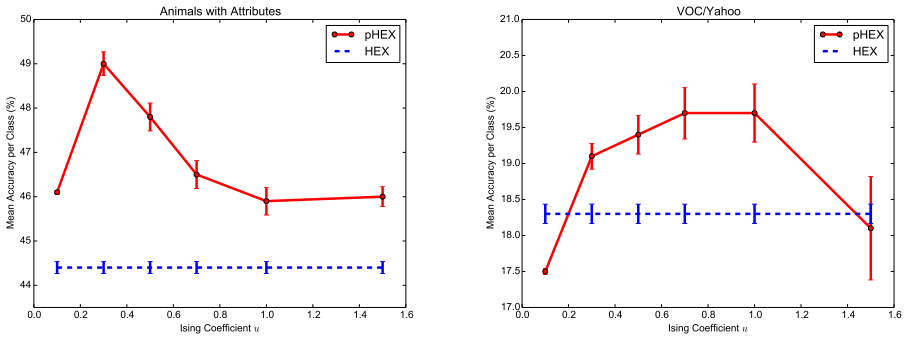

Figure 11: Mean accuracy per class vs relation strength u for the Zero-shot Learning Experiments. Left: animals with attributes. The results of the pHEX graphs are in the red solid curves, and the results of the HEX graphs are in the blue dashed horizontal lines. Right: VOC/Yahoo images with attributes.

[8] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classfication using label relation graphs. In *Proceedings of the 13th European Conference on Computer Vision*, pages 48–64, Zurich, Switzerland, September 2014. Springer.

[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR*, 2009.

[10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[11] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009.

[12] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[13] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *In Proceedings of Uncertainty in AI*, pages 467–475, 1999.

[14] V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg. From large scale image categorization to entry-level categories. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[15] M. Palatucci, G. Hinton, D. Pomerleau, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[16] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[17] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[18] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where - and why? semantic relatedness for knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 910–917. IEEE, 2010.

[19] V. Sharmanska, N. Quadrianto, and C. H. Lampert. Augmented attribute representations. In A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *European Conference on Computer Vision*, volume 7576 of *Lecture Notes in Computer Science*, pages 242–255. Springer, 2012.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 2014. arXiv:1409.4842.

[21] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, Dec. 2005.

[22] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 969–976, New York, NY, USA, 2006. ACM.

[23] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[24] C. Wu, I. Lenz, and A. Saxena. Hierarchical semantic labeling for task-relevant rgb-d perception. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[25] F. Yu, L. Cao, R. Feris, J. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, OR, June 2013.

[26] A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.