Improving Image Understanding with Concept Graph

Libo Yin
The Australian National University

October 16, 2015

1 The Original HEX Model

1.1 Structure

The original HEX model [1] is an extension of the baseline flat multiclass classification model. There are two types of multiclass classification: exclusive, where the classifier predicts exactly one out of all possible states to be true; and independent, where each state is predicted true or false independently. HEX finds a balance between these two ends of the spectrum. It models the hierarchical and exclusive relationship between concepts, extended by their hypernyms, with a semantic graphical model. Each node in this graphical model corresponds to a concept in the extended concept space being true or false. States of neighbouring nodes are constrained in that if a is a hypernym of b, then is not allowed that $a = 0,\ b = 1$; and if a and b are exclusive, then they cannot both be true. HEX model classifies an image into a hierarchy that satisfies the above semantic consistency. A simple HEX graph is shown in figure 1.

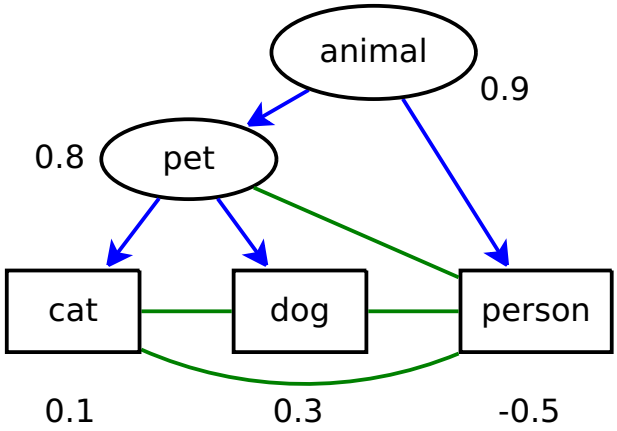

Figure 1: A simple HEX graph with three nodes in the original concept space (in rectangles) and their hypernyms (in ellipsoids). Directed edges denote semantical subsumption: $(a \to b)$ if a is a hypernym of b according to WordNet; and undirected edges denote exclusion. Since exclusive relationship is not covered by WordNet, the exclusive subgraph is initialized greedily: two concepts are exclusive unless they share a common descendant in the hierarchical subgraph. Note that the hierarchical subgraph is in general a DAG rather than a tree; and it is not necessarily the case that all concepts in the original concept space are bottom-level nodes in the hierarchical subgraph. Values beside each nodes are to be used later.

Denoting the set of vertices by V, the set of hierarchical edges by E_h, and the set of exclusive

edges by E_e, the joint assignment $y \in \{0,1\}^V$ can be defined as a conditional random field:

$$\tilde{p}(y|x) = \prod_{i \in V} \exp\{f_i(x;w) I[y_i = 1]\} I[y \text{ legal}] \tag{1}$$

where unary input $f_i(x;w)$ is the confidence on concept i, predicted by an arbitrary underlying classifier[1]. Local semantic constraints on hierarchical and exclusive edges is formalised by

$$I[y \text{ legal}] = \prod_{(v_i,v_j) \in E_h} I[(y_i, y_j) \neq (0,1)] \prod_{(v_i,v_j) \in E_e} I[(y_i, y_j) \neq (1,1)] \tag{2}$$

Thanks to these semantic constraints, the state space of a HEX graph is significantly smaller than independent multiclass classification on the same concept space. For example, with local confidence given beside each node in figure 1, the valid assignments of the HEX graph and their respective potentials are listed in table 1:

assignment	$\tilde{p}(y\|x)$
$\varnothing$	$\exp(0)$
{animal}	$\exp(0.9)$
{animal, pet}	$\exp(0.9 + 0.8)$
{animal, person}	$\exp(0.9 - 0.5)$
{animal, pet, cat}	$\exp(0.9 + 0.8 + 0.1)$
{animal, pet, dog}	$\exp(0.9 + 0.8 + 0.3)$

Table 1: Valid assignments of the HEX graph in figure 1 and their respective potentials. In this example, the HEX model predicts {animal, pet, dog}.

Finally, an important assumption of [1] is that mechanical Turks tend to label an image to more general concepts. For example, an image of a yellow Labrador may be labelled to "dog". Such behaviour is modelled by randomly relabelling an image to its immediate parents. As will be shown in section 2.3, the higher the relabelling rate is, the larger is the advantage of the HEX model compared to the softmax baseline.

1.2 Observations

Note that a valid assignment does not have to have an active node in the original concept space. (Actually, it does not have to have an active node at all, as $\varnothing$ is a valid assignment according to (2).) This allows an image to be classified to abstract concepts when the classifier is not confident enough to classify to a more concrete one. This is by no doubt a desirable feature for deployment. However, since all images are labelled in the original concept space of the dataset, classifying to the extended concept space makes performance evaluation troublesome. This issue was not addressed in [1]. In this work, accuracy is tested both in the extended concept space and in the original one, by limiting the legal state space to assignments with an active bottom-level node in the hierarchical subgraph. This is only valid for datasets where all concepts in the original concept space are bottom-level nodes in the HEX graph.

Also note that in table 1, the competition between assignment {animal, pet, cat} and {animal, pet, dog} depends entirely on the confidence of node "dog" and "cat". However, to discriminate between {animal, pet, dog} and {animal, person} requires examining the confidence

[1] According to [1]. It shall be explained in section 1.2 that the choice of underlying unary classifier is actually not arbitrary.

along different paths. From this example it is clear that, in the testing stage, confidence is passed down the hierarchy from more abstract concepts to more concrete ones. The other side of the same coin is that, during the training stage of the underlying unary classifier, a node corresponding to abstract concepts receive all training data of its children. This can be seen as confidence being passed up in the hierarchy, from more concrete concepts to more abstract ones. Such bidirectional propagation of confidence explains the advantage of the HEX model under the realistic labelling assumption. Note that the exclusive subgraph does not take part in the propagation of confidence.

The third observation is a combined consequence of the greedy exclusion setup, as discussed under table 1, and the greedy nature of the potential function (1). The prerequisite for the original HEX model is that the decision boundary of $f_i(x; w)$ is zero for all i. A unary prediction above zero gives support to a node being true, whereas a unary prediction below zero gives support to that node being false. In [1] this is not a problem, as a convolutional neural network [6] is used as the underlying unary classifier; and that the a neural network can learn its decision boundary form training data. However, if a probabilistic classifier is used, in which case $f_i(x; w) \in [0, 1]$ and the decision boudary is 0.5, then it is guaranteed that a bottom-level node will be activated (see appendix A for proof). This means that the effective state space size is reduced to the number of bottom-level nodes. In other words, the HEX model will not be able to predict to an abstract concept. This defeats the purpose of HEX, although a smaller state space means faster inference and higher accuracy.

Finally, listed below are two less important observations:

1. There are no learnable variables in this CRF. In other words, all learning is performed in the underlying unary classifier.

2. Mathematically, CRF requires $\forall y : \tilde{p}(y|x) > 0$. However, computationally, assigning zero to $\tilde{p}(y|x)$ can be interpreted as assigning an infinitesimal value. Therefore, the above definition is computationally a legitimate CRF.

1.3 Problems

Consider the situation illustrated in figure 2. Following the same logic as table 1, the original HEX model predicts {animal, pet, dog, Labrador}. However, this is a result of more active nodes rather than active nodes of higher confidence. Intuitively, {animal, person} seems a better prediction. In addition, the two aforementioned assignments are only separated by 0.1 in log unnormalised potential. It will be desirable if the model can make a prediction that is further from the decision boundary by taking into account the confidence of other nodes, especially those with unary predictions far from the decision boundary.

The second problem is a combined consequence of the realistic labelling assumption and using CNN as the underlying unary classifier. When the relabelling rate is high, most images are relabelled to their immediate parents. This means that for each concept X, most instances of it are correctly labelled to all its ancestors, but incorrectly labelled "NOT X". This create a highly unbalanced and internally inconsistent dataset. In such case, because CNN is trained as an independent multiclass classifier, the unary classifiers will simply learn to predict "false" for any input images. (This hypothesis will be confirmed in section 2.3.) The original HEX model cannot cope with such situation, as the potential function (1) will deactivate a bottom-level node as long as its confidence is below the decision boundary.

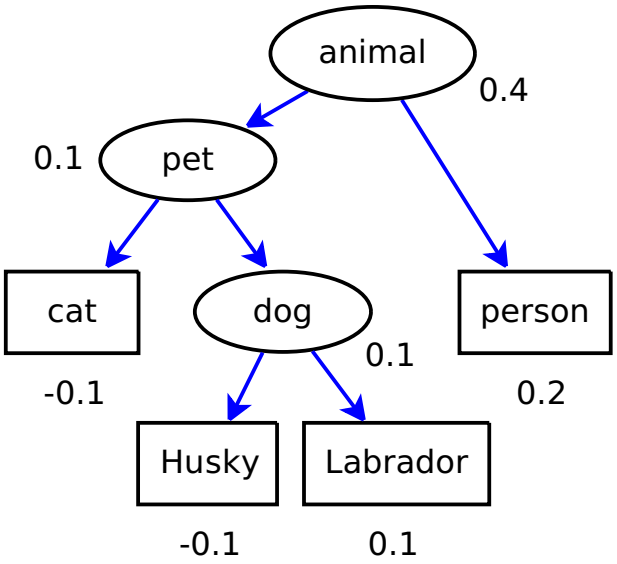

Figure 2: Where an assignment with more active nodes of lower confidence wins over an assignment with fewer active nodes of higher confidence. Exclusive edges are not drawn, as they do not carry further information once the state space has been calculated.

2 Reimplementation

2.1 Dataset

The original HEX model is reimplemented as the baseline of this work. While [1] used ILSVRC 2012 dataset for its rich structure in the label space, this work employs PASCAL VOC 2012 [3] for its simplicity (see figure 3 for the complete hierarchical subgraph). Note that the two datasets are designed for different tasks: ILSVRC is an exclusive multiclass classification problem, whereas PASCAL is an independent multiclass classification problem. To adapt to ILSVRC task 1, the train + val dataset of the PASCAL dataset is filtered through the following criteria:

1. If there is only one annotated object in the image, the image is labelled to that object.
2. If the largest annotated object in the image is more than twice as large as the second largest one, the image is labelled to its dominating object.

Another major difference between ImageNet and PASCAL is that the former one is a balanced dataset, while the latter is highly unbalanced. For example, after filtering, there are 5324 images labelled as "person", whereas the second most frequent label "dog" has only 817 instances. To rebalance the dataset, 950 images of "person" are subsampled from the dataset. After preprocessing, the dataset contains 8473 images in total. These images are then split 3:1:1 into train/validation/test set. The distribution of images across labels is illustrated in figure 4.

2.2 Algorithms

For consistency with [1], the underlying unary classifier is based on [6] rather than the current state-of-the-art [8]. The CNN is fine-tuned as an independent multiclass classifier with Caffe [5], where each image is projected to a 27-dimensional vector. Note that since the original HEX model has no learning part, it was built as a layer into the CNN, achieving end-to-end learning. In this work, CNN and CRF are implemented separately in order to test multiple variants of the HEX model with the same underlying unary classifier. The fine-tuning process finished within five hours on a GeForce GTX 470 graphical card.

One consequence of a simpler concept space is the change of inference algorithm. In [1], inference is performed by MAP loopy belief propagation on the HEX graph, where the local state space of each clique is limited by semantic restrictions. In this work, due to the tiny concept space of dataset, the inference is performed by calculating the potential function directly for each possible states in the global state space.

Finally, to guarantee that the benefit of the HEX model is consistent regardless of the underlying unary classifier, different variants of the HEX model are also paired with SVM as the underlying unary classifier. An array of 27 SVMs is trained, each corresponding to a node in the HEX graph. The SVM array accepts the output of the last-but-one layer of [6] as input, assuming that 4096 neurons provide sufficiently diverse and accurate feature responses without fine-tuning on the PASCAL dataset. The SVM array predicts either distance to decision boundaries (corresponding to raw CNN output), or the probability of each concept being activated (corresponding to CNN output with sigmoid transformation). The training process for the SVM array finished within ten hours without parallel training.

2.3 Experiments

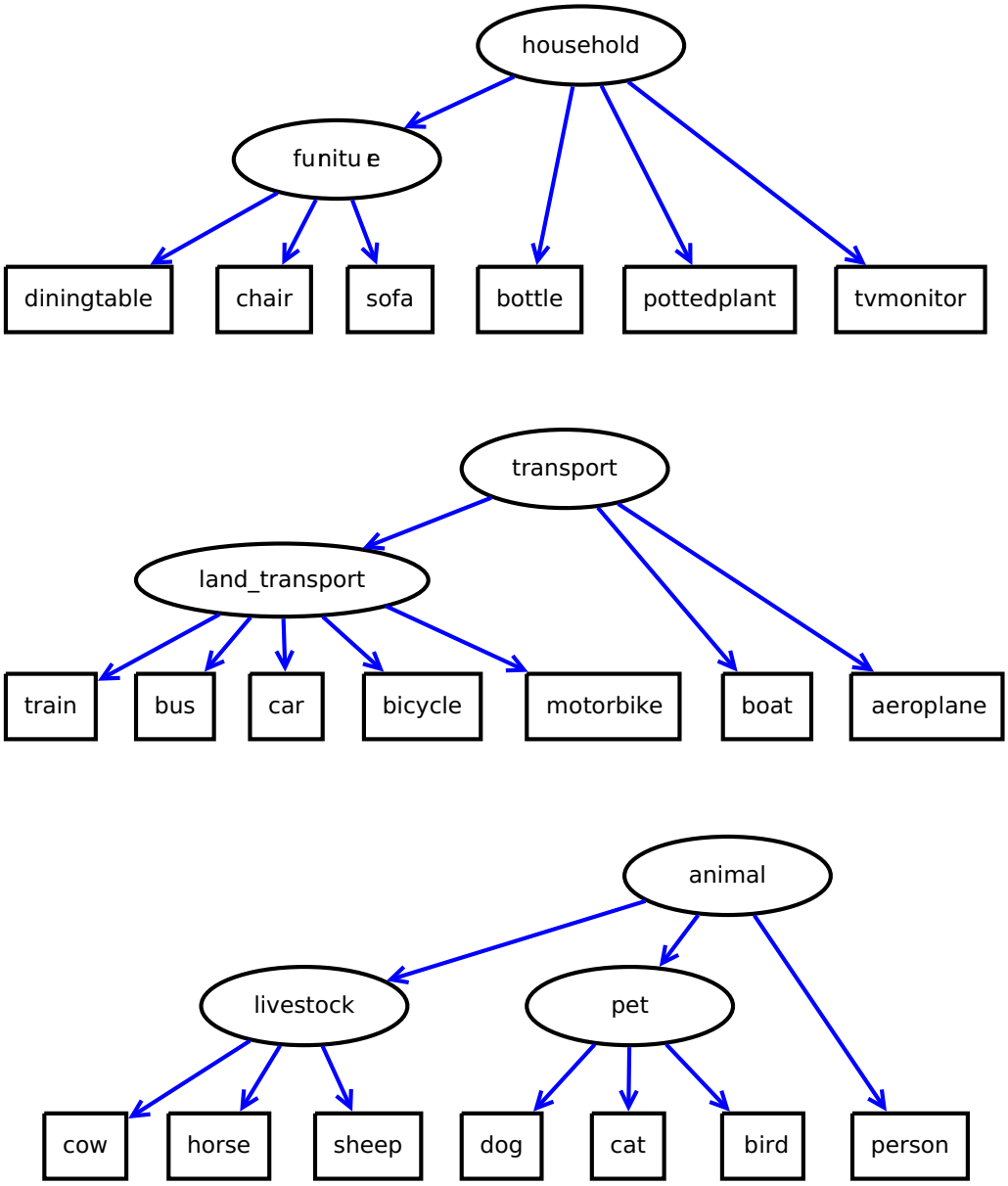

Figure 3: The PASCAL dataset has 20 concepts. In this work, these concepts are augmented by 7 of their hypernyms, creating a forest of three trees with 27 nodes and 24 hierarchical edges. Exclusive edges are not drawn since they can be implied by the hierarchical subgraph. The state space of this HEX graph has size 28. In comparison, the HEX graph in [1] contains 1000 nodes corresponding to the original concept space, and 820 nodes corresponding to their hypernyms.

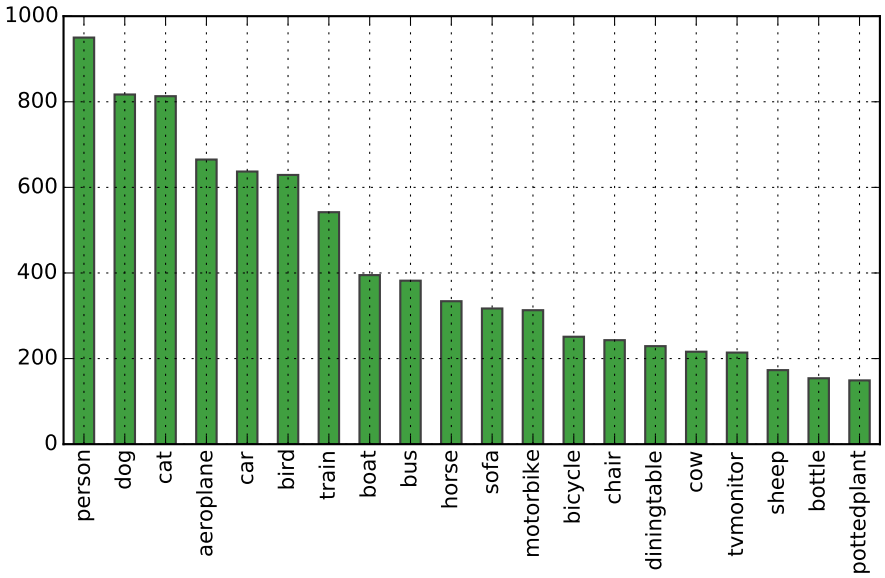

Figure 4: Distribution of images according to labels, after preprocessing

	0%	50%	90%	95%
Softmax	0.626(0.843)	0.564(0.796)	0.529(0.772)	0.508(0.760)
Logistic	N/A	0.210(0.452)	0.093(0.272)	0.056(0.172)
HEX	N/A	0.582(0.808)	0.553(0.794)	0.524(0.772)
Softmax	0.7686	0.3997	0.0148	N/A
Logistic	0.171(0.733)	0.004(0.625)	0.000(0.172)	N/A
HEX	0.657(0.777)	0.318(0.719)	0.000(0.477)	N/A

Table 2: Comparison of empirical test result in [1] (upper) with reimplementation in this work (lower). Performance is reported in top-1 accuracy (top-3 accuracy). Due to smaller dataset, 95% relabelling rate was not attempted in this work. Detailed analysis of the reimplemented system is provided in section 3.2. Note that while the original HEX model managed to beat the baseline in [1], it falls below the baseline on the PASCAL dataset.

	0%	50%	90%
diningtable	0.4285 ± 0.3011	0.2490 ± 0.2288	0.0108 ± 0.0116
furniture	0.6976 ± 0.3298	0.7524 ± 0.2784	0.7330 ± 0.3065
household	0.8431 ± 0.2729	0.8598 ± 0.2490	0.8590 ± 0.2494
dog	0.7517 ± 0.3424	0.4533 ± 0.2970	0.0222 ± 0.0209
pet	0.7971 ± 0.3271	0.7828 ± 0.3327	0.7832 ± 0.3287
animal	0.9004 ± 0.2301	0.9005 ± 0.2290	0.8999 ± 0.2322

Table 3: Under different relabelling rates, with fully trained CNN, the mean and standard deviation of response from validation images labelled "diningtable" (upper) and "dog" (lower). Note that "household" is a hypernym of"furniture", which is further a hypernym of "dog"; and "animal" is a hypernym of "pet", a further hypernym of "dog". With 490 instances in the training set, label "dog" is a frequent concept, whereas rare concept "diningtable" has 137 instances. Clearly, the unary confidence on abstract concepts are not affected by relabelling rate, yet the unary response on concrete ones drop to below the decision boundary. This experiment also shows that CNN learns highly abstract concepts accurately, even for concepts without obvious visual clues.

3 Improvements

3.1 Algorithms

Based on the original HEX model, this work delivers improvements in three stages. The high-level goal is stated as follows: With little confidence on bottom-level nodes, the classifier should still attempt to classify to an assignment with an active bottom-level concept. However, in case the classifier really cannot make a decision, it should still be allowed to predict to an assignment in the extended concept space.

In the first stage, the confidence on inactive nodes are taken into account, in addition to active ones. The resulting potential function is shown as follows:

$$\tilde{p}(y|x) = \prod_{i \in V} \exp\{f_i(x;w)I[y_i = 1] + (1 - f_i(x;w))I[y_i = 0]\} I[y \text{ legal}] \tag{3}$$

where $f_i(x;w)$ denotes the confidence of node i being true, and $1 - f_i(x;w)$ denotes the confidence of node i being false. This requires the range of unary prediction to be within $[0, 1]$, and the decision boundary to be 0.5. To achieve this, unary predictions from the underlying CNN is normalised with the sigmoid function.

Theoretically, stage 1 fix should benefit in two aspects: First, the potential function considers the same number of nodes for every legal assignment, therefore fixing the unnormalised depth problem discussed in section 1.3. Second, the potential function is able to refer to the confidence on inactive nodes, hence benefiting from those that are far from the decision boundary. However, a massage on (3) shows that it is not far from the original potential function (1):

$$\begin{aligned}
\tilde{p}(y|x) &= \prod_{i \in V} \exp\{f_i(x;w)I[y_i = 1] + (1 - f_i(x;w))(1 - I[y_i = 1])\} I[y \text{ legal}] \\
&= \prod_{i \in V} \exp\{(2f_i(x;w) - 1)I[y_i = 1] + (1 - f_i(x;w))\} I[y \text{ legal}] \\
&= \tilde{p}(\varnothing|x) \prod_{i \in V} \exp\{(2f_i(x;w) - 1)I[y_i = 1]\} I[y \text{ legal}]
\end{aligned} \tag{4}$$

where $\tilde{p}(\varnothing|x)$ denotes the unnormalised potential of assignment $\varnothing$. Since it is a constant for all y, it can be absorbed into $\frac{1}{Z(x)}$. Also, the transformation on $\exp\{(2f_i(x;w) - 1)I[y_i = 1]\}$ does not affect the rank of $\tilde{p}(y|x)$ for all y. Therefore, the difference between (1) and (3) is no more than the sigmoid transformation.

In the second stage, pairwise term is added to the potential function. This can be seen as a remedy to the low confidence on bottom-level nodes when the relabelling rate is high, as the confidence on their immediate parents are not affected. In addition, two regularisation terms are added to the potential function so that the contributions from unary terms and pairwise terms are balanced. The resulting potential function is shown as follows:

$$\begin{aligned}
p(y|x) = \frac{1}{Z(x)} &\exp\left\{\frac{1}{|V|} \sum_{i \in V} x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0]\right\} \\
&\cdot \exp\left\{\frac{1}{|E_h|} \sum_{(i,j) \in E_h} x_i x_j \cdot I[y_i = y_j = 1]\right\} \cdot I[y \text{ legal}]
\end{aligned} \tag{5}$$

In the third stage, a weighting factor is multiplied to each unary and pairwise term. This weighting factor is learned form all images labelled to bottom-level nodes in the dataset. Note that such training scheme cannot cover concepts from the original label space that are not

bottom-level nodes in the HEX graph, as it is not possible to distinguish between images that are correctly labelled to that concept, or images that are labelled to its immediate parents. The resulting potential function is shown as follows:

$$p_\theta(y|x) = \frac{1}{Z(x)} \exp\left\{ \frac{1}{|V|} \sum_{i \in V} w_i \big(x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0] \big) \right\} \cdot \exp\left\{ \frac{1}{|E_h|} \sum_{(i,j) \in E_h} t_{ij} \cdot x_i x_j \cdot I[y_i = y_j = 1] \right\} \cdot I[y \text{ legal}] \tag{6}$$

where θ is the concatenation of $\{\forall i \in V : w_i\}$ and $\forall (i,j) \in E_h : t_{ij}$. θ is selected by optimising for log-likelihood:

$$\theta = \arg\min_\theta \left\{ -\frac{C}{N} \log \prod_{(x,y) \in D} p_\theta(y|x) + \frac{1}{2} \|\theta\|^2 \right\} \tag{7}$$

where the weighting of regularisation term $\frac{1}{2}\|\theta\|^2$ is controlled by C, a constant chosen by cross-validation. In this work, $C = 1000$. The calculation of gradient is shown as follows:

$$\begin{aligned} \nabla_\theta \log \prod_{(x,y) \in D} p_\theta(y|x) &= \begin{bmatrix} \nabla_w \log \prod_{(x,y) \in D} p_\theta(y|x) \\ \nabla_t \log \prod_{(x,y) \in D} p_\theta(y|x) \end{bmatrix} \\ &= \begin{bmatrix} \sum_{(x,y) \in D} \nabla_w \log p_\theta(y|x) \\ \sum_{(x,y) \in D} \nabla_t \log p_\theta(y|x) \end{bmatrix} \end{aligned} \tag{8}$$

$$\begin{aligned} \log p_\theta(y|x) = &\frac{1}{|V|} \sum_{i \in V} w_i (x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0]) \\ &+ \frac{1}{|E_h|} \sum_{(i,j) \in E_h} t_{ij} \cdot x_i x_j \cdot I[y_i = y_j = 1] - \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \end{aligned} \tag{9}$$

$$\begin{aligned} \nabla_w \log p_\theta(y|x) &= \frac{1}{|V|} \Big[x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0] \Big]_{i \in V} - \nabla_w \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\ &\text{define } \phi_u(x,y) = \frac{1}{|V|} \Big[x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0] \Big]_{i \in V} \\ &= \phi_u(x,y) - \sum_{\hat{y}} p_\theta(\hat{y}|x) \cdot \phi_u(x,\hat{y}) \end{aligned} \tag{10}$$

$$
\begin{aligned}
\nabla_w \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) &= \frac{1}{\sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x)} \nabla_w \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\
&= \frac{1}{Z(x)} \sum_{\hat{y}} \nabla_w \exp\left\{ \frac{1}{|V|} \sum_{i \in V} w_i \big(x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0] \big) \right\} \\
&\quad \cdot \exp\left\{ \frac{1}{|E_h|} \sum_{(i,j) \in E_h} t_{ij} \cdot x_i x_j \cdot I[y_i = y_j = 1] \right\} \\
&= \frac{1}{Z(x)} \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \cdot \frac{1}{|V|} \Big[x_i \cdot I[\hat{y}_i = 1] + (1 - x_i) \cdot I[\hat{y}_i = 0] \Big]_{i \in V} \\
&= \sum_{\hat{y}} p_\theta(\hat{y}|x) \cdot \phi_u(x, \hat{y}) \qquad (11)
\end{aligned}
$$

$$
\begin{aligned}
\nabla_t \log p_\theta(y|x) &= \underbrace{\frac{1}{|E_h|} \Big[t_{ij} \cdot x_i x_j \cdot I[y_i = y_j = 1] \Big]_{(i,j) \in E_h}}_{\phi_t(x,y)} - \nabla_t \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\
&= \phi_t(x, y) - \sum_{\hat{y}} p_\theta(\hat{y}|x) \cdot \phi_t(x, \hat{y}) \qquad (12)
\end{aligned}
$$

$$
\begin{aligned}
\nabla_t \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) &= \frac{1}{\sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x)} \nabla_t \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\
&= \frac{1}{Z(x)} \sum_{\hat{y}} \nabla_t \exp\left\{ \frac{1}{|V|} \sum_{i \in V} w_i \big(x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0] \big) \right\} \\
&\quad \cdot \exp\left\{ \frac{1}{|E_h|} \sum_{(i,j) \in E_h} t_{ij} \cdot x_i x_j \cdot I[y_i = y_j = 1] \right\} \\
&= \frac{1}{Z(x)} \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \cdot \frac{1}{|E_h|} \Big[t_{ij} \cdot x_i x_j \cdot I[\hat{y}_i = \hat{y}_j = 1] \Big]_{(i,j) \in E_h} \\
&= \sum_{\hat{y}} p_\theta(\hat{y}|x) \cdot \phi_t(x, \hat{y}) \qquad (13)
\end{aligned}
$$

The derivation agrees with the property that the gradient of CRF potential is the expectation of a feature function. Note that since weights represent the credibility of unary confidence, all-entries of θ are non-negative.

3.2 Experiments

The summary of accuracy is shown in table 4.

Stage 1: enhanced numerical stability with low confidence???

State 2: Original relationships are weak. A promising direction is to emphasize on more relationships

Stage 3: It has been noticed that some weights are set to zero by optimisation.

	0%	50%	90%
Baseline	0.7686	0.3997	0.0148
Independent	0.1716(0.7339)	0.0047(0.6258)	0.0000(0.1728)
Original HEX	0.6573(0.7779)	0.3182(0.7193)	0.0000(0.4774)
Stage 1	0.6579(0.7790)	0.3182(0.7197)	0.0000(0.5154)
Stage 2	0.6923(0.7969)	0.4631(0.7470)	0.0005(0.5908)
Stage 3	0.6882(0.7802)	0.4643(0.6496)	0.1454(0.3646)

Table 4: complete(top1/top3). Signs of overtrainning of CRF on the training set, but the benefit soon takes over as relabelling rate grows.

Scheme	0%	50%	90%
1	TODO	TODO	TODO
2	0.7268(0.8889)	0.6739(0.8705)	0.3230(0.5385)
3	0.7209(0.8788)	0.6852(0.8574)	0.5029(0.7838)
4	0.7214(0.8622)	0.6799(0.8337)	0.5000(0.7850)
5	0.7238(0.7957)	0.6252(0.7470)	0.3129(0.5908)

Table 5: limited(top1/top3). Scheme 6 only apply to the extended concept space.

4 Discussions

4.1 Scalability

Scalable until stage 2 with MAP LBP. Stage 3 is not scalable due to the brute force calculation of partition function. [download large hex graph from pHEX paper, see state space size]

4.2 Relationship to Probabilistic HEX

Afte the ECCV 14 paper, the original authors extended HEX to pHEX by relaxing 0/1 hard constraints to $u \in (0, 1)$, and turning the HEX graph into an Ising model [2]. pHEX managed to beat HEX by [TODO]. However, the crucial u is chosen by cross-validation, and it still applies to all constraints in the graph. In addition, pHEX require a long time to train, although still much shorter than training the underlying CNN. Also, it uses the same potential function as original HEX. Compared to pHEX, this work focuses on using a more flexible potential function. In term of running time, these two works are not directly comparable since this work is not scalable.

A Original HEX Fails with Positive Unary Inputs

Denote the set of active nodes in a state by $\mathcal{S}$. Then there are three cases:

1. Assume there exists exactly one node t, such that $\forall s \in \mathcal{S} : s \to t$. In other words, t is the only node among all active ones with no out-edges. If t has children, then labelling any of them as true improves (1).

2. Assume there exists two active nodes with no out-edges. Denote them by t_1 and t_2. Then they must share a common descendant, as otherwise they would be connected by an exclusive edge.

   (a) If they share a common immediate child c, then labelling c as true improves (1).

   (b) If they only share a distant descendant, then the state of t_1 and t_2 are independent. The problem is thus equivalent to two case 1 in parallel.

3. There exists more than two active nodes with no out-edges. then each pair of them must have a common descendant, and the case is equivalent to pairwise case 2.

Hence, it is guaranteed that for positive unary inputs, the original HEX labels to a node in the original concept space.

B Attributed HEX

In earlier stages of this project, one of the proposed directions was the joint analysis of concept and attributes. That is, for example, to classify an image of a yellow Labrador into "animal, pet, dog, yellow, furry", etc. An example HEX graph is shown as follows:

While the training data is provided by aPascal & aYahoo dataset [4]. [TODO: discuss dataset properties] The possibility of using CNN as feature extractor has been confirmed in [7].

We only connected attributes to bottom-level concepts. In other words, the aHEX graph can be seen as a semantic part and an attribute part. Such design is for the sake of reducing loops in the aHEX graph. While junction-tree algorithm could potentially handle such a loopy graph during inference stage, the loops makes the graph very tricky to learn as a CRF. Similar design has been applied to pixel labelling problem such as [TODO: find references].

However, it was not the loops that caused such idea to be dropped. Under the same non-learning model, potential function is submodular, therefore such extension s trivial. Under the learnable CRF model, the aHEX graph can be learned part-wise: the semantic subgraph can be learned in the same way as discussed in [TODO: cross-ref learnable CRF], whereas for the attributes part, all bottom-level concepts can be grouped into a super-concept with 20 states (corresponding to 20 PASCAL labels), then the attribute graph becomes a tree, which is learnable[TODO: find references]. With such model, the inference is the same as an non-learning system, as the potential function for the attributed part still has a submodular structure.

References

[1] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Computer Vision–ECCV 2014*, pages 48–64. Springer, 2014.

[2] Nan Ding, Jia Deng, Kevin Murphy, and Hartmut Neven. Probabilistic label relation graphs with ising models. *arXiv preprint arXiv:1503.01428*, 2015.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[4] Alireza Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[5] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.