Higher-Order Inference for Vision Problems

Chris Russell

Abstract

Many problems of image understanding can be formulated as semantic segmentation, or the assignment of a 'class' label to every pixel in the image. Until recently, for reasons of efficiency, the problem of generating a good labelling of an image has been formulated as the minimisation of a pairwise Markov random field. However, these pairwise fields are unable to capture the higher-order statistics of natural images which can be used to enforce the coherence of regions in the image or to encourage particular regions to belong to a certain class.

Despite these limitations, the use of pairwise Markov models is prevalent in vision. This can largely be attributed to the pragmatism of computer vision researchers; although such models do not fully capture image statistics, they service as an effective discriminative model that prevents individual pixels from being mislabelled. Moreover, unlike many optimisation approaches for higher-order models, approximation algorithms exist for pairwise models, that are guaranteed to find a solution whose cost must lie within a fixed bound of the cost of global optima.

In this thesis, we show that the optimisation of many higher-order models can also be performed by approximate algorithms which have the same guarantees and effectiveness as those used for the optimisation of pairwise algorithms. We first consider the optimisation of the higher-order Associative Hierarchical Networks, and by transforming them into pairwise models, propose new approximate algorithms for efficiently minimising them. This work is the first to prove approximation bounds, independent of the size of cliques, for the widespread P^n and robust P^n models. We consider the problem of optimising the set of labels present in an image and the labelling of the image concurrently, and show how they can be optimised simultaneously using a variety of techniques. In the final chapter, we move beyond this, and try to address the question, “Which higher-order functions can be efficiently minimised with graph-cuts?” Although this question is not yet amenable to an algebraic answer, we propose novel linear programming based techniques for exploring the space of solutions.

Contents

Publications

ICCV 2011	Automated Articulated Structure and 3D Shape Recovery from Point Correspondences *Fayad, Russell, Agapito*
BMVC 2011	Efficient Second Order Multi-Target Tracking with Exclusion Constraints *Russell, Setti, Agapito*
CVPR 2011	Energy based Multiple Model fitting for NRSfM *Russell, Fayad, Agapito*
Best Paper BMVC 2010	Joint Optimisation for Object Class Segmentation and Dense Stereo Reconstruction *Ladicky, Sturgess, Russell, Sengupta, Bastanlar, Clocksin, Torr*
Best Paper ECCV 2010	Graph Cut based Inference with Co-occurrence Statistics *Ladicky, Russell, Kohli, Torr*
ECCV 2010	What, Where and How Many? Combining Object Detectors and CRFs *Ladicky, Russell, Sturgess, Alahari, Torr*
UAI 2010	Exact and Approximate Inference in Associative Hierarchical Networks using Graph Cuts *Russell, Ladicky, Kohli, Torr*
CVPR 2010	Efficient Piecewise Learning for Conditional Random Fields *Alahari, Russell, Torr*
ICCV 2009	Associative Hierarchical CRFs for Object Class Image Segmentation *Ladicky, Russell, Kohli, Torr*
ICCV/MMBIA 2007	Using the P^n Potts Model with Learning Methods To Segment Live Cell Images *Russell, Restif, Metaxas, Torr*

Contributions of the Thesis

I have been privileged in the people I have been able to work with over the course of my PhD. In particular, my supervisor Philip Torr has done a great job in filling his group with PhD students who are able to work well together, and have done exciting work. Unfortunately, this presents some challenges for me in claiming the contribution of my thesis, as all of my work has been in collaboration with others.

As such material in the first two chapters should be seen as motivational rather than as a theoretical contribution of my thesis. In particular, chapter 2 describes the papers Ladicky et al. (2009) and Ladicky et al. (2010b) – these are not to be seen as a contribution of my thesis but as motivating the need for the efficient methods of inference developed in chapter 3.

The work in chapters 3 and 4 was done in close collaboration with Lubor Ladicky, and it is difficult to say who did exactly what. To the best of my recollection, Lubor initially proposed a nested P^n style graph-cut, but was unsure what cost function it optimised. In response to this, I proposed the pairwise formulation of the model, which allowed us to further define unary and pairwise potentials over super-pixels, and showed how standard α-expansion could be performed on it - leading to the inference bounds reported in this thesis. Of the co-occurrence potentials, again, Lubor first proposed the graph construct (also independently discovered by Hoiem et al. (2007) and Delong et al. (2010)) while I characterised the space of functions *i.e.* those that are *monotonically increasing* that can be solved by them, and proposed alternate methods of inference.

Chapter 5 is unpublished work with Srikumar Ramalingam, and Lubor Ladicky. The general formulation, up to and including Theorem 1, which allows any higher-order function that can be solved with graph-cuts to be solved, is entirely my own work. The remainder of the chapter, which is concerned with the compact representation of 4 clique potentials is joint work that can not be easily divided.

Acknowledgements

Firstly, I would like to thank to my examiners Andrew Fitzgibbon and Fabio Cuzzolin who's suggestions have substantially improved the clarity of my thesis. All of the mistakes remaining are my own.

As said, I have been extraordinarily lucky in the people I've been able to work with over the course of my PhD. I'm grateful to my PhD seniors: Pawan Kumar, Pushmeet Kohli, and Carl Ek for providing stimulating conversation both in the lab and the pub. My co-authors: Christophe Restif, Lubor Ladicky, Pushmeet (again), Sri Ramalingam, Karteek Alahari, Paul Sturgess, Sunando Sengupta, Yalin Bastanlar, Joao Fayad, and Francesco Setti are all people I'm glad to be able to call friends first, and colleagues second. Others in the lab, including Jon Rihan, David Jarzebowski, Sam Hare, Glenn Sheasby, Natasha Govender, Sunando Sengupta and Ziming Zhang have been a pleasure to work with and around.

Phil Torr has a done a fantastic job as a supervisor, and I am grateful to him for giving me the time to grow as a researcher. I'm similarly grateful to my current supervisor Lourdes Agapito for giving me the time to complete my thesis.

And finally, I'd like to thank my wife Di, who is the reason I chose to stay in Oxford.

Chapter 1

Introduction

> The purpose of computing is insight, not numbers.
>
> **RW Hamming (1971)***Introduction to Applied Numerical Analysis*

1.1 Semantic Segmentation

Semantic Segmentation refers to the process of automatically providing a dense annotation in which every pixel within an image is labelled with one of a predetermined set of classes. This labelling should match human annotations. Note that the concept of a person's annotation is inherently ill defined: for instance the correct label of a particular pixel in an image could be any one of **car**, **Toyota**, **1.5 m from the camera lens**, or any combination of the three, depending on the choice of domain (see figure 1.1 for an example).

Typically, the problem of finding a labelling close to human based labellings is formulated as the minimisation of a cost function. This function is either hand-crafted by researchers, or more typically *learnt* from human annotations of different data. Such learning methods include probabilistic approaches, which can be loosely categorised as generative (Besag, 1986), or conditional (Lafferty et al., 2001) which can be learnt using pseudo-likelihood (Besag, 1975; Sutton and

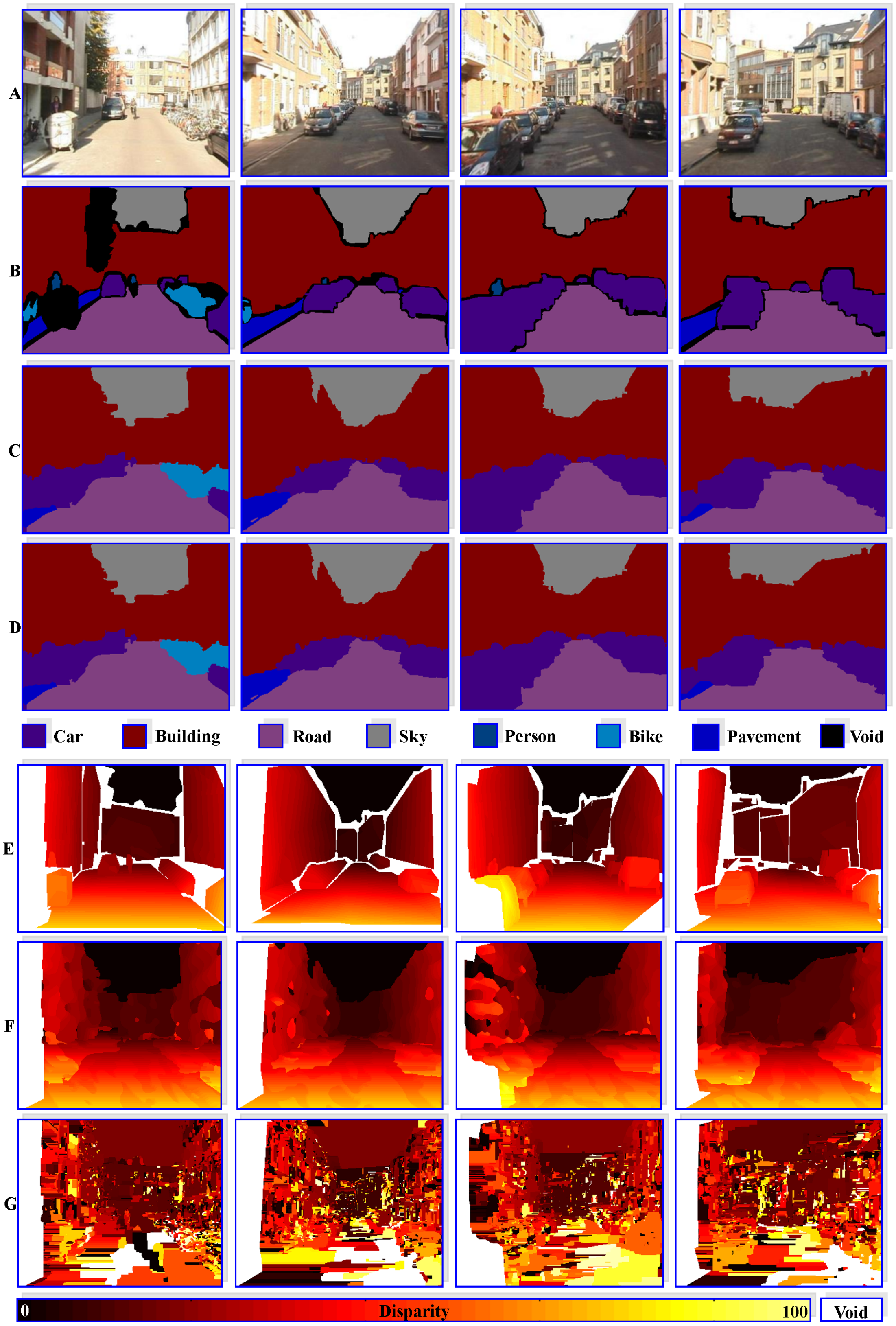

Figure 1.1: *Example annotations of a road scene.* **A** *Natural images,* **B, E** *human annotations of object class and depth respectively.* **D,G** *shows* AHN *and* CRF *based labellings of object class and depth.* **C, F** *show a joint labelling of object class and depth via an* AHN*. See Ladicky et al. (2010d) for more details.*

McCallum, 2007), or as discriminative approaches such as (Taskar et al., 2003; Alahari et al., 2010). In this thesis, we assume that the problem of formulating the cost function has already been solved, and we are interested in finding a minimum cost labelling of this function.

The first half of this chapter discusses prior works on semantic segmentation. Readers familiar with Markov Random Fields, Conditional Random Fields and the P^n Model, may wish to gloss over it. In section 2 we introduce the Associative Hierarchical Networks, and show their application to semantic segmentation. Section 2.3 shows how they can be integrated with detectors for improved accuracy.

1.2 Existing Models

We begin by introducing the mathematical models used by standard approaches. Note that while much of the notation, and early methods used in semantic segmentation were probabilistic, we choose to take a discriminative approach, and phrase the problem of semantic segmentation as the minimisation of some arbitrary cost function.

Assume that every pixel in the image takes a label from a set $\mathcal{L} = \{l_1, \ldots, l_k\}$. Let $\mathbf{X} = \mathbf{X_1} \times \mathbf{X_2} \times \ldots \times \mathbf{X_n} = \mathcal{L}^\mathbf{n}$ be the set of possible variables assignments representing all semantic labellings of the pixels in the image $\{1, \ldots, n\}$. We use $\mathbf{x} = \{x_1, \ldots, x_n\} \in \mathcal{L}^n$ to refer to a *labelling* of $\mathbf{X}$. We will use $C(\mathbf{x})$, where

$$C(\cdot) : \mathbf{X} \to \mathbb{R} \tag{1.1}$$

to refer to the cost of a labelling $\mathbf{x}$.

Given a predefined function C and label space $\mathbf{X}$, we define the problem of inference as finding a minimal cost labelling $\mathbf{x}^*$ such that:

$$C(\mathbf{x}^*) = \min_{\mathbf{x}\in\mathbf{X}} C(\mathbf{x}). \tag{1.2}$$

1.2.1 Independent model

Two components, or subsets of variables, $\mathbf{U}, \mathbf{V} \subseteq \mathbf{X}$ such that $\mathbf{U} \cup \mathbf{V} = \mathbf{X}$ and $\mathbf{U} \cap \mathbf{V} = \emptyset$ are said to be independent with respect to C if there exists two functions $C_1(\cdot) : \mathbf{U} \to \mathbb{R}$, $C_2(\cdot) : \mathbf{V} \to \mathbb{R}$ such that:

$$C(\mathbf{u}, \mathbf{v}) = C_1(\mathbf{u}) + C_2(\mathbf{v}) \ \ \forall \mathbf{u} \in \mathbf{U}, \mathbf{v} \in \mathbf{V} \tag{1.3}$$

We say that a model is *independent* if for all possible choices of $\mathbf{U}$, $\mathbf{V}$ there exists two appropriate functions $C_1(\cdot)$, $C_2(\cdot)$ such that condition (1.3) holds. For an independent model, the cost $C(\mathbf{x})$ can be written in the form:

$$C(\mathbf{x}) = \sum_{i=1}^{n} \psi_i(x_i) \tag{1.4}$$

This implies that the cost associated with the labelling of any variable X_i is independent of the state of any other $\mathbf{X_j}$. This allows us to use a piecewise greedy labelling strategy to find the optimal solution.

The nature of ψ_i

Although independent models exhibit no dependence between the labels of different pixels, the labelling of any one pixel must still depend on the appearance of surrounding pixels if they are to have any hope of correctly classifying pixels (see figure 1.2 for an illustration of this). In practice pixel-based classifiers, which give rise to the cost ψ_i rely upon an aggregate of statistical features taken from the region surrounding a particular pixel. For example: TextonBoost (Shotton et al.,

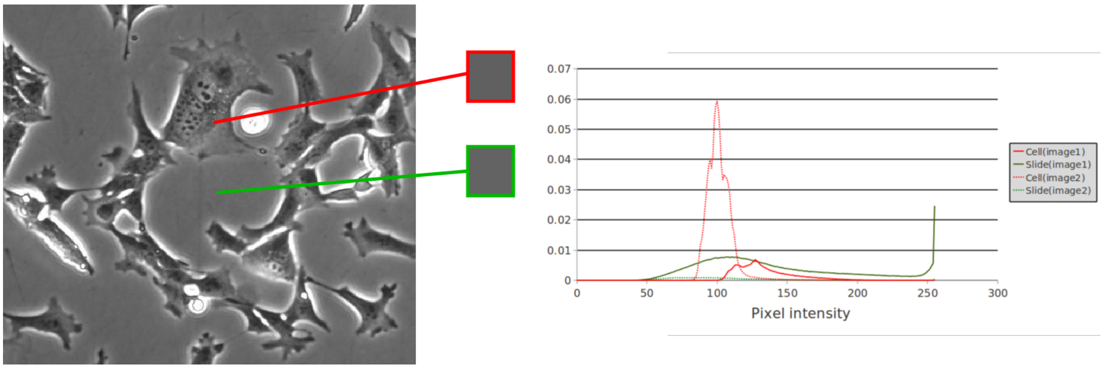

Figure 1.2: **Recognising individual pixels without context.** *An illustration of some issues in labelling individual pixels without knowledge of the surrounding image.* **(Left)** *a slide containing cells.* **(Centre)** *Pixels extracted from cells and non-cell* **(Right)** *The distribution of grey-scale values of both cells and non-cells, over two different images. See Russell et al. (2007), and section 1.2.1 for further details.*

2006) performs boosting based on weak features which describe the number of a particular type of texton lying in a rectangle near the pixel; the example of figure 1.2 can be easily classified using the variance of a small patch about each pixel (see Russell et al. (2007)); Kumar and Hebert (2006) used a variety of statistical moments defined over a rectangular block to describe regions. For challenging problems, the use of a single set of features over a region is not enough. The unary potentials of Ladicky et al. (2009) relied upon a combination of textons, signed HOG (Dalal and Triggs, 2005), colour HOG (Villamizar et al., 2009), location and local colour to achieve state the art performance on the MSRC data set (Shotton et al., 2006) at the time of writing.

1.2.2 Conditional Random Fields and Markov Random Fields

We can generalise the form of an independent model by allowing smoothing terms between pairs of variables. The cost $C(\mathbf{x})$ now takes the form:

$$C(\mathbf{x}) = \sum_{i=1}^{n} \psi_i(x_i) + \sum_{(i,j)\in\mathcal{N}} \psi_{i,j}(x_i, x_j) + C \tag{1.5}$$

where $\mathcal{N}$ is the *neighbourhood* structure which describes which *ordered* pairs of variables share cost dependencies, and C is a *reparameterisation term* used to hide constant terms that do not affect the location of minima. Where possible the constant term C will be omitted from future equations for clarity.

We will also use the word *reparameterisation* to refer to a different decomposition of the cost $C(\cdot)$. For example, let $\mathcal{N}_i = \{j|\,(i,j) \in \mathcal{N} \vee (j,i) \in \mathcal{N}\}$ *i.e.* the set of all neighbours of i, then letting:

$$\psi'_{i,j}(x_i, x_j) = \frac{1}{|\mathcal{N}_i|}\psi_i(x_i) + \psi_{i,j}(x_i, x_j) + \frac{1}{|\mathcal{N}_j|}\psi_j(x_j), \tag{1.6}$$

we have

$$\begin{aligned}
C(\mathbf{x}) &= \sum_{i=1}^{n} \psi_i(x_i) + \sum_{(i,j)\in\mathcal{N}} \psi_{i,j}(x_i, x_j) + C & (1.7)\\
&= \sum_{i\in\mathcal{V}} \sum_{(i,j)\in\mathcal{N}_i} \left(\frac{1}{|\mathcal{N}_i|}\psi_i(x_i) + \psi_{i,j}(x_i, x_j) + \frac{1}{|\mathcal{N}_j|}\psi_j(x_j)\right) + C & (1.8)\\
&= \sum_{(i,j)\in\mathcal{N}} \left(\frac{1}{|\mathcal{N}_i|}\psi_i(x_i) + \psi_{i,j}(x_i, x_j) + \frac{1}{|\mathcal{N}_j|}\psi_j(x_j)\right) + C & (1.9)\\
&= \sum_{(i,j)\in\mathcal{N}} \psi'_{i,j}(x_i, x_j) + C & (1.10)
\end{aligned}$$

and (1.10) is a reparameterisation of (1.5).

We refer to sub-costs of the form $\psi_i(x_i)$ as *unary potentials* and those of the form $\psi_{i,j}(x_i, x_j)$ as *pairwise potentials.*

If the pairwise potentials $\psi_{i,j}(x_i, x_j)$ are defined without inspection of the image, the model may be referred to as a *Markov Random Field* (MRF). If $\psi_{i,j}(x_i, x_j)$ changes with the appearance of pixels it is referred to as a *Conditional Random Field* (CRF) (Lafferty et al., 2001). From a discriminative perspective, and as a problem of optimisation, there is no difference between MRFs and CRFs, except that expressibility of CRFs strictly dominates that of MRFs and typically offer better performance. The probabilistic perspective on the difference between MRFs and CRFs is more involved and we refer the interested reader to (Lafferty et al., 2001). These pairwise potential encode a smoothness prior which encourages neighbouring pixels in the image to take the same label, resulting in a *shrinkage bias* (Kohli et al., 2008).

The pairwise CRF formulation suffers from a number of problems stemming from its inability to express high-level dependencies between pixels [1]. Despite these limitations, pairwise models are widely used and highly effective.

The presence of pairwise inter-dependencies make the problem of finding a minimal cost solution NP-hard in the general case (Dahlhaus et al., 1994), however several notable exceptions exist.

Binary submodular costs If the label space is binary, *i.e.* $\mathcal{L} = \{0, 1\}$, and all pairwise potentials $\psi_{i,j}$ satisfy the inequality

$$\psi_{i,j}(0,1) + \psi_{i,j}(1,0) \geq \psi_{i,j}(0,0) + \psi_{i,j}(1,1), \tag{1.11}$$

then the cost function $C(\mathbf{x})$ is said to be *pairwise submodular.*

Uniquely, out of all pairwise definitions in this chapter, the property of sub-

[1] For example, such dependencies may express the belief that a set of pixels should belong to the same class (Kohli et al., 2009), or to one particular class (Ladicky et al., 2009), or to a particular ordered range of classes (Woodford et al., 2008).

modularity is invariant to reparameterisation. If we have two parameterisations of the function $C(\cdot)$

$$C(\mathbf{x}) = \sum_{i=1}^{n} \psi_i(x_i) + \sum_{(i,j)\in\mathcal{N}} \psi_{i,j}(x_i, x_j) + C = \sum_{i=1}^{n} \psi'_i(x_i) + \sum_{(i,j)\in\mathcal{N}} \psi'_{i,j}(x_i, x_j) + C'. \tag{1.12}$$

Then all $\psi_{i,j}$ are submodular if and only if all $\psi'_{i,j}$ are.

A global minimum of any pairwise submodular energy can be efficiently found using a graph-cuts algorithm (see section 1.3.1).

Convex costs Convex costs (Ishikawa, 2003; Schlesinger, 2007) can be seen as a generalisation of pairwise submodular costs to a larger label space. We say that a function f is convex over a domain $[0, k-1]$ if it satisfies the following constraint:

$$\begin{gathered} f(ty_2 + (1-t)y_2) \leq t\, f(y_2) + (1-t)\, f(y_2) \\ \forall y_1, y_2 \in [0, k-1], t \in [0, 1]. \end{gathered} \tag{1.13}$$

Given an ordered set of labels $\mathcal{L} = \{0, \ldots, k-1\}$ we say that the set of pairwise potentials $\psi_{i,j}(x_i, x_j)$ is convex if and only if $\forall i, j$ there exists some convex function f such that $\psi_{i,j}(x_i, x_j) = f(|x_i - x_j|)$ [2]. As with pairwise submodular costs, a minimal labelling can be efficiently found using graph-cuts. In the degenerate case in which $\mathcal{L} = \{0, 1\}$ a cost can be reparameterised to take a convex form if and only if it is submodular.

Metrics A pairwise energy defined over a label space $\mathcal{L}$ is said to be metric (Boykov et al., 2001) if it satisfies the following three properties:

[2] Note that this definition of convexity differs from that used when talking about higher order functions as in section 1.2.3, or figure 1.3. These higher order costs are defined as convex functions over *the total number of variables taking a particular label* in a region or clique, and are not dependent on choice of ordering.

1. Positive definiteness

$$\psi_{i,j}(a,b) = 0 \iff a = b \;\forall a,b \in \mathcal{L}. \tag{1.14}$$

$$\psi_{i,j}(a,b) \geq 0 \;\forall a,b \in \mathcal{L}. \tag{1.15}$$

2. Symmetry

$$\psi_{i,j}(a,b) = \psi_{j,i}(a,b) \;\;\forall a,b \in \mathcal{L}. \tag{1.16}$$

3. Triangle inequality

$$\psi_{i,j}(a,c) \leq \psi_{i,j}(a,b) + \psi_{i,j}(b,c) \;\forall a,b,c \in \mathcal{L}. \tag{1.17}$$

The most prevalent metric in semantic segmentation is the *generalised Potts potential* which takes the form:

$$\psi_{i,j}(x_i,x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ w_{i,j} & \text{otherwise.} \end{cases} \tag{1.18}$$

This cost is a statement that although we prefer pairs of neighbouring variables to take the same class, we are indifferent as to how this preference may be violated.

Finding the global minimum of an arbitrary metric is a significantly harder problem then minimising a convex energy. It has been shown that minimisation of a generalised Potts model containing at least 3 labels is NP hard (Dahlhaus et al., 1994). Nonetheless, approximate inference is possible. In particular, if the terms $\psi_i(x_i)$ and C are always non-negative, the graph cut based algorithm *α-expansion* (see section 3.4.1) is guaranteed to find a solution who's cost lies within a factor of 2 of the globally minimum cost. We will refer to such factors as *bounds* later within the text. For any arbitrary metric, α-expansion is guaranteed to find a solution whose cost lies within a bound of $2\frac{\max_{x_i \neq x_j} \psi_{i,j}(x_i,x_j)}{\min_{x_i \neq x_j} \psi_{i,j}(x_i,x_j)}$ of the cost of the global solution.

Semi-Metrics Semi-metrics (Boykov et al., 2001) or pairwise associative energies (Taskar et al., 2004) are a generalisation of metrics that remove the requirement that the triangle inequality holds (1.17). While specialist algorithms such as $\alpha\beta$ swap (Boykov et al., 2001) exist that are designed to optimise these energies, bounds and better results can be obtained by locally approximating them as a metric, and using standard metric (Boykov et al., 2001) or tree metric (Kumar and Koller, 2009) optimisation techniques to approximate them.

Truncated convex potentials Truncated convex potentials (Kumar and Torr, 2008b; Veksler, 2007) are an important class of semi-metrics, defined over an ordered set of labels. They take the form:

$$\psi_{i,j}(x_i, x_j) = w_{i,j} \min(f(|x_i - x_j|), k) \tag{1.19}$$

where f is a convex function.

Such potentials can be understood as lying halfway between convex energies (1.13) and the generalised Potts model (1.18). This insight allows the formulation of efficient optimisation algorithms that outperform general metric solving techniques, such as α-expansion, by hybridising the techniques used to solve convex costs with α-expansion. These techniques will become important in chapter 3, where we argue that the higher order potentials of section 1.2.3[3], can be understood as truncated convex potentials defined over an *unordered* range.

[3] See also figure 1.3 for an illustration of these costs, and chapter 2 for a generalisation of them.

1.2.3 Associative Higher Order Potentials and the $\mathbf{P}^n$ Model

Unary and pairwise costs can be generalised to potentials defined over a set of variables (henceforth a *clique*) of arbitrary size. Given a subset of variables $c \subseteq \mathbf{X}$ we write $\mathbf{x}_c$ to refer to the state of the variables in c. We use the notation

$$\psi_c(\mathbf{x}_c) \tag{1.20}$$

to refer to the cost of the potential defined over clique c. Potentials defined over a clique of size greater than 2 will be referred to as *higher order potentials.* We write the global cost function as

$$C(\mathbf{x}) = \sum_{i=1}^{n} \psi_i(x_i) + \sum_{(i,j)\in\mathcal{N}} \psi_{i,j}(x_i, x_j) + \sum_{\substack{c\in\mathcal{C}\\ |c|>2}} \psi_c(\mathbf{x}_c) + C, \tag{1.21}$$

where $\mathcal{C}$ is the set of all cliques, or equivalently as

$$C(\mathbf{x}) = \sum_{c\in\mathcal{C}} \psi_c(\mathbf{x}_c), \tag{1.22}$$

when the distinction between constant, unary, pairwise, and higher order terms is irrelevant.

Associative Higher order Potentials

In Taskar (2004), the author defined a *higher order associative potential* over a clique c as:

$$\psi_c(\mathbf{x}_c) = \sum_{l\in\mathcal{L}} -k_{c,l} \prod_{i\in c} \Delta(x_i = l) \tag{1.23}$$

where $k_{c,l} \geq 0$, and Δ is the Dirac function i.e.

$$\Delta(\cdot) = \begin{cases} 1 & \text{if } \cdot \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{1.24}$$

We can explicitly write these potentials as:

$$\psi_c(\mathbf{x}_c) = \begin{cases} -k_{c,l} & \text{if } x_i = l, \forall i \in c \\ 0 & \text{otherwise.} \end{cases} \tag{1.25}$$

Independently, Kohli et al. (2007) defined a P^n potential as:

$$\psi'_c(\mathbf{x}_c) = \begin{cases} \gamma_{c,l} & \text{if } \forall i \in c : x_i = l \\ \gamma_{c,\max} & \text{otherwise} \end{cases} \tag{1.26}$$

$$\text{where } \gamma_{c,\max} > \gamma_{c,l} > 0 \;\; \forall l. \tag{1.27}$$

Note that, for all choices of $k_{c,l}$ there exists a corresponding set of $\gamma_{c,l}$ and $\gamma_{c,\max}$ such that:

$$\psi_c(\mathbf{x}_c) = \psi'_c(\mathbf{x}_c) - \gamma_{c,\max}, \tag{1.28}$$

and visa versa. Consequently, the two models are equivalent under reparameterisation.

The Robust P^n Model

These models have been generalised to the robust P^n model (Kohli et al., 2009) which takes the form:

$$\psi_c(\mathbf{x}_c) = \min\left(\gamma_{c,\max}, \min_{l \in \mathcal{L}} \left(\gamma_{c,l} + \sum_{i \in c} k_{c,i} \Delta(x_i \neq l)\right)\right), \tag{1.29}$$

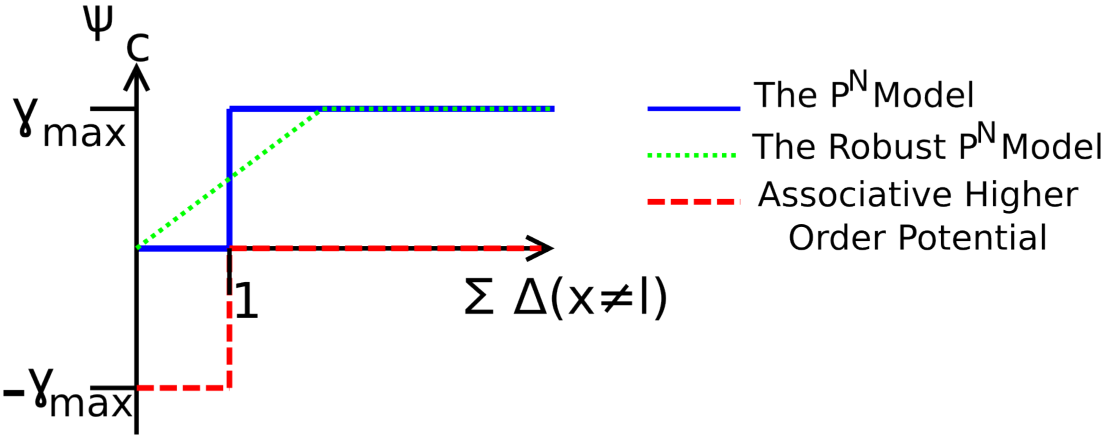

Figure 1.3: The above figure shows standard graphical representations of the three higher-order potentials discussed in the text. While the P^n model, and Taskar's Associative higher order potential may appear to be non-convex (see equation (1.13)) with respect to $\sum_{i \in c} \Delta(x_i \neq l)$, this is an artifact of the discrete state space, and the potentials can be redrawn as convex in a similar manner to the Robust P^n model.

where the γ terms follow the constraints of (1.27) and $k_{c,i} \geq 0 \ \forall c, i$. In the degenerate case in which we set the terms $k_{c,i} = \gamma_{c,\max}$ it can readily be seen that the Robust P^n model becomes equivalent to the P^n model and consequently the family of Robust P^n potentials strictly contains all P^n potentials and Taskar's associative higher order potentials.

These robust potentials can be understood as a truncated majority voting scheme on the base layer. Where possible, they encourage the entirety of the clique to assume a homogeneous labelling. However, beyond a certain threshold of disagreement they implicitly recognise that no consistent labelling is likely to occur, and no further penalty is paid for increasing heterogeneity. This family of potentials have been successfully applied to diverse problems such as object class recognition (Kohli et al., 2009), document classification (Taskar et al., 2004) and texture based video segmentation (Kohli et al., 2007), where they obtained state of the art results.

1.3 Inference

1.3.1 Graph-Cuts

The family of graph-cut algorithms (an excellent review of them, and their application to vision problems is given in Boykov and Kolmogorov (2004)) are highly efficient solvers which compute the minimum cut required to separate two predefined vertices (these vertices are referred to in the literature as the *source* and the *sink*) on a directed graph containing no negative edge-weights. The majority of these algorithms find the minimum cut by maximising a *dual* formulation. These dual formulations exploit the fact that the cost of a minimum cut between source and sink is equal to the maximum flow that can be pushed through the graph from the source to the sink[4], and that a minimal cut can be extracted by discarding edges that are saturated in the max-flow solution, *i.e.*, they have had flow exactly equal to their capacity pushed through them. However, for the purposes of this thesis, graph-cuts will be treated as a black box algorithm.

As previously described, a global minimum of all pairwise submodular costs (1.11) can be found using graph cuts, a detailed explanation for this can be found in Kolmogorov and Zabih (2004). A brief sketch of the proof can be given as follows.

We wish to minimise a pairwise function $C(\cdot)$ defined over $\mathbf{X} = \{0,1\}^n$. To do this we define a graph $G = \langle V, E \rangle$, as a set of $|\mathbf{X}| + 2$ vertices V, and a set of directed edges E between the vertices. The set V contains one vertex V_i for every variable $X_i \in \mathbf{X}$ and two additional vertices, which we refer to as the *source* S, and the *sink* T. We define a *cut* as a partitioning of the vertices of the graph into two sets, one of which must containing the source S, and the other contains the sink T. To create an equivalence between partitioning the vertices, and finding a

[4] If the maximum amount of flow that can be pushed through the edge is bounded by the cost of breaking that edge.

labelling of $\mathbf{X}$, we associate a vertex V_i belonging to the same set as S with the a labelling $x_i = 0$, while if V_i belongs to the same set as T, it is equivalent to the labelling $x_i = 1$. We associate a weight, $w_{i,j}$ with every edge $e = (i, j) \in E$, and define the cost of a cut as

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{T}} \Delta((i,j) \in E) w_{i,j} \tag{1.30}$$

where $\mathcal{S}$ and $\mathcal{T}$ are the sets containing S and T respectively. Providing all of these weights $w_{i,j}$ are non-negative, the minimal cut can be found effectively using a graph-cuts algorithm.

To show that any pairwise submodular function can be solved using graph-cuts, we assert without proof that a pairwise function $C(\cdot)$ defined over $\mathbf{X}$ is submodular if and only if it can be written in the form:

$$C(\mathbf{x}) = \sum_{i \in \mathbf{X}} a_i x_i - \sum_{i,j \in \mathbf{X}} b_{i,j} x_i x_j + C \tag{1.31}$$

where $a_i \in \mathbb{R}\ \forall i$ and $b_{i,j} \in \mathbb{R}_0^+\ \forall i, j$.

It follows from equation (1.30) that a directed edge from a vertex V_i to V_j, and of weight $w_{i,j}$ has a cost of 0, unless V_i is partitioned with the source, and V_j with the sink, in which case the edge is broken with a cost of $w_{i,j}$. This is equivalent to the pseudo-Boolean[5] cost $w(1 - x_i)x_j$. We can use this to rewrite the pairwise costs of equation (1.31) as

$$C(\mathbf{x}) = \sum_{i \in \mathbf{X}} \left(a_i - \sum_{j<i} b_{j,i} \right) x_i - \sum_{i,j \in \mathbf{X}, i<j} b_{i,j}(1 - x_i)x_j + C. \tag{1.32}$$

Clearly, the pairwise terms under this formulation are of the same form as equation (1.30), and can be optimised with graph-cuts, the only question remaining

[5] pseudo-Boolean is a technical term used to refer to any cost function that maps from $\{0, 1\}^n$ to $\mathbb{R}$.

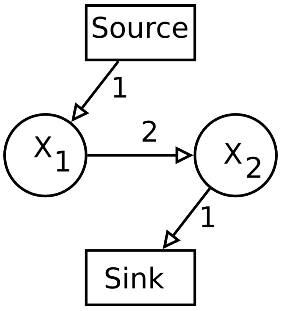

Figure 1.4: The graph-cut used to find the minimal solution of $x_1 + x_2 - 2x_1x_2$. First the energy is transformed into $x_1 + (1-2)x_2 + 2(1-x_1)x_2$, and from this form, a graph is derived. By inspection, it should be readily apparent that both problems share multiple optima, which occur whenever $x_1 = x_2$.

is how to deal with arbitrary unary potentials. Again this is straightforward; if the unary term $a'_i = a_i - \sum_{j<i} b_{j,i}$ is non-negative, this can be expressed as a directed edge from the source to the vertex X_i of cost a'_i, while if it is negative, it can be expressed as a directed edge from vertex X_i to the sink, of cost $-a'_i$, and replacing the constant term C with $C + a_i$. See figure 1.4 for an example.

1.3.2 Move making algorithms

There are two principle approaches to approximately solving the NP-hard labelling problems that frequently occur in vision. The first is one of *relaxation*, where we soften the constraints (such as each pixel must take exactly one label) that make the problem NP-hard and consider a larger space of solutions. The second approach is one of *constriction*, rather than trying to find the optimal labelling from a large set of labels $\mathcal{L}^n$, a constrained and much smaller set of labels is considered[6] and the optimal labelling of this subset is found instead. *Move-making* algorithms are an extension of *constriction*-based methods that solves a sequence of simple problems to explore a larger area of the label space.

These methods start from an arbitrary initial solution of the problem and

[6] Typically this subset is of approximate size $O(2^n)$.

$\mathbf{x}$	$\dots$	β	γ	α	γ	γ	$\dots$
Proposed Moves	$\dots$	α	α	α	α	α	$\dots$
Move choice $\mathbf{t}$	$\dots$	1	0	0	0	1	$\dots$
$\mathbf{x}'$	$\dots$	α	γ	α	γ	α	$\dots$

Figure 1.5: *An illustration of move encoding in α-expansion. Starting from an initial model $\mathbf{x}$ a new move $\mathbf{t}$ is proposed which causes two variables to change their label to α. This results in the new labelling $\mathbf{x}'$.*

proceed by solving a constrained problem which leads to a solution of the same or lower energy (Boykov et al., 2001). At each step, the algorithms formulates a constricted problem by project a set of candidate moves into a Boolean space, along with their cost function. If the resulting projected cost function (also called the *move energy*) is both submodular and pairwise, it can be exactly minimised in polynomial time by solving an equivalent st-mincut problem. These optima can then be mapped back into the original space, returning the optimal move within the move set. The move algorithms run this procedure until convergence, iteratively picking the best candidate as different choices of range are cycled through. The algorithm is said to have converged when no lower energy solution can be found.

Examples of move making algorithms include α-expansion which can only be applied to metrics, $\alpha\beta$ swap which can be applied to semi-metrics (Boykov et al., 2001), and range moves (Kumar and Torr, 2008b; Veksler, 2007) for truncated convex potentials. These moves differ in the size of the space searched for the optimal move. While expansion and swap search a space of size at most 2^n while minimising a function of n variables, the range moves explores a much larger space of K^n where K is a parameter of the energy (see Veksler (2007) for more details).

Encoding moves

The moves proposed by algorithms that function over a range of 2^n, can be can be encoded as a *transformation vector* of binary variables $\mathbf{t} = \{ t_i, \forall i \in \mathcal{V} \}$. Each component t_i of $\mathbf{t}$ encodes a partial decision, about what the state of the variable x_i change to. In the case of $\alpha\beta$-swap t_i encodes a decision whether x_i should take the label α or β. While in α-expansion, t_i encodes a decision if x_i should remain constant, or transition to a new label α. See figure 1.5 for an illustration of the use of a transformation vector in α-expansion.

The use of these transformation vectors makes the problem of finding the optimal move equivalent to minimising a pseudo-Boolean cost function. Consequently, if these cost functions can be shown to be pairwise submodular, the optimal move can be efficiently found using graph-cut, as described in section 1.3.1. Sections 1.3.3 through to 1.3.6 discuss how swap and expansion can be performed using graph-cuts on pairwise costs, and some higher-order models.

1.3.3 $\alpha\beta$ Swap

The algorithm $\alpha\beta$-swap takes $\mathbf{x}$, a current labelling of $\mathbf{X}$, and returns a new labelling $\mathbf{x}'$ which is the labelling with the lowest cost that can be reached by re-labelling some of the x_i currently taking label α or β as either β or α respectively. A local optima is then found by iteratively apply this operation through all possible choices of α, β.

Calculation of the optimal swap move for a semi-metric is pairwise submodular, and so can be efficiently solved using graph-cuts.

Proof We decompose the label space $\mathbf{X}$ into those labels currently taking label α or β — which we write as $\mathbf{X}_{\alpha\beta}$ and their complement $\bar{\mathbf{X}}_{\alpha\beta}$ of variables not currently taking labels α or β. Then

$$\begin{aligned}
C(\mathbf{x}) &= \sum_{i=1}^{n} \psi_i(x_i) + \sum_{(i,j)\in\mathcal{N}} \psi_{i,j}(x_i, x_j) && (1.33)\\
&= \sum_{X_i\in\mathbf{X}_{\alpha\beta}} \psi_i(x_i) + \sum_{\substack{(i,j)\in\mathcal{N},\\ X_i,X_j\in\mathbf{X}_{\alpha\beta}}} \psi_{i,j}(x_i, x_j) && (1.34)\\
&+ \sum_{\substack{(i,j)\in\mathcal{N},\\ X_i\in\mathbf{X}_{\alpha\beta}\\ X_j\notin\mathbf{X}_{\alpha\beta}}} \psi_{i,j}(x_i, x_j) + \sum_{\substack{(i,j)\in\mathcal{N},\\ X_i\notin\mathbf{X}_{\alpha\beta}\\ X_j\in\mathbf{X}_{\alpha\beta}}} \psi_{i,j}(x_i, x_j) && (1.35)\\
&+ \sum_{X_i\notin\mathbf{X}_{\alpha\beta}} \psi_i(x_i) + \sum_{\substack{(i,j)\in\mathcal{N},\\ X_i,X_j\notin\mathbf{X}_{\alpha\beta}}} \psi_{i,j}(x_i, x_j) && (1.36)
\end{aligned}$$

We associate a variable taking label α after the move with the i^{th} component of $\mathbf{t}$, t_i, taking label 1 and same variable taking β with $t_i = 0$.

Over the range of moves considered (1.36) is constant, (1.35) is equivalent to a unary potential, and (1.34) is a combination of unary and pairwise energies. As every pairwise cost is a semi-metric, it is positive definite, and therefore satisfies (1.11) and is submodular. □

1.3.4 α-Expansion

α-expansion is a move-making algorithm similar to $\alpha\beta$-swap, but instead of allowing all labels currently taking label α or β to change their label, it allows all labels to keep their current label or take label α. We will now show that if all pairwise costs are a metric, computation of the optimal expansion move is pairwise submodular. The proof is similar to $\alpha\beta$ swap.

Proof This time we associate $t_i = 0$ with remaining in the same state as the current labelling x_i and 1 with switching to label α. Consider two variables X_i, X_j in a pairwise cost, currently taking labels β and γ respectively. We make

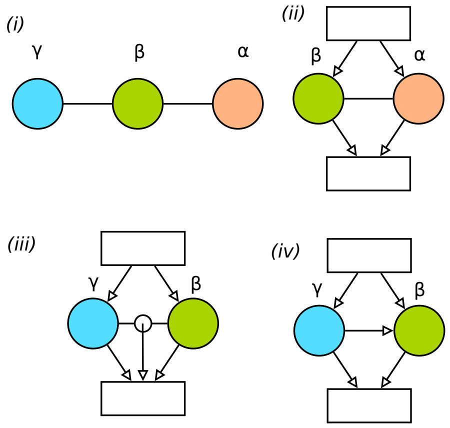

Figure 1.6: *(i)* A simple 3-label 3 variable pairwise cost, where each label is currently taken by one variable . *(ii)* Graph-cut to compute optimal $\alpha\beta$-swap. *(iii)* Graph-cut originally proposed in Boykov et al. (2001) to solve α-expansion. *(iv)* Efficient variant of the same using the techniques of Kolmogorov and Zabih (2004). Note that in *(ii)* the vertex currently taking label γ is removed, as it can not change under a $\alpha\beta$-swap, while in *(iii)* and *(iv)* the variable taking label α is removed for efficiency reasons, as it can not change label under an α-expansion.

no assumptions of uniqueness, β and γ may be equal or may equal α.

By the triangle inequality (1.17).

$$\psi_{i,j}(\beta,\gamma) \leq \psi_{i,j}(\beta,\alpha) + \psi_{i,j}(\alpha,\gamma) \tag{1.37}$$

By positive definiteness

$$\psi_{i,j}(\alpha,\alpha) = 0 \tag{1.38}$$

hence,

$$\psi_{i,j}(\alpha,\alpha) + \psi_{i,j}(\beta,\gamma) \leq \psi_{i,j}(\beta,\alpha) + \psi_{i,j}(\alpha,\gamma). \tag{1.39}$$

This is constraint (1.11), and the move is pairwise submodular. □

1.3.5 Higher Order Inference

Taskar suggested that higher order inference could be performed by relaxing the set of integer constraint $x_i \in \mathcal{L}\ \forall i$ to linear constraints:

$$x_i \in [0,1]^{\mathcal{L}}\, \forall i \tag{1.40}$$

$$\text{such that } \forall i,\ \sum_{j\in\mathcal{L}} x_{i,j} = 1 \tag{1.41}$$

where $x_{i,j}$ is the jth component of x_i. This approach has several disadvantages. Firstly the optimisation of these linear programmes (LP) must be performed via off-the-shelf optimisation packages, and is extremely slow and highly memory inefficient[7]. Secondly, while in the binary case in which $\mathcal{L} = \{0,1\}$ the LP is *tight* *i.e.* given the optimal LP solution an integer solution of the same cost (and hence also optimal) can be found, this does not hold if $|\mathcal{L}| > 2$ Taskar et al. (2004). In such cases probabilistic rounding schemes such as those proposed in Chekuri et al. (2005); Kleinberg and Tardos (1999) can be used to find solutions that are expected to lie within some approximation bound. However, these approximation bounds depend on the size of the cliques the potentials are defined over, and a direct application of these techniques results in a bound that is $O(|c|)$, where c is the largest clique (Gould et al., 2009a). As the problems we consider typically have cliques containing hundreds of thousands of variables, such bounds are meaningless. Another issue arises with qualitative performance: one important use of pairwise and higher-order potentials is to smooth the solution and to generate a solution which is not obviously wrong to the human eye. However, in the event of a fractional solution being found by the LP-solver, a rounding scheme,

[7] In chapter 4 we compare higher order LP formulations against graph-cut based methods. As an LP the memory requirements are the principal bottleneck preventing us from reasoning about images that contain more than 20×20 pixels, versus the 500×500 pixels of a standard image in the VOC data-set. Inference was 30,000 slower than a like-with-like comparison of graph-cuts. Owing to the computational complexity of LP inference, the relative difference in inference speed is likely to grow with the complexity of the problem.

used to give an integer solution, may heterogeneously label large regions of $\mathbf{x}$ with elements drawn at random from a subset of $\mathcal{L}$. Finally, the generalisation from the strict P^N proposed by Taskar, to the Robust P^N potentials is mathematically involved and incurs an additional, substantial, computational overhead.

1.3.6 Move-making algorithms for higher order energies

Kohli et al. (2007, 2009) proposed the use of move-making algorithms for the optimisation of the P^n and robust P^n model. Specifically, they demonstrated graph-cut based variants of $\alpha\beta$-swap and α-expansion for these energies, and demonstrated how inference was possible with them.

While $\alpha\beta$-swap and α-expansion appear to be inexorably linked to graph-cuts in the literature, in practice graph-cuts is only chosen due to its efficiency and speed of convergence. Any other method that can find the optimal move from the range considered can be used in its place. For example, the LP formulation of Taskar could be used to compute optimal moves, as the range considered is binary, allowing the optimal moves to be found.

We will exploit this fact in providing a proof that optimal expansion and swap moves can be computed using graph-cuts for the P^n model. We will first show that optimal swap and expansion moves can be expressed using Taskar's higher-order potentials (1.23), then we will show how these binary higher-order potentials can be expressed using graph-cuts.

Higher-order swap

We only need to show that the optimal swap moves can be found for a cost function defined over a single higher-order clique by itself. As the potentials considered are additive, if we can solve for one clique, we can solve the sum of several.

Consider a potential ψ_c defined over a clique c. We wish to perform a swap over the labels α and β. Let $c_{\alpha,\beta} \subseteq c$ be the set of all variables currently taking label α or β. Let $c_{\bar{\alpha\beta}} \subseteq c$ be their complement *i.e.* all variables in c currently taking labels other than α or β. Then there are 3 cases to consider, and we will show that each one of these can be expressed using Taskar's higher order formulation .

1. $c = c_{\bar{\alpha\beta}}$ *i.e.* c contains no variables taking labels α or β — If this is the case then the cost of the potential must remain constant, as the swap move can not change the label of any variable in the clique. This is trivially representable as one of Taskar's potentials.

2. $c_{\alpha\beta}, c_{\bar{\alpha\beta}} \neq \emptyset$ *i.e.* the current labelling of the clique is a mixture of elements from $\{\alpha, \beta\}$ and its complement — For all possible moves the cost of this potential must be $\gamma_{c,\max}$, as no homogeneous labelling is possible.

3. $c_{\alpha,\beta} = c$ *i.e.* the clique only contains labels α and β — This is a standard P^N potential (1.27) defined over the clique, as discussed in section 1.2.3, it can be reparameterised into the same form as Taskar's.

Higher-order expansion

Proof that optimal expansion moves can be computed using Taskar's potentials follows much the same structure as swap moves.

Consider a potential ψ_c defined over a clique c, while we perform a expansion over the label α. Let $c_\beta \subseteq c$ be the set of all variables currently taking label β. There are two case to consider:

1. $\exists \beta \neq \alpha : c_\beta = c$ — in this case the entire clique is homogeneously labelled. Consequently, the cost of all moves considered can be formed as a standard P^N potential and solved in the manner discussed previously.

2. $\nexists\beta : c_\beta = c$ — As the clique is currently heterogeneously labelled, the only way it can become homogeneously labelled is to completely take label α. This is equivalent to a P^n potential of the form:

$$\psi_c(\mathbf{x}_c) = \begin{cases} \gamma_\alpha, c & \text{if } x_i = \alpha \ \forall i \in c \\ \gamma_{c,\max} & \text{otherwise.} \end{cases} \tag{1.42}$$

As such the optimal moves can be proposed using Taskar's LP formulation.

Solving Higher-order Associative Potentials with graph-cuts

We will now show that a potential of the form

$$\psi_c(\mathbf{x}_c) = -k_c \prod_{i \in c} \Delta(x_i = 0) \tag{1.43}$$

can be solved using graph-cuts over a two label range.

To do this, we consider a graph $G =< V, E >$ defined as in section 1.3.1, which contains vertices V_i for every $X_i \in c$ and additional source s and sink t vertices. We adjoin an extra auxiliary variable V_{aux} to the vertices in the graph, which we connect with a directed edge to all vertices V_i in the clique, and to the sink with weight k_c. We want to show that whatever labels are taken by variables in the clique, the cost of the final minimum cut will be the same as equation (1.43) up to reparameterisation. The resulting pairwise cost is of the form:

$$k_c V_{\text{aux}} + \sum_{i \in c} k_c V_i (1 - V_{\text{aux}}). \tag{1.44}$$

Fixing $\mathbf{V}_c$ the cost of the minimum cut is:

$$\min_{V_{\text{aux}}} \left(k_c V_{\text{aux}} + \sum_{i \in c} k_c V_i (1 - V_{\text{aux}}) \right). \tag{1.45}$$

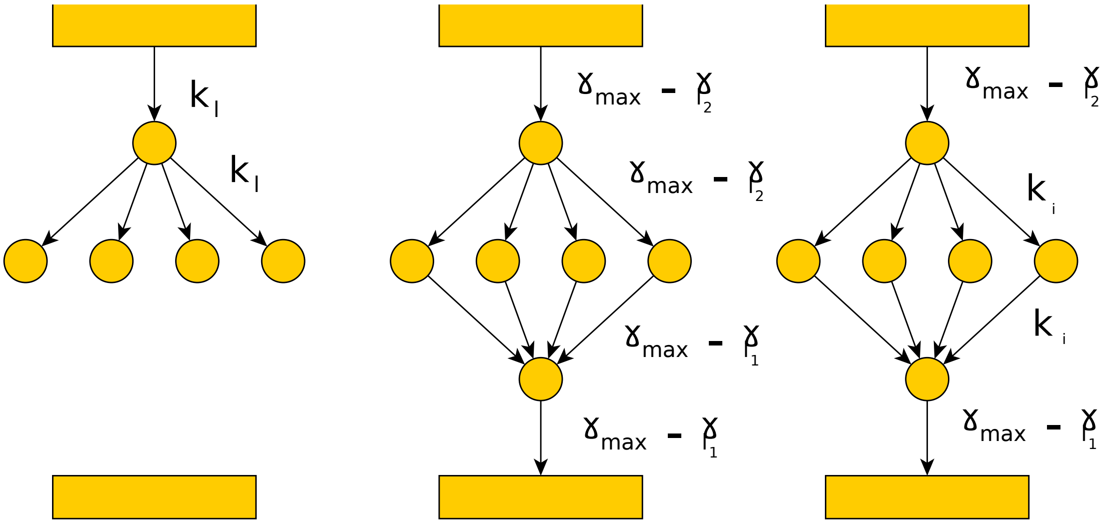

Figure 1.7: *Illustration of the graph construct used to solve higher-order potentials in section 1.3.6. From left to right: The graph construct used to solve Taskar's Associative Higher Order potential; The P^n potential; and the robust P^n potential.*

By inspection, this has a a cost of 0 if all v_I are tied to the source, or equivalently, if all of $\mathbf{x}_c$ takes label zero; and a cost of at most k_c otherwise; and is always of at least cost k_c if one variable in $\mathbf{x}_c$ does not take label 0. By reparameterising the cost by $-kc$, we get a potential of the form of equation (1.43).

By symmetry, a similar approach, that swaps V_i with $(1 - V_i)$ and V_{aux} with $(1 - V_{\text{aux}})$ gives potentials of the form $\prod_{i \in c} \Delta(V_i = 1)$. Consequently, Taskar's Higher-Order potentials can be exactly solved in the binary case using graph-cuts, and it is possible to perform efficient $\alpha\beta$-swap and α-expansion over these higher-order potentials.

Chapter 2

Associative Hierarchical Networks

2.1 Overview

This section introduces the new model of Associative Hierarchical Networks (AHN), first proposed in Ladicky et al. (2009). We will first motivate their application as a true multi-scale model for object class segmentation that can integrate cues from an arbitrary number of scales in a principled manner. Moreover, this model allows for efficient MAP estimation as described in the following chapter. Secondly, we will show a recent application proposed in Ladicky et al. (2010c), which integrates detectors with segmentation.

2.2 Introduction

A fundamental problem in semantic segmentation lies in the choice of image quantisation. Rather than individually labelling each pixels in an image, the image may be first clustered into super-pixels, and then the super-pixels may be classified themselves. Each choice in scale carries with it its own advantages and

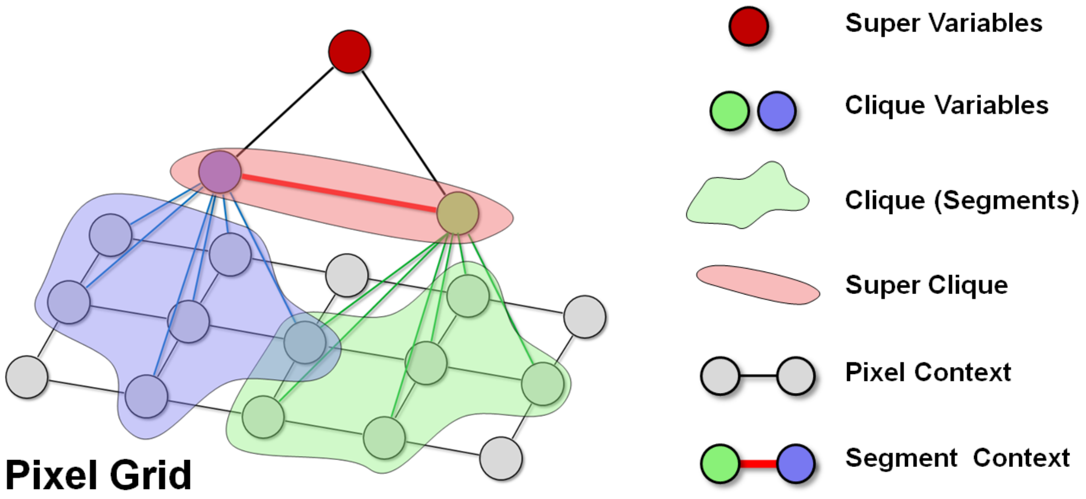

Figure 2.1: A simplified 3-layer AHN, composed of a pixel-based grid layer, a layer representing a segment CRF and a higher-order consistency term defined over segments. Note that although the diagram does not show multiple intersecting hierarchies, this is for clarity's sake, and not a limitation of our model.

disadvantages — overly large super-pixels may span object classes, preventing the image from being correctly labelled. On the other hand, a choice to use very small super-pixels or pixels means that in order to make use of coarse image features, defined over large regions of the image, requires the use of aggregate features from over-lapping regions. This runs the risk of over counting features that occur only once in the image, but in many of these over-lapping regions, leading to errors in the final labelling of the image. See figure 2.2, for an example of this issue.

One approach to dealing with the difficulty of choosing a good quantisation a priori is to delay the choice of quanta until much later. This allows us to pick super-pixels that are consistent with a 'good' labelling of the image. Gould et al. (2009b) proposed an approach in which the choice of super-pixels was integrated with the labelling of the image with object instances. Under their interpretation, super-pixels should physically exist and represent either the entirety of an object or a planar facet if the object class is amorphous and can not be decomposed into individual objects (this includes classes such as *grass*, *building*, or *sky*). Con-

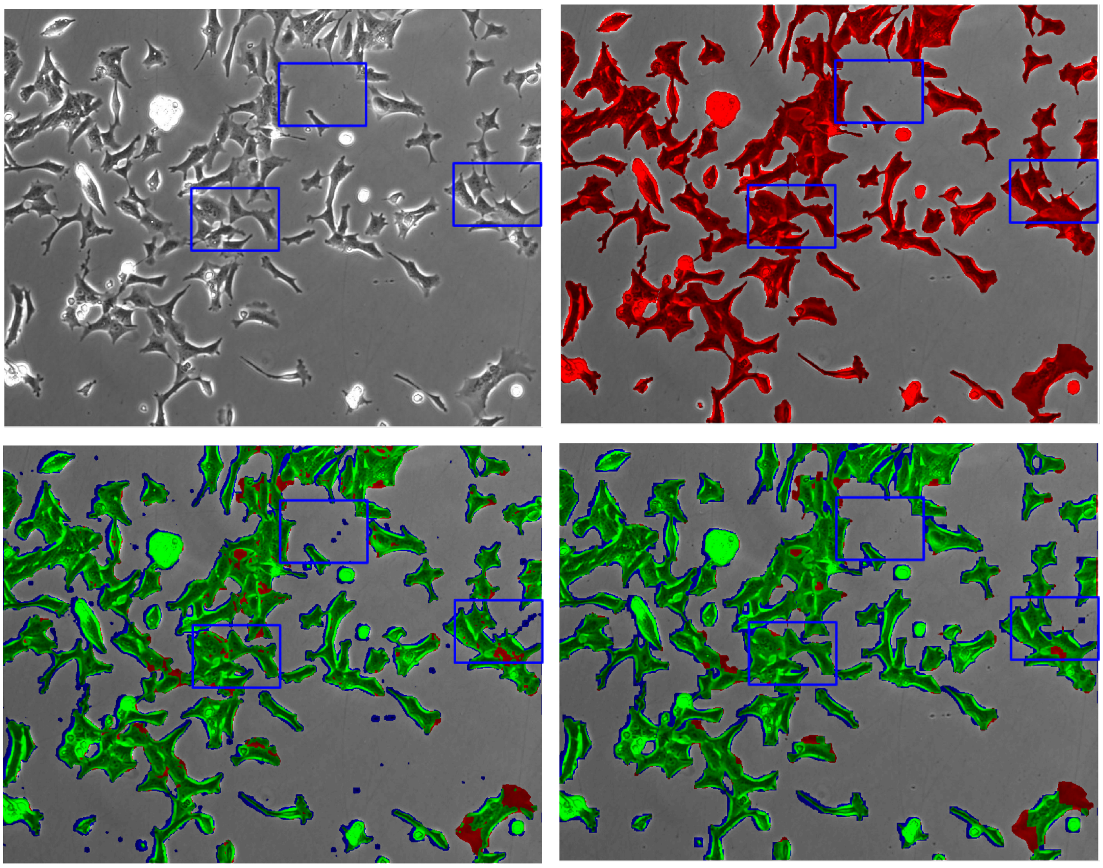

Figure 2.2: A demonstration of the problems caused by over-counting: *Top Left:* a sample image of cells on a slide;*Top Right:* the human based ground truth labelling;*Bottom Left:* The labelling from a pairwise CRF in which the unary potentials are based on an aggregate features about each pixel; *Bottom Right:* The labelling from a P^n CRF in which the appearance of each region directly governs the labelling of that entire region. *Key for bottom row:* **Green:** True positives, correctly labelled cell pixels; **Red:** False negatives, cell structure incorrectly labelled as background; **Blue:** False positives, background incorrectly labelled as slide. In comparison to the higher order CRF the pairwise CRF is more vulnerable to anomalous textures. In the pairwise CRF overly smoothed or specular regions *and their neighbours* are mislabelled and this makes it much less likely that pairwise smoothing terms will be able to recover from these errors. The elimination of these errors leads to a much tighter boundaries in the higher order CRF. The average distance of a false positive from the nearest true positive dropped from 11.3 pixels (pairwise CRF) to 3.4 pixels (higher-order CRF). See Russell et al. (2007) for details.

sequently, in their final labelling each pixel belongs to exactly one super-pixel chosen to represent a single instance of an object.

This process of shaping super-pixels to match object outlines is computationally challenging. As discussed in Gould et al. (2009c), the optimisation techniques proposed frequently fail to recognise individual instances. Their algorithm is often unable to merge the super-pixels contained within a single instance, even if the super-pixels are correctly labelled by class. The recent work by Kumar and Koller (2010) goes some way to addressing these issues. By using sophisticated LP-relaxations they are able to trade computation time against the quality of the solution found. However, the solutions found still lack the approximation guarantees of our work, as described in the following chapter.

Another approach, and one closely related to ours was proposed in Kohli et al. (2008). By formulating the labelling problem as a CRF defined over pixels, they were able to recover from misleading segments which spanned multiple object classes. Further, they could encourage individual pixels within a single segment to share the same label, by defining higher order potentials (functions defined over cliques of size greater than 2) that penalised heterogeneous labellings of segments. Their method can be understood as a relaxation of the hard constraint of previous methods, that state that the image labelling must follow the quantisation of the image space, to a softer constraint in which a penalty is paid for failure to conform.

In this section we describe a novel hierarchical CRF formulation of the object class segmentation problem that allows us to unify multiple disparate quantisations of the image space, avoiding the need to make a decision of which is most appropriate. It allows for the integration of features derived from different quantisation levels (pixel, segment, and segment union/intersection). By way of comparison with the work of Gould et al. (2009b), while they explicitly choose a super-pixel for every pixel, we allow each pixel to simultaneously belong to several super-pixels. Under our framework, each of these super-pixels proposes a set of hypotheses. These hypotheses describe the correlation between members of the super-pixel, the likelihood that the super-pixel will predominantly belong to some particular class, or the correlation between the label of this super-pixel and its neighbours. We will demonstrate how many of the state-of-the-art methods based on different fixed image quantisations can be seen as special cases of our model.

Inferring the minimum cost solution in this framework involves the minimisation of a extremely high order function. In the following chapter, we show that the solutions of such difficult function minimisation problems may be efficiently computed using graph cut based move-making algorithms, and provide bounds which guarantee the quality of the solution found. We evaluate the efficacy of our framework on some of the most challenging data-sets for object class segmentation, and show that it outperforms existing state of the art methods based on individual image quantisation levels.

Robust P^n and Hierarchical CRFs The pairwise CRF formulation of (Lafferty et al., 2001; Shotton et al., 2006) was extended by (Kohli et al., 2008) with the incorporation of robust higher order potentials defined over segments. Their formulation was based upon the observation that pixels lying with in the same super-pixel or cluster are more likely to take the same label. As discussed in chapter 1.2.3, the energy of the higher order CRF proposed by (Kohli et al., 2008) was of the form:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{N}} \psi_{ij}(x_i, x_j) + \sum_{\substack{c \in \mathcal{C} \\ |c| > 2}} \psi^h_c(\mathbf{x}_c), \tag{2.1}$$

where $\mathcal{C}$ refers to a set cliques corresponding to image regions (or segments), and ψ_c are higher order potentials defined over them. As described in the previous chapter, their higher order potentials took the form of the Robust P^N model defined as:

$$\psi_c^h(\mathbf{x}_c) = \min_{l \in \mathcal{L}} \left(\gamma_{c,\max}, \gamma_{c,l} + \sum_{i \in c} k_{c,i} \Delta(x_i \neq l) \right), \tag{2.2}$$

where $\gamma_c^l \leq \gamma_c^{\max}, \forall l \in \mathcal{L}$. This framework enabled the integration of multiple quantisations (segmentations) of the image space in a principled manner. However, they did not use these costs to define unary potentials for segments and were unable to model contextual relations between segments.

In the following chapter, we show that this potential (2.2) can be represented as a pairwise graph using a single auxiliary variable y_c, that takes values from an extended label set $\mathcal{L} \cup \{l_F\}$ as

$$\psi_c^h(\mathbf{x}_c, y_c) = \phi_c(y_c) + \sum_{x_i \in c} \phi_c(y_c, x_i). \tag{2.3}$$

where the unary auxiliary potential $\phi_c(y_c)$ assigns the cost γ_c^l for y_c taking the first $|\mathcal{L}|$ labels and $\gamma_c^{\max}$ for the *free* label l_F and the pairwise potential $\phi_c(y_c, x_i)$ is defined as:

$$\phi_c(y_c, x_i) = \begin{cases} 0 & \text{if} \quad y_c = l_F \text{ or } y_c = x_i \\ k_c^l & \text{if} \quad y_c = l \in \mathcal{L} \text{ and } x_i \neq l. \end{cases} \tag{2.4}$$

This framework can be naturally generalised to a hierarchical model, in which pairwise connections between elements of the same layer of the hierarchy are supported, and the connection between layers take the same form as the robust P^N model (Eq. 2.2).

It is defined as:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{S}^{(1)}} \psi_i^{(1)}(x_i^{(1)}) + \sum_{(i,j) \in \mathcal{N}^{(1)}} \psi_{ij}^{(1)}(x_i^{(1)}, x_j^{(1)}) + \min_{\mathbf{x}^{(2)}} E^{(2)}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}), \tag{2.5}$$

where $E^{(2)}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ is recursively defined as:

$$E^{(n)}(\mathbf{x}^{(n-1)}, \mathbf{x}^{(n)}) = \sum_{c \in \mathcal{S}^{(n)}} \psi_c^{(n)}(x_c^{(n-1)}, x_c^{(n)}) + \sum_{(c,d) \in \mathcal{N}^{(n)}} \psi_{cd}^{(n)}(x_c^{(n)}, x_d^{(n)}) + \min_{\mathbf{x}^{(n+1)}} E^{(n+1)}(\mathbf{x}^{(n)}, \mathbf{x}^{(n+1)}), \tag{2.6}$$

without losing the ability to solve this model with graph-cut based move making algorithms (see following chapter).

In our application, we form hierarchies by recursively applying multiple clustering algorithms to previous generated clusters. Just as in the P^N model, pixels are associated with clusters and encouraged to share the same label, in our model auxiliary clusters themselves will be associated with 'super-clusters' and encouraged to take the same label.

2.2.1 Relation to Previous Models

Next we draw some comparisons with the current state of the art models for object segmentation (Galleguillos et al., 2008; Pantofaru et al., 2008; Rabinovich et al., 2007; Yang et al., 2007) and show that at certain choices of the parameters of our model, these methods fall out as special cases. Thus, our method does not only generalise the standard pairwise CRF formulation over pixels, but also the previous work upon super-pixels and (as we shall see) provides a global optimisation framework which allows us to combine all choices of image quantisations.

Equivalence to CRFs based on Segments Consider a hierarchy defined over two layers: the pixel grid and a first layer of clusters, and without unary or pairwise potentials defined over individual pixels, such that all segments $c \in \mathcal{S}$ are disjoint (non-overlapping)[1]. The potentials $\psi_c^h(\mathbf{x}_c, y_c)$ are both *semi-metric*

[1] This is equivalent to the case where only one particular quantisation of the image space is considered.

and *symmetric* (Boykov et al., 2001) for any pair of pixels within the segment, forcing them to take the same label in a minimal cost labelling. Thus, the optimal labelling will be segment-consistent. The cost of every segment-consistent labelling is

$$E(\mathbf{y}) = \sum_{c \in \mathcal{S}} \psi_c(y_c) + \sum_{(c,d) \in \mathcal{N}^{(2)}} \psi_{cd}(y_c, y_d) \tag{2.7}$$

and is exactly the same as the cost associated with the pairwise CRF defined over segments with $\psi_c(y_c)^l = \gamma_c^l$ is the unary cost and ψ_{cd} as the pairwise costs for each segment. For sufficiently large $\gamma_c^{\max}$ auxiliary variables will not take the label l_F, which means that all pixels connected to them will take the same label (behave as one unit). In this case, our model becomes equivalent to the pairwise CRF models defined over segments rather than pixels such as those given by Batra et al. (2008); Galleguillos et al. (2008); Rabinovich et al. (2007); Yang et al. (2007).

Equivalence to tree structured associative models Various tree structured hierarchies such as Zhu and Yuille (2005); Lim et al. (2009); Nowozin et al. (2010); Reynolds and Murphy (2007) have been proposed for semantic segmentation. The structure of these models is clearly a strict subset of ours, as it does not support pairwise connections between variables in the same level, and each variable may only be attached to one variable in the layer above. Consequently if the label space and edge costs between parent and child are of the same form as those we consider, these models can also be contained in our approach. See figure 2.3 for more details.

The Relationship with Directed Models A hierarchical, two-layer, directed model was proposed in Kumar and Hebert (2005). This is a hybrid model relatively similar to ours, with unary and pairwise potentials defined over both super-pixels and pixels and pairwise connections between the layers, enforcing consistency. However, it principally differs from ours in the use of directed edges between layers. These directed edges mean that max-marginals can be computed in a piecewise manner, and propagated from one layer to the other, making it suitable for inference with message passing algorithms (our model is shown to be ill-suited for message passing algorithms such as belief propagation and TRW-S in the following chapter).

This directed approach does not propagate information through-out the structure. In order to arrive at a consistent hypothesis, that takes account conflicting clues from all levels of the hierarchy, there are two desirable criteria for the propagation of information.

1. We wish for information to be transmitted from the pixel to the segment level and from there back to the pixel level. That is, the labelling of one pixel should affect the label of the segment potential and, from this, the label of other pixels in the same segment.

2. Information should also be transmitted from a segment to the pixel level and back to the segment level. This means that if two segments overlap, the optimal label of one segment should indirectly depend on the labelling of the other.

If the connections between layers form a directed acyclic graph (DAG) as they do in Kumar and Herbet's model, and in the related structure of *Deep Belief Nets* (Bengio et al., 2007; Hinton et al., 2006), at most one of these conditions can hold — both conditions together describe a cycle. Both deep belief nets, and Kumar and Hubert's approach only satisfy the first criteria. In simple hierarchies such as those proposed by Kumar and Herbert, in which each pixel belongs to only one segment, the second criteria is less important. However, the integration of multiple segmentations into a single coherent hypothesis depends upon the transmission of information in both directions.

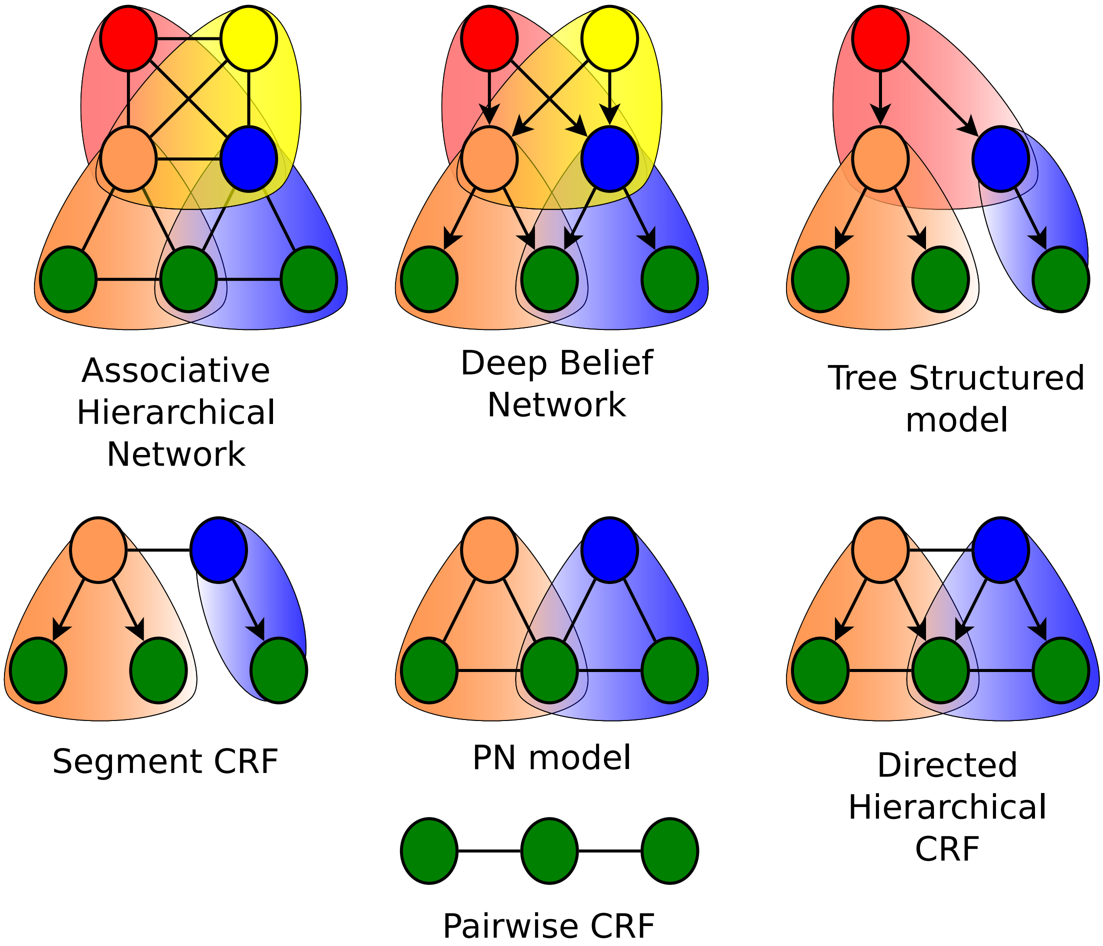

Figure 2.3: Pairwise graphical representations of our approach and various other models. As neither the segment CRF nor the tree-based CRF contains any loops in the directed portion of the graph they are equivalent to undirected models (Pearl, 1998).

	Global	Average	Building	Grass	Tree	Cow	Sheep	Sky	Aeroplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
(Shotton et al., 2008)	72	67	49	88	79	97	97	78	82	54	87	74	72	74	36	24	93	51	78	75	35	66	18
(Shotton et al., 2006)	72	58	62	98	86	58	50	83	60	53	74	63	75	63	35	19	92	15	86	54	19	62	07
(Batra et al., 2008)	70	55	68	94	84	37	55	68	52	71	47	52	85	69	54	05	85	21	66	16	49	44	32
(Yang et al., 2007)	75	62	63	98	89	66	54	86	63	71	83	71	79	71	38	23	88	23	88	33	34	43	32
Pixel CRF	81	72	73	92	85	75	78	92	75	76	86	79	87	96	95	31	81	34	84	53	61	60	15
Segment CRF	75	60	64	95	78	53	86	99	71	75	70	71	52	72	81	20	58	20	89	26	42	40	05
Hierarchical CRF	86	75	80	96	86	74	87	99	74	87	86	87	82	97	95	30	86	31	95	51	69	66	09

Table 2.1: *Quantitative results on the MSRC data set. The pixel accuracy (%) for different object classes.*

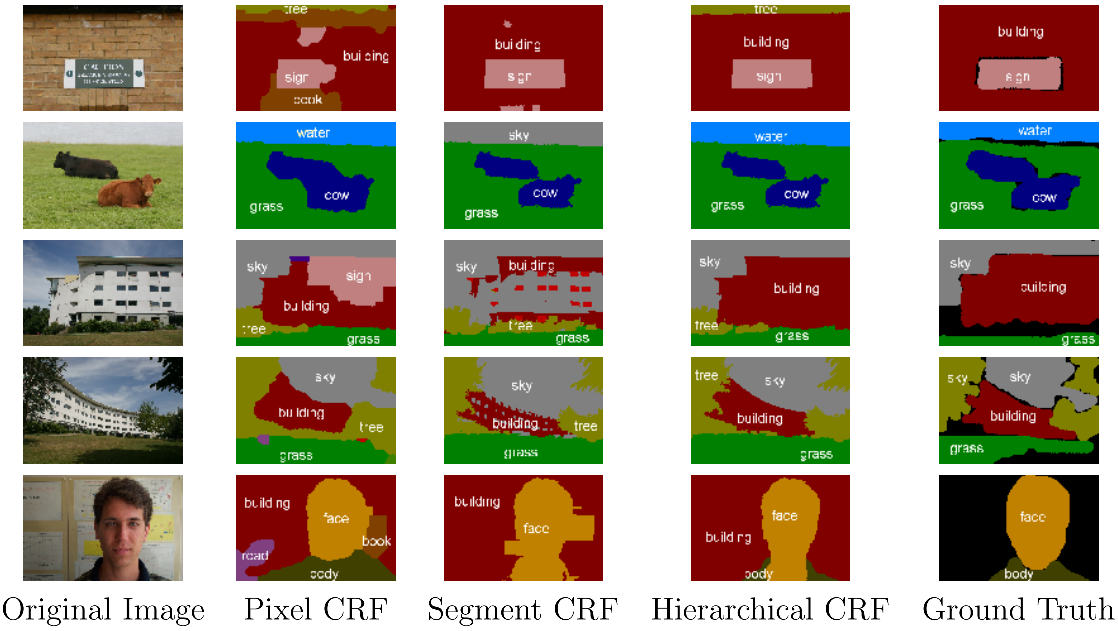

Figure 2.4: *Qualitative results on the MSRC-21 data-set using the* range α-expansion *algorithm discussed in the next chapter. Pixels marked black in the hand-labelled ground truth image are unlabelled. The potentials used in these experiments are described in Ladicky et al. (2009).*

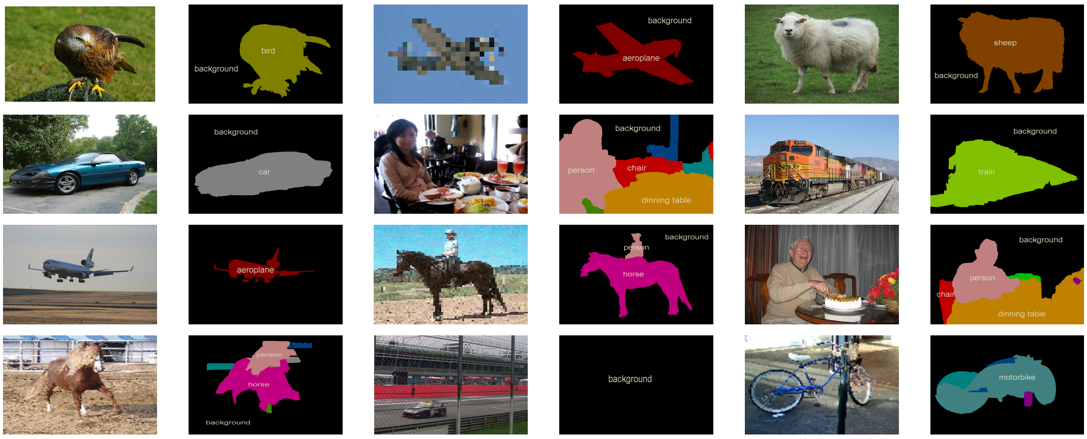

Figure 2.5: *Qualitative results on the VOC-2008 data-set Qualitative results on the MSRC-21 data-set using the* range α-expansion *algorithm discussed in the next chapter. Successful segmentations (top 3 rows) and standard failure cases — context error, detection failure and miss-classification (bottom).*

	Average	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	Tv/monitor
XRCE	25.4	75.9	25.8	15.7	19.2	21.6	17.2	27.3	25.5	24.2	7.9	25.4	9.9	17.8	23.3	34.0	28.8	23.2	32.1	14.9	25.9	37.3
UIUC / CMU	19.5	79.3	31.9	21.0	8.3	6.5	34.3	15.8	22.7	10.4	1.2	6.8	8.0	10.2	22.7	24.9	27.7	15.9	4.3	5.5	19.0	32.1
MPI	12.9	75.4	19.1	7.7	6.1	9.4	3.8	11.0	12.1	5.6	0.7	3.7	15.9	3.6	12.2	16.1	15.9	0.6	19.7	5.9	14.7	12.5
Hierarchical CRF	20.1	75.0	36.9	4.8	22.2	11.2	13.7	13.8	20.4	10.0	8.7	3.6	28.3	6.6	17.1	22.6	30.6	13.5	26.8	12.1	20.1	24.8

Table 2.2: *Quantitative analysis of VOC2008 results. Note that all other methods used classification and detection priors trained from the whole data-set including non-segmented images.*

2.3 Combining Object Detectors and AHNs

For the purpose of this section, a detector should be considered to be a black box process which takes an image as an input and returns a set of bounding boxes that says where an instance of an object is likely to appear in the image. The use of detectors is consequently restricted to class such as *person* or *sheep* which can be decomposed into instances. This makes them well suited to be integrated with standard approaches to object class segmentation, as the classifiers used are typically texture-based and better suited for amorphous classes such as *road* or *water* that can not readily be broken down into instances. Within the literature, instance-based classes are typically referred to as *things*, while amorphous classes are referred to as *stuff* (Adelson, 2001).

In Associative Hierarchical Networks, the process of inference can be understood as a soft competition among different hypotheses (defined over pixel or segment random variables), in which the final solution maximises the weighted agreement between them. These weighted hypotheses are potentials in the AHN. In object class recognition, these hypotheses encourage: (i) variables to take particular labels (unary potentials), and (ii) agreement between variables (typically pairwise). Existing methods including a naive application of AHNs and (He et al., 2004; Yang et al., 2007) are limited to such hypotheses provided by pixels and/or

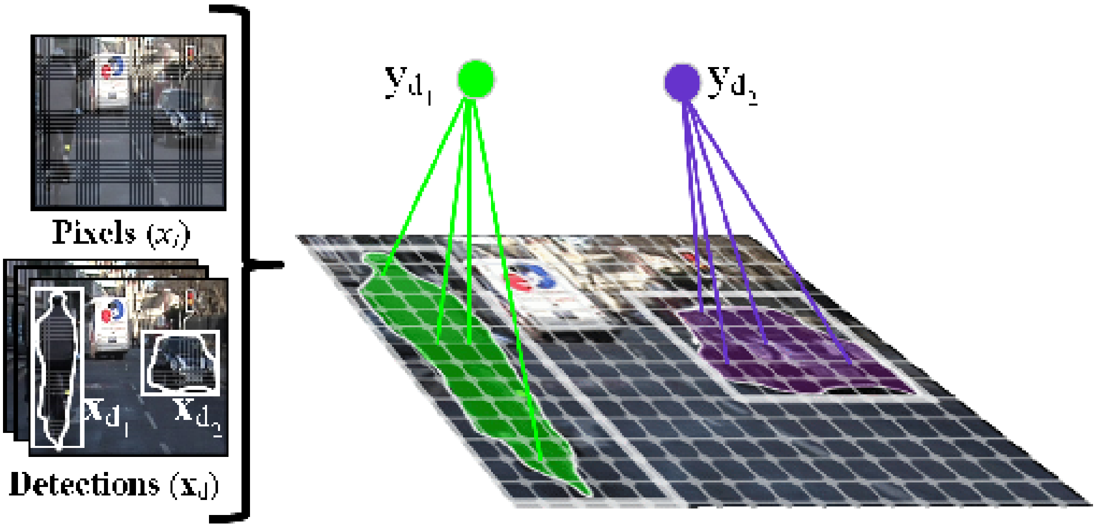

Figure 2.6: *Inclusion of object detector potentials into a* AHF*. We show a pixel-based* CRF *as an example here. The set of pixels in a detection d_1 (corresponding to the bicyclist in the scene) is denoted by $\mathbf{x}_{d_1}$. A higher order clique is defined over this detection window by connecting the object pixels $\mathbf{x}_{d_1}$ to an auxiliary variable $y_{d_1} \in \{0, 1\}$. This variable allows the inclusion of detector responses as soft constraints. (**Best viewed in colour**)*

segments only. We introduce an additional set of hypotheses representing object detections for the recognition framework.

Some object detection approaches (Felzenszwalb et al., 2008; Larlus and Jurie, 2008) have used their results to perform a segmentation within the detected areas[2]. These approaches include both the true and false positive detections, and segment them assuming they all contain the objects of interest. There is no way of recovering from these erroneous segmentations. Our approach overcomes this issue by using the detection results as hypotheses that can be rejected in the global CRF energy. In other words, all detections act as soft constraints in our framework, and must agree with other cues from pixels and segments before affecting the object class segmentation result.

Let $\mathcal{D}$ denote the set of object detections, which are represented by bounding boxes enclosing objects, and corresponding scores $H_d, d \in \mathcal{D}$ that indicate the strength of the detections. We define a novel clique potential ψ_d over the set of

[2] As evident in some of the PASCAL VOC 2009 segmentation challenge entries.

pixels $\mathbf{x}_d$ belonging to the d-th detection (e.g.pixels within the bounding box), with a score H_d and detected label l_d. Figure 2.6 shows the inclusion of this potential graphically on a pixel-based CRF. The new energy function is given by:

$$E(\mathbf{x}) = E_{pix}(\mathbf{x}) + \sum_{d \in \mathcal{D}} \psi_d(\mathbf{x}_d, H_d, l_d), \tag{2.8}$$

where $E_{pix}(\mathbf{x})$ is any standard pixel-based energy. The minimisation procedure should be able to reject false detection hypotheses on the basis of other potentials (pixels and/or segments). To do this, we create a new layer in our hierarchy to hold detector potentials. Given a box d, for class c with a detector response s , we wish to create a segment associated with it, that captures the object lying in the centre of the box. This can be easily done using either a parametric max-flow (Gallo et al., 1989) centred on the box, to create a segment of the correct size, or with grab-cut (Rother et al., 2004). We then define the unary potential for this segment as:

$$\psi_d(x_d^{(2)}) = \begin{cases} \alpha H_d & \text{if } x_d^{(2)} = c \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

where α is some arbitrary weighting that describes much attention should be paid to a detector response versus the other potentials of the AHN.

Figure 2.6 illustrates this model. Note that this model is of the standard form defined in ((2.5)), and consequently can be solved efficiently using the techniques in the following chapter. As expected the inclusion of these potentials provides a substantial improvement to results (see figures 2.3, 2.7).

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	TV/monitor	**Average**
BONN_SVM-SEGM	**83.9**	64.3	21.8	21.7	**32.0**	**40.2**	**57.3**	49.4	**38.8**	5.2	**28.5**	22.0	19.6	33.6	45.5	33.6	27.3	**40.4**	18.1	33.6	**46.1**	**36.3**
CVC_HOCRF	80.2	**67.1**	**26.6**	**30.3**	31.6	30.0	44.5	41.6	25.2	5.9	27.8	11.0	23.1	**40.5**	**53.2**	32.0	22.2	37.4	**23.6**	40.3	30.2	34.5
UOCTTI_LSVM-MDPM	78.9	35.3	22.5	19.1	23.5	36.2	41.2	50.1	11.7	8.9	**28.5**	1.4	5.9	24.0	35.3	33.4	**35.1**	27.7	14.2	34.1	41.8	29.0
NECUIUC_CLS-DTCT	81.8	41.9	23.1	22.4	22.0	27.8	43.2	**51.8**	25.9	4.5	18.5	18.0	**23.5**	26.9	36.6	**34.8**	8.8	28.3	14.0	35.5	34.7	29.7
LEAR_SEGDET	79.1	44.6	15.5	20.5	13.3	28.8	29.3	35.8	25.4	4.4	20.3	1.3	16.4	28.2	30.0	24.5	12.2	31.5	18.3	28.8	31.9	25.7
BROOKESMSRC_AHCRF	79.6	48.3	6.7	19.1	10.0	16.6	32.7	38.1	25.3	5.5	9.4	25.1	13.3	12.3	35.5	20.7	13.4	17.1	18.4	37.5	36.4	24.8
Our method	81.2	46.1	15.4	24.6	20.9	36.9	50.0	43.9	28.4	**11.5**	18.2	**25.4**	14.7	25.1	37.7	34.1	27.7	29.6	18.4	**43.8**	40.8	32.1

Table 2.3: *Quantitative analysis of* AHN + *detectors on the*VOC *2009 test data-set results (Everingham et al., 2009) using the intersection vs union performance measure. Our method is ranked* ***third*** *when compared the 6 best submissions in the 2009 challenge. The method* UOCTTI_LSVM-MDPM *is based on an object detection algorithm (Felzenszwalb et al., 2008) and refines the bounding boxes with a Grab-Cut style approach. The method* BROOKESMSRC_AHCRF *is the* CRF *model used as an example in our work. We perform better than both these baseline methods by* 3.1% *and* 7.3% *respectively. Underlined numbers in bold denote the best performance for each class.*

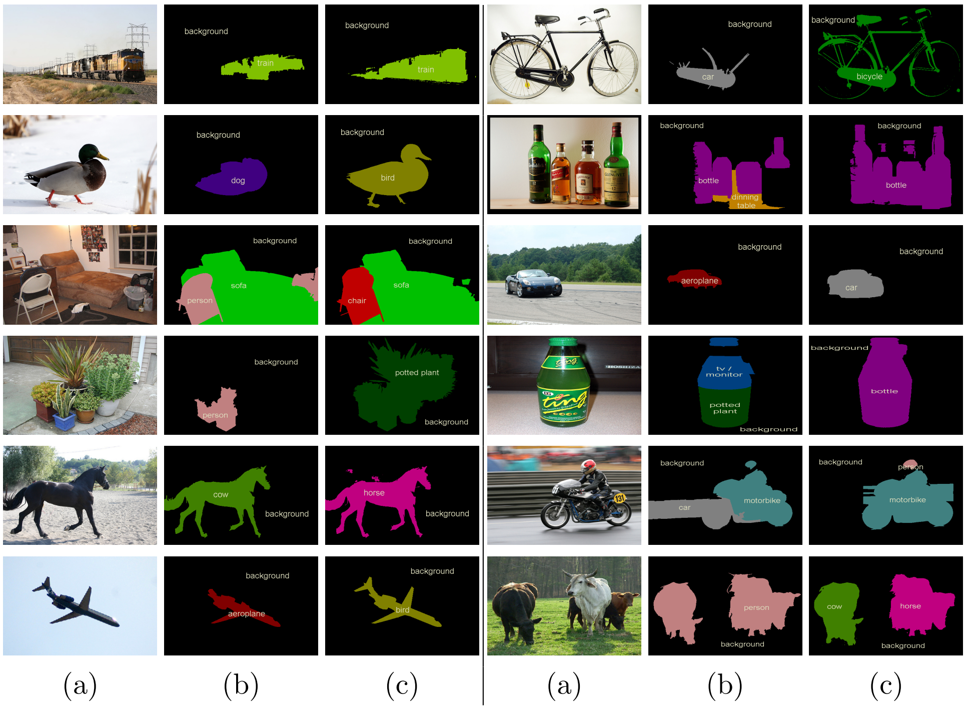

Figure 2.7: *(a) Original test image from* PASCAL VOC *2009 data-set (Everingham et al., 2009), (b) The labelling obtained by* AHN*s without object detectors, (c) The labelling provided by our method which includes detector based potentials. Note that no ground truth is publicly available for test images in this data-set. Examples shown in the first five rows illustrate how detector potentials not only correctly identify the object, but also provide very precise object boundaries, e.g. bird (second row), car (third row). Some failure cases are shown in the last row. This was caused by a missed detection or incorrect detections that are very strong and dominate all the other potentials. (****Best viewed in colour****)*

Chapter 3

Exact and Approximate Inference in Associative Hierarchical Networks using Graph Cuts

3.1 Overview

Within this chapter we provide a computationally efficient method for approximate inference based on graph cuts. Our method performs well for networks containing hundreds of thousand of variables, and higher order potentials are defined over cliques containing tens of thousands of variables. Due to the size of these problems standard linear programming techniques are inapplicable. We show that our method has a bound of 4[1] for the solution of general associative hierarchical network with arbitrary clique size. Apart from this work, we are unaware of any methods that provide bounds that are independent of clique size.

[1] This means that the cost of the solution found is guaranteed to lie within a factor of 4 of the cost of the minimal cost labelling.

3.2 Introduction

The last few decades have seen the emergence of Markov networks or random fields as the most widely used probabilistic model for formulating problems in machine learning and computer vision. This interest has led to a large amount of work on the problem of estimating the maximum a posteriori (MAP) solution of a random field (Szeliski et al., 2006; Kolmogorov, 2006; Komodakis and Paragios, 2008; Kumar and Torr, 2008a; Sontag et al., 2008; Wainwright et al., 2005; Weiss and Freeman, 2001). However, most of this research effort has focused on inference over pairwise Markov networks. Of particular interest are the families of associative pairwise potentials (Taskar et al., 2004) discussed in chapter 1, in which connected variables are assumed to be more likely than not to share the same label. Inference algorithms targeting these associative potentials, which include truncated convex costs (Kumar and Torr, 2008b), metrics (Boykov et al., 2001), and semi metrics (Kumar and Koller, 2009), often carry bounds which guarantee the cost of the solution found must lie within a bound, specified as a fixed factor of n of the cost of the minimal solution.

Although higher order Markov networks (i.e. those with a clique size greater than two) have been used to obtain impressive results for a number of challenging problems in computer vision (Roth and Black, 2005; Komodakis and Paragios, 2009; Vicente et al., 2009; Ladicky et al., 2010c; Kohli et al., 2009; Lan et al., 2006; Werner, 2009; Potetz and Lee, 2008; Woodford et al., 2008; Rother et al., 2009) the problem of bounded higher order inference has been largely ignored.

In this chapter, we address the problem of performing graph cut based inference in a new model: the Associative Hierarchical Networks (AHNs) described in chapter 2, that includes the higher order Associative Markov Networks (AMNs) (Taskar et al., 2004) or P^n potentials (Kohli et al., 2007) and the Robust P^n (Kohli et al., 2008) model as special cases, and derive a bound of 4.

In general, these AHNs are suitable for the representation of any problem that is Potts-like (i.e. those that encourage homogeneous labellings and penalise all forms of heterogeneity equally) across any number of arbitrary scales.

For a set of variables $\mathbf{x}^{(1)}$ AHNs are characterised by energies (or costs) of the form:

$$E(\mathbf{x}^{(1)}) = E'(\mathbf{x}^{(1)}) + \min_{\mathbf{x}^a} E^a(\mathbf{x}^{(1)}, \mathbf{x}^a) \tag{3.1}$$

where E' and E^a are pairwise AMNs and $\mathbf{x}^a$ is a set of auxiliary variables. The AHN is a AMN containing higher order cliques, defined as a function of $\mathbf{x}^{(1)}$, but can also be seen as a pairwise AMN defined in terms of $\mathbf{x}^{(1)}$ and $\mathbf{x}^a$. We propose new move making algorithms over the pairwise energy $E'(\mathbf{x}^{(1)}) + E^a(\mathbf{x}^{(1)}, \mathbf{x}^a)$ which have the important property of *transformational optimality.*

Move making algorithms function by efficiently searching through a set of candidate labellings and proposing a *optimal* candidate i.e. one with the lowest energy to move to. The set of candidates is then updated, and the algorithm repeats till convergence.

We call a move making algorithm *transformationally optimal* if and only if any move $(\mathbf{x}^*, \mathbf{x}^a)$ proposed by the algorithm satisfies the property:

$$E^a(\mathbf{x}^*, \mathbf{x}^a) = \min_{\mathbf{x}'} E^a(\mathbf{x}^*, \mathbf{x}') \tag{3.2}$$

i.e. $\mathbf{x}^a$ is a minimiser of $E^a(\mathbf{x}^*, \cdot)$. Inserting this into equation (3.1) we have:

$$E(\mathbf{x}^*) = E'(\mathbf{x}^*) + E^a(\mathbf{x}^*, \mathbf{x}^a). \tag{3.3}$$

This implies that the partial move $\mathbf{x}^*$ proposed by a transformationally optimal algorithm over $E'(\mathbf{x}^{(1)}) + E^a(\mathbf{x}^{(1)}, \mathbf{x}^a)$ must function as a move that directly minimise the higher order cost of equation (3.1). Experimentally, our transformationally optimal algorithms converge faster, and to better solutions than standard approaches, such as α-expansion. Moreover, unlike standard approaches, our transformationally optimal algorithms always find the exact solution for binary AHNs.

Outline of the chapter Existing models generalised by the associative hierarchical network, and the full definition of AHNs are given in section 3.3. In section 3.4 we discuss work on efficient inference, and show how the pairwise form of associative hierarchical networks can be minimised using the α-expansion algorithm, and derive bounds for our approach. Section 3.5 discusses the application of novel move making algorithms to such energies, and we show that under our formulation the moves of the robust P^n model become equivalent to a more general form of range moves over unordered sets. We derive transformational optimality results over hierarchies of these potentials, guaranteeing the optimality of the moves proposed. We experimentally verify the effectiveness of our approach against other methods in section 3.6, and conclude in section 3.7.

At points within this chapter, we will want to distinguish between the original variables of the energy function, whose optimal values we are attempting to find, and the auxiliary variables which we will introduce to convert our higher order function into a pairwise one. We refer to the original variables as the base layer $\mathbf{x}^{(1)}$ (as they lie at the bottom of the hierarchical network). All auxiliary variables at any level h of the hierarchy are denoted by $\mathbf{x}^{(h)}$. The set of indices of variables constituting level h of the hierarchy is denoted by $\mathcal{V}^h$. Similarly, the set of all pairwise interactions at level h is denoted by $\mathcal{E}^h$.

3.3 Associative Hierarchical Networks

Existing higher-order models Taskar et al. (2004) proposed the use of higher order potentials that encourage the entirety of a clique to take some label, and discusses how they can be applied to predicting protein interactions and document classification. These potentials were introduced into computer vision along with an efficient graph cut based method of inference, as the strict P^n Potts model (Kohli et al., 2007).

A generalisation of this approach was proposed by Kohli et al. (2008), who observed that in the image labelling problem, most (but not all) pixels belonging to image segments computed using an unsupervised clustering/segmentation algorithm take the same object label. They proposed a higher order MRF over segment based cliques. The energy took the form:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{ij \in \mathcal{E}} \psi_{ij}(x_i, x_j) + \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c), \tag{3.4}$$

$$\text{where } \psi_c(\mathbf{x}_c) = \min_{l \in \mathcal{L}} \left(\gamma_c^{max}, \gamma_c^l + \sum_{i \in c} k_c^i \Delta(x_i \neq l) \right) \tag{3.5}$$

as discussed in chapter 1. The potential function parameters k_c^i, γ_c^l, and γ_c^{max} are subject to the restriction that $k_c^i \geq 0$ and $\gamma_c^l \leq \gamma_c^{max}, \forall l \in \mathcal{L}$.

We now demonstrate that the higher order potentials $\psi_c(\mathbf{x}_c)$ of the Robust P^n model (2.2) can be represented by an equivalent pairwise function $\psi_c(\mathbf{x}_c^{(1)}, x_c^{(2)})$ defined over a two level hierarchical network with the addition of a single auxiliary variable $x_c^{(2)}$ for every clique $c \in \mathcal{C}$. This auxiliary variable take values from an extended label set $\mathcal{L}^e = \mathcal{L} \cup \{L_F\}$, where L_F, the *'free'* label of the auxiliary variables, allows its child variables to take any label without paying a pairwise penalty.

In general, every higher order cost function can be converted to a $2-$layer associative hierarchical network by taking an approach analogous to that of factor graphs (Kschischang et al., 2001) and adding a single multi-state auxiliary variable. However, to do this for general higher order functions requires the addition of an auxiliary variable with an exponential sized label set (Wainwright and

Jordan, 2008). Fortunately, the class of higher order potentials we are concerned with can be compactly described as AHNs with auxiliary variables that take a similar sized label set to the base layer, permitting fast inference.

The corresponding higher order function can be written as:

$$\begin{aligned} \psi_c(\mathbf{x}_c^{(1)}) &= \min_{x_c^{(2)}} \psi_c(\mathbf{x}_c^{(1)}, x_c^{(2)}) \\ &= \min_{x_c^{(2)}} \left[\phi_c(x_c^{(2)}) + \sum_{i \in c} \phi_{ic}(x_i^{(1)}, x_c^{(2)}) \right]. \end{aligned} \tag{3.6}$$

The unary potentials $\phi_c(x_c^{(2)})$ defined on the auxiliary variable $x_c^{(2)}$ assign the cost γ_l if $x_c^{(2)} = l \in \mathcal{L}$, and γ_{max} if $x_c^{(2)} = L_F$. The pairwise potential $\phi_{ic}(x_i, x_c^{(2)})$ is defined as:

$$\phi_{ic}(x_i, x_c^{(2)}) = \begin{cases} 0 & \text{if } x_c^{(2)} = L_F, \text{ or } x_c^{(2)} = x_i. \\ k_c^i & \text{if } x_c^{(2)} = l \in \mathcal{L}, \text{ and } x_i \neq l. \end{cases} \tag{3.7}$$

General Formulation The scheme described above can be extended by allowing pairwise and higher order potentials to be defined over $\mathbf{x}^{(2)}$ and further over $\mathbf{x}^{(i)}$, which corresponds to higher order potentials defined over the layer $\mathbf{x}^{(i-1)}$. The higher order energy corresponding to the general hierarchical network can be written using the following recursive function:

$$\begin{aligned} E^{(1)}(\mathbf{x}^{(1)}) &= \sum_{i \in \mathcal{V}} \psi_i^{(1)}(x_i^{(1)}) + \sum_{ij \in \mathcal{E}^{(1)}} \psi_{ij}^{(1)}(x_i^{(1)}, x_j^{(1)}) \\ &+ \min_{\mathbf{x}^{(2)}} E^{(2)}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \end{aligned} \tag{3.8}$$

where $E^{(2)}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ is recursively defined as:

$$
\begin{aligned}
E^{(n)}&(\mathbf{x}^{(n-1)}, \mathbf{x}^{(n)}) \\
&= \sum_{c \in V^{(n)}} \phi_c(x_c^{(n)}) + \sum_{c \in \mathcal{V}^{(n)}, i \in c} \phi_{ic}^{(n)}(x_i^{(n-1)}, x_c^{(n)}) \\
&+ \sum_{(c,d) \in \mathcal{E}^{(n)}} \psi_{cd}^{(n)}(x_c^{(n)}, x_d^{(n)}) + \min_{\mathbf{x}^{(n+1)}} E^{(n+1)}(\mathbf{x}^{(n)}, \mathbf{x}^{(n+1)}) \qquad (3.9)
\end{aligned}
$$

and $\mathbf{x}^{(n)} = \{x_c^{(n)} | c \in \mathcal{V}^n\}$ denotes the set of variables at the n^{th} level of the hierarchy, $\mathcal{E}^{(n)}$ represents the edges at this layer, and $\phi_{ic}^{(n)}(\mathbf{x}_c^{(n-1)}, x_c^{(n)})$ denotes the inter-layer potentials defined over variables of layer $n-1$ and n.

While the hierarchical formulation of both Taskar's and Kohli's models can be understood as a mathematical convenience that allows for fast and efficient bounded inference, our earlier work (described in the previous chapter and in Ladicky et al. (2009)) used it for true multi-scale inference, modelling constraints defined over many quantisations of the image.

3.4 Inference

Inference in Pairwise Networks Although the problem of MAP inference is NP-hard for most associative pairwise functions defined over more than two labels, in real world problems many conventional algorithms provide near optimal solutions over grid connected networks (Szeliski et al., 2006). However, the dense structure of hierarchical networks results in frustrated cycles (or fractional tied solutions) and makes traditional reparameterisation based message passing algorithms for MAP inference such as loopy belief propagation (Weiss and Freeman, 2001) and tree-reweighted message passing (Kolmogorov, 2006) slow to converge and unsuitable (Kolmogorov and Rother, 2006). Many of these frustrated cycles can be eliminated via the use of cycle inequalities (Sontag et al., 2008; Werner, 2009), but only by significantly increasing the run time of the algorithm. The graph cut based move making algorithms discussed in chapter 1.3.2 do not suffer from this problem and have been successfully used for minimising pairwise functions defined over densely connected networks that are frequently encountered in vision.Of these move making approaches, only $\alpha\beta$ swap can be directly applied to associative hierarchical networks as the term $\phi_{ic}(x_i, x_c)$, is not a metric nor truncated convex.

Minimising Higher Order Functions A number of researchers have worked on the problem of MAP inference in higher order AMNs. Lan et al. (2006) proposed approximation methods for BP to make efficient inference possible in higher order MRFs. This was followed by the recent works of Potetz and Lee (2008); Tarlow et al. (2008, 2010) in which they showed how belief propagation can be efficiently performed in networks containing moderately large cliques. However, as these methods were based on BP, they were quite slow and took minutes or hours to converge, and lack bounds.

To perform inference in the P^n models, Kohli et al. (2007, 2008), first showed that certain projection of the higher order P^n model can be transformed into submodular pairwise functions containing auxiliary variables. This was used to formulate higher order expansion and swap move making algorithms as discussed in 1.2.3.

The only existing work that addresses the problem of bounded higher order inference is (Gould et al., 2009a) which showed how theoretical bounds could be derived given move making algorithms that proposed optimal moves by exactly solving some sub-problem. In application they used approximate moves which do not exactly solve the sub-problems proposed. Consequently, the bounds they derive do not hold for the methods they propose. However, their analysis can be applied to the P^n (Kohli et al., 2007) model and inference techniques, which do propose optimal moves, and it is against these bounds that we compare our results.

3.4.1 Inference with α-Expansion

We show that by restricting the form of the inter-layer potentials $\psi_c^{(n)}(\mathbf{x}_c^{(n-1)}, x_c^{(n)})$ to that of the weighted Robust P^n model (Kohli et al., 2008) (see (2.2)), we can apply α-expansion to the pairwise form of the AHN.

This requires a transform of all functions in the pairwise representation so that they can be representable as a metric (Boykov et al., 2001). This transformation is non-standard and should be considered a contribution of this work.

We alter the form of the potentials in two ways. First, we assume that all variables in the hierarchy take values from the same label set $\mathcal{L}^e = \mathcal{L} \cup \{L_F\}$. Where this is not true — original variables $\mathbf{x}^{(1)}$ at the base of the hierarchy can not take label L_F — we artificially augment the label set with the label L_F and associate an infinite unary cost with it. Secondly, we make the inter-layer pairwise potentials *symmetric* by performing a local reparameterisation operation.

Before showing the stating the result we will first demonstrate it for a simple 3 label (α, L_F, β) problem using matrix notation.

Expressing unary and pairwise costs as matrices It is frequently convenient to describe pairwise costs as a matrix in which the element of the matrix in the x_i^{th} row, and x_j^{th} column holds the cost of the pairwise potential $\psi_{i,j}(x_i, x_j)$. Unary potentials can also be expressed as matrices under this framework, after all a unary potential defined over X_i can be seen as a pairwise potential defined over X_i and X_j that doesn't vary with the labelling of x_j. For a matrix describing a pairwise cost $\phi_{i,j}(\cdot,\cdot)$ a unary potential of X_i will correspond to a matrix in which each *row* is constant, while a unary term potential of X_j will correspond to a *column* constant matrix.

We wish to find a decomposition of $\phi_{i,j}$ such that:

$$\phi_{i,j}(x_i, x_j) = \psi_i^{(n-1)}(x_i^{(n-1)}) + \Phi_{ic}^{(n)}(x_i^{(n-1)}, x_c^{(n)}) + \psi_c^{(n)}(x_c^{(n)}), \tag{3.10}$$

and Φ is symmetric. Specifically, over a three label range of (α, L_F, β), the decomposition will look like this:

$$k_i \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} = -k \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix} + k \begin{pmatrix} 0 & \frac{1}{2} & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & \frac{1}{2} & 0 \end{pmatrix} + k \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix}. \tag{3.11}$$

To prove this in the general case, we have lemma 1.

Lemma 1. *The inter-layer pairwise functions*

$$\phi_{ic}^{(n)}(x_i^{(n-1)}, x_c^{(n)}) = k_c^i \begin{cases} 0 & \text{if } x_c^{(n)} = L_F \text{ or } x_c^{(n)} = x_i^{(n-1)} \\ 1 & \text{if } x_c^{(n)} = l \in \mathcal{L} \text{ and } x_i^{(n-1)} \neq l \end{cases} \tag{3.12}$$

of (3.9) can be written as:

$$\phi_{ic}^{(n)}(x_i^{(n-1)}, x_c^{(n)}) \;=\; \psi_i^{(n-1)}(x_i^{(n-1)}) + \psi_c^{(n)}(x_c^{(n)})\Phi_{ic}^{(n)}(x_i^{(n-1)}, x_c^{(n)}), \tag{3.13}$$

where

$$\Phi_{ic}^{(n)}(x_i^{(n-1)}, x_c^{(n)}) = k_c^i \begin{cases} 0 & \text{if } x_i^{(n-1)} = x_c^{(n)} \\ \frac{1}{2} & \text{if } (x_i^{(n-1)} = L_F \text{ or } x_c^{(n)} = L_F) \\ & \text{and } x_i^{(n-1)} \neq x_c^{(n)} \\ 1 & \text{otherwise,} \end{cases} \tag{3.14}$$

and

$$\psi_c^{(n)}(x_c^{(n)}) = k_c^i \begin{cases} 0 & \text{if } x_c^{(n)} \in \mathcal{L} \\ -\frac{1}{2} & \text{otherwise,} \end{cases} \tag{3.15}$$

$$\psi_i^{(n-1)}(x_i^{(n-1)}) = k_c^i \begin{cases} 0 & \text{if } x_i^{(n-1)} \in \mathcal{L} \\ \frac{1}{2} & \text{otherwise.} \end{cases} \tag{3.16}$$

Proof *Consider a clique containing only one variable, the general case will follow by induction. Note that if no variables take state L_F the costs are invariant to reparameterisation. This leaves three cases:*

$$\begin{array}{|r|} \hline \mathbf{x_c^{(n)} = L_F, x_i^{(n-1)}} \in \mathcal{L} \\ \psi_c(x_c^{(n)}) + \psi_{ic}(x_c^{(n)}, x_i^{(n-1)}) = -k/2 + k/2 = 0 \\ \hline \mathbf{x_c^{(n)}} \in \mathcal{L}, \mathbf{x_i^{(n-1)} = L_F} \\ \psi_i(x_i^{(n-1)}) + \psi_{ic}(x_i^{(n-1)}, x_c^{(n)}) = k/2 + k/2 = k \\ \hline \mathbf{x_c^{(n)} = L_F, x_i^{(n-1)} = L_F} \\ \psi_i(x_i^{(n-1)}) + \psi_{ic}(x_i^{(n-1)}, x_c^{(n)}) + \psi_c(x_c^{(n)}) = \frac{k-k}{2} = 0 \\ \hline \end{array} \tag{3.17}$$

□

Bounded Higher Order Inference We now prove bounds for α-expansion over an AHN.

1. The pairwise function of lemma 1, is positive definite, symmetric, and satisfies the triangle inequality

   $$\psi_{a,b}(x, z) \leq \psi_{a,b}(x, y) + \psi_{a,b}(y, z) \; \forall x, y, z \in \mathcal{L} \cup \{L_F\}. \tag{3.18}$$

   Hence it is a metric, and the algorithms $\alpha\,\beta$ swap and α-expansion can be used to minimise it.

2. By the work of Boykov et al. (2001), the α-expansion algorithm is guaranteed to find a solution within a factor of $2 \max\left(2, \max_{E \in \mathcal{E}^1} \frac{\max_{x_i, x_j \in \mathcal{L}} \psi_E(x_i, x_j)}{\min_{x_i, x_j \in \mathcal{L}} \psi_E(x_i, x_j)}\right)$ (i.e. 4 where the potentials defined over the base layer of hierarchy take the form of a Potts model) of the global optimum.

3. The following two properties hold:

$$\min_{\mathbf{x}^{(1)}} E(\mathbf{x}^{(1)}) = \min_{\mathbf{x}^{(1)}, \mathbf{x}^a} \left(E'(\mathbf{x}^{(1)}) + E^a(\mathbf{x}^{(1)}, \mathbf{x}^a)\right), \tag{3.19}$$

$$E(\mathbf{x}^{(1)}) \leq E'(\mathbf{x}^{(1)}) + E^a(\mathbf{x}^{(1)}, \mathbf{x}^a) \ \ \forall \mathbf{x}^a. \tag{3.20}$$

Hence, if there exists a labelling $(\mathbf{x}', \mathbf{x}^*)$ such that

$$E'(\mathbf{x}') + E^a(\mathbf{x}', \mathbf{x}^*) \leq k \min_{\mathbf{x}^{(1)}, \mathbf{x}^a} \Big(E'(\mathbf{x}^{(1)}) + E^a(\mathbf{x}^{(1)}, \mathbf{x}^a)\Big). \tag{3.21}$$

then

$$E(\mathbf{x}') \leq k \min_{\mathbf{x}^{(1)}} E(\mathbf{x}^{(1)}). \tag{3.22}$$

Consequently, the bound is preserved in the transformation that maps the pairwise energy back to its higher order form. □

By way of comparison, the work of Gould et al. (2009a) provides a bound of $2|c|$ for the higher order potentials of the strict P^n model (Kohli et al., 2007), where c is the largest clique in the network. Using their approach, no bounds are possible for the general class of Robust P^n models or for associative hierarchical networks.

The moves of our new range-move algorithm (see next section) strictly contain those considered by α-expansion and thus our approach automatically inherits the above approximation bound.

3.5 Novel Moves and Transformational Optimality

In this section we propose a novel graph cut based move making algorithm for minimising the hierarchical pairwise energy function defined in the previous section.

Let us consider a generalisation of the swap and expansion moves proposed in Boykov et al. (2001). In a standard swap move, the set of all moves considered is those in which a subset of the variables currently taking label α or β change labels to either β or α. In our range-swap the moves considered allow any variables taking labels α,L_F or β to change their state to any of α,L_F or β. Similarly, while a normal α expansion move allows any variable to change to some state α, our range expansion allows any variable to change to states α or L_F.

This approach can be seen as a variant on the ordered range moves proposed in Veksler (2007); Kumar and Torr (2008b), however while these works require that an ordering of the labels $\{l_1, l_2, \dots, l_n\}$ exist such that moves over the range $\{l_i, l_{i+1} \dots l_{i+j}\}$ are convex for some $j \geq 2$ and for all $0 < i \leq n - j$, our range moves function despite no such ordering existing.

We now show that the problem of finding the optimal swap move can be solved exactly in polynomial time. Consider a label mapping function $f_{\alpha,\beta}$: $\mathcal{L} \rightarrow \{1, 2, 3\}$ defined over the set $\{\alpha, L_F, \beta\}$ that maps α to 1, L_F to 2 and β to 3. Given this function, it is easy to see that the reparameterised inter-layer potential[2] $\Phi^{(n)}_{ic}(x^{(n-1)}_i, x^{(n)}_c)$ defined in lemma 1 can be written as a convex function of $f_{\alpha,\beta}(x^{(n-1)}_i) - f_{\alpha,\beta}(x^{(n)}_c)$ over the range α, L_F, β. Hence, we can use the Ishikawa construct (Ishikawa, 2003) to minimise the swap move energy to find the optimal move. A similar proof can be constructed for the range-expansion

[2] Exactly these reparameterised potentials, over this ordered range, are illustrated in matrix form in (3.11).

move described above.

The above defined move algorithm gives improved solutions for the hierarchical energy function used for formulating the object segmentation problem. We can improve further upon this algorithm. Our novel construction for computing the optimal moves explained in the following section, is based upon the original energy function (before reparameterisation) and has a strong transformational optimality property. We first describe the construction of a three label range move over the hierarchical network, and then show in section 3.5.2 that under a set of reasonable assumptions, our methods are equivalent to a swap or expansion move that exactly minimises the equivalent higher order energy defined over the base variables $E(\mathbf{x}^{(1)})$ of the hierarchical network (as defined in (3.8)).

3.5.1 Construction of the Range Move

We now explain the construction of the submodular quadratic pseudo Boolean (QPB) move function for range expansion. The construction of the swap based move function can be derived from this range move.

In essence, we demonstrate that the cost function of (3.12) over the range $x_c \in \{\beta, L_F, \alpha\}, x_i \in \{\delta, L_F, \alpha\}$ where β may or may not equal δ is expressible as a submodular QPB potential. To do this, we create a QPB function defined on 4 variables c_1, c_2, i_1 and i_2. We associate the states $i_1 = 1, i_2 = 1$ with x_i taking state α, $i_1 = 0, i_2 = 0$ with the current state of $x_i = \delta$, and $i_1 = 1, i_2 = 0$ with state L_F. We prohibit the state $i_1 = 0, i_2 = 1$ by incorporating the pairwise term $\infty(1 - i_1)i_2$ which assigns an infinite cost to the state $i_1 = 0, i_2 = 1$, and do the same respectively with x_c and c_1 and c_2. To simplify the resulting equation, we write I instead of $\Delta(\beta \neq \delta)$, and k as a substitute for $\psi_{i,c}(\alpha, \delta) = k_c^i$ following

(3.12) then:

$$\psi_{i,c}(x_i, x_c) = k\left((1-I)c_2(1-i_2) + Ic_2 + (1-c_1)i_1\right) \tag{3.23}$$

over the range $x_c \in \{\beta, L_F, \alpha\}, x_i \in \{\delta, L_F, \alpha\}$.

The proof follows from inspection of the function. Below we tabulate the possible inputs and outputs to allow easy comparison.

	$x_i{=}\delta$ $i_1{=}1, i_2{=}1$	$x_i{=}L_F$ $i_1{=}1, i_2{=}0$	$x_i{=}\alpha$ $i_1{=}0, i_2{=}0$
$x_c{=}\beta$ $c_1{=}1, c_2{=}1$	kI	$k\left((1-I)+I\right)$	$k\left((1-I)+I\right)$
$x_c{=}L_F$ $c_1{=}1, c_2{=}0$	0	0	0
$x_c{=}\alpha$ $c_1{=}0, c_2{=}0$	k	k	0

(3.24)

Note that $c_2 = 1$ if and only if $x_c = \beta$ while $c_1 = 0$ if and only if $c = \alpha$. If $x_c = L_F$ then $c_2 = 0$ and $c_1 = 1$ and the cost is always 0. If $x_c = \alpha$ the first two terms take cost 0, and the third term has a cost of k associated with it unless $x_i = \alpha$. Similarly, if $x_c = \beta$ there is a cost of k associated with it, unless x_i also takes label β. □

3.5.2 Optimality

Note that both variants of unordered range moves are guaranteed to find the global optimum if the label space of $\mathbf{x}^{(1)}$ contains only two states. This is not the case for the standard forms of α expansion or $\alpha\beta$ swap as auxiliary variables may take one of three states.

Transformational optimality Consider an energy function defined over the variables $\mathbf{x} = \{\mathbf{x}^{(h)}, h \in \{1, 2, \ldots, H\}\}$ of a hierarchy with H levels. We call a move making algorithm *transformationally optimal* if and only if any proposed move $(\mathbf{x}^*, \mathbf{x}^a)$ satisfies the property:

$$E(\mathbf{x}^*) = E'(\mathbf{x}^*) + E^a(\mathbf{x}^*, \mathbf{x}^a). \tag{3.25}$$

where $\mathbf{x}^a = \bigcup_{h \in 2,\dots,H} \mathbf{x}_*^{(h)}$ represents the labelling of all auxiliary variables in the hierarchy. Note that any move proposed by transformationally optimal algorithms minimises the original higher order energy (3.8). We now show that when applied to hierarchical networks, the *range* moves are transformationally optimal.

Move Optimality To guarantee transformational optimality we need to constrain the set of higher order potentials. Consider a clique c with an associated auxiliary variable $x_c^{(i)}$. Let $\mathbf{x}_l$ be a labelling such that $x_c^{(i)} = l \in \mathcal{L}$ and $\mathbf{x}_{L_F}$ be a labelling that differs from it only in that the variable $x_c^{(i)}$ takes label L_F. We say a clique potential is *hierarchically consistent* only if it satisfies the constraint:

$$E(\mathbf{x}_l) \geq E(\mathbf{x}_{L_F}) \implies \frac{\sum_{i \in c} k_c^i \Delta(x_i = l)}{\sum_{i \in c} k_c^i} > 0.5. \tag{3.26}$$

The property of hierarchical consistency is also required in computer vision for the cost associated with the hierarchy to remain meaningful. The labelling of an auxiliary variable within the hierarchy should be reflected in the state of the clique associated with it. If an energy is not hierarchically consistent, it is possible that the optimal labelling of regions of the hierarchy will not reflect the labelling of the base layer.

To understand why this consistency is important, we consider a case where this is violated. Consider a simple energy function consisting of a base layer of 10 pixels $\mathbf{x}^{(1)}$ and only one clique, with associated auxiliary variable x_c, defined over the base layer. We assume that all the pixels in the base layer wish to belong to one class *sheep* while the higher order potential defined over the clique expresses a preference for class *cow*.

More formally we set:

$$\psi_i(x_i) = \begin{cases} 2 & \text{if } x_i = \textit{sheep} \\ 0 & \text{if } x_i = \textit{cow} \end{cases} \quad \forall x_i \in \mathbf{x}^{(1)} \tag{3.27}$$

$$\phi_c(x_c) = \begin{cases} 0 & \text{if } x_c = \textit{sheep} \\ 20 & \text{if } x_c = \textit{cow} \\ 20 & \text{if } x_c = L_F \end{cases} \tag{3.28}$$

And we define the pairwise terms between the clique variables as

$$\phi_{c,i}(x_c, x_i) = \begin{cases} 1 & \text{if } x_c \neq L_F \wedge x_c \neq x_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.29}$$

For simplicity, we set all pairwise terms within the base layer to 0, and disregard them. Then a minimal labelling of the solution occurs when, $x_i = \textit{cow}\, \forall x_i \in \mathbf{x}^{(1)}$ and $x_c = \textit{sheep}$. This labelling is incoherent, insomuch as we believe at the base scale that a region is *cow*, and at a coarser scale that the same region is *sheep*. Our requirement of *hierarchical consistency* prohibits such solutions by insisting that the minimal cost labelling of higher levels in the hierarchy, give an fixed labelling of the base layer, must correspond to either the dominant label in base layer, or to the label L_F.

The constraint (3.26) is enforced by construction, weighting the relative magnitude of $\psi_i(l)$ and $\psi_{i,j}(b_j, x_c^{(i)})$ to guarantee that:

$$\psi_i(l) + \sum_{j \in N_i/c} \max_{b_j \in \mathcal{L} \cup \{L_f\}} \psi_{i,j}(b_j, x_c^{(i)}) < 0.5 \sum_{i \in c} k_i \forall l \in \mathcal{L}. \tag{3.30}$$

If this holds, in the degenerate case where there are only two levels in the hierarchy, and no pairwise connections between the auxiliary variables, our network is exactly equivalent to the P^n model.

At most one $l \in \mathcal{L}$ at a time can satisfy (3.26), assuming the hierarchy is consistent. Given a labelling for the base layer of the hierarchy $\mathbf{x}^{(1)}$, an optimal labelling for an auxiliary variable in $\mathbf{x}^{(2)}$ associated with some clique must be one of two labels: L_F and some $l \in \mathcal{L}$. By induction, the choice of labelling of any clique in $\mathbf{x}^{(j)} : j \geq 2$ must also be a decision between at most two labels: L_F and some $l \in \mathcal{L}$.

3.5.3 Transformational Optimality under Unordered Range Moves

Swap range moves

Swap based optimality requires an additional constraint to that of (3.26), namely that there are no pairwise connections between variables in the same level of the hierarchy, except in the base layer. From (3.7) if an auxiliary variable x_c may take label γ or L_F, and one of its children $x_i|i \in c$ take label δ or L_F, the cost associated with assigning label γ or L_F to x_c is independent of the label of x_i with respect to a given move.

Under a swap move, a clique currently taking label $\delta \notin \{\alpha, \beta\}$ will continue to do so. This follows from (2.2) as the cost associated with taking label δ is only dependent upon the weighted average of child variables taking state δ, and this remains constant. Hence the only clique variables that may have a new optimal labelling under the swap are those currently taking state α, L_F or β, and these can only transform to one of the states α, L_F or β. As the range moves map exactly this set of transformations, the move proposed must be transformationally optimal, and consequently the best possible $\alpha\beta$ swap over the energy (3.1).

Expansion Moves

In the case of a range-expansion move, we can maintain transformational optimality while incorporating pairwise connections into the hierarchy — provided condition (3.26) holds, and the energy can be exactly represented in our submodular moves.

In order for this to be the case, the pairwise connections must be both convex over any range α, L_F, β and a metric. The only potentials that satisfy this are linear over the ordering $\alpha, L_F, \beta \ \forall \alpha, \beta$. Hence all pairwise connections must be of the form:

$$\psi_{i,j}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ \lambda/2 & \text{if } x_i = L_F \text{ or } x_j = L_F \text{ and } x_i \neq x_j \\ \lambda & \text{otherwise.} \end{cases} \tag{3.31}$$

where $\lambda \in \mathbb{R}_0^+$. By lemma 1, it can be readily seen that the connections in the hierarchical network are a constrained variant of this form.

A similar argument to that of the optimality of $\alpha\beta$ swap can be made for α-expansion. As the label α is 'pushed' out across the base layer, the optimal labelling of some $x^{(n)}$ where $n \geq 2$ must either remain constant or transition to one of the labels L_F or α. Again, the range moves map exactly this set of transforms and the suggested move is both transformationally optimal, and the best expansion of label α over the higher order energy of (3.8).

3.6 Experiments

We evaluate α-expansion, $\alpha\beta$ swap, TRW-S, Belief Propagation, Iterated Conditional Modes, and both the expansion and swap based variants of our unordered range moves on the problem of object class segmentation over the MSRC data-

Method	Best	$E(\text{meth}) - E(\text{min})$	$\frac{E(\text{meth})}{E(\text{min})}$	Time
Range-exp	265	75	1.000	6.1s
Range-swap	137	9034	1.059	20s
α-expansion	109	256	1.002	6.3s
$\alpha\beta$ swap	42	9922	1.060	42s
TRW-S	12	38549	1.239	500s
BP	6	13456	1.081	120s
ICM	5	45955	1.274	25s

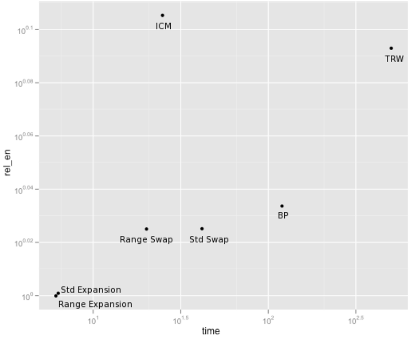

Figure 3.1: *Comparison of methods on 295 testing images. From left to right the columns show the number of times they achieved the best energy (including ties), the average difference (E(method) $-$ E(min)), the average ratio (E(method)/E(min)) and the average time taken. All three approaches proposed in this chapter: α-expansion under the reparameterisation of section 3.5, and the transformationally optimal range expansion and swap significantly outperformed existing inference methods both in speed and accuracy. See figures 3.2 3.3 for individual examples.*

set (Shotton et al., 2006), in which each pixel within an image must be assigned a label representing its class, such as grass, water, boat or cow. We express the problem as a three layer hierarchy. Each pixel is represented by a random variable of the base layer. The second layer is formed by performing multiple unsupervised segmentations over the image, and associating one auxiliary variable with each segment - note that this use of several hierarchies results in overlapping segments. The children of each of these variables in $x^{(2)}$ are the variables contained within the segment, and pairwise connections are formed between adjacent segments. The third layer is formed in the same manner as the second layer by clustering the image segments. Further details are given in Ladicky et al. (2009).

We tested each algorithm on 295 test images, with an average of 70,000 pixels/variables in the base layer and up to 30,000 variables in a clique, and ran them either until convergence, or for a maximum of 500 iterations. In the table in figure 3.1 we compare the final energies obtained by each algorithm, showing the number of times they achieved an energy lower than or equal to all other methods, the average difference $E(\text{method}) - E(\text{min})$ and average ratio $E(\text{method})/E(\text{min})$.

Empirically, the message passing algorithms TRW-S and BP appear ill-suited to inference over these dense hierarchical networks. In comparison to the graph cut based move making algorithms, they had higher resulting energy, higher memory usage, and exhibited slower convergence.

While it may appear unreasonable to test message passing approaches on hierarchical energies when higher order formulations such as (Komodakis and Paragios, 2009; Potetz and Lee, 2008) exist, we note that for the simplest hierarchy that contains only one additional layer of nodes and no pairwise connections in this second layer, higher order and hierarchical message-passing approaches will be equivalent, as inference over the trees that represent higher order potentials is exact. Similar relative performance by message passing schemes was observed in these cases. Further, application of such approaches to the general form of (3.8) would require the computation of the exact min-marginals of $E^{(2)}$, a difficult problem in itself.

In all tested images both α-expansion variants outperformed TRW-S, BP and ICM. These later methods only obtained minimal cost labellings in images in which the optimal solution found contained only one label i.e. they were entirely labelled as grass or water. The comparison also shows that unordered range move variants usually outperform vanilla move making algorithms. The higher number of minimal labellings found by the range-move variant of $\alpha\beta$ swap in comparison to those of vanilla α-expansion can be explained by the large number of images in which two labels strongly dominate, as unlike standard α-expansion both range move algorithms are guaranteed to find a global optimum of such a two label subproblem (see section 3.5.2). The typical behaviour of all methods alongside the lower bound of TRW-S can be seen in figure 3.1 and further, alongside qualitative results, in figures 3.2, 3.3, and 3.4.

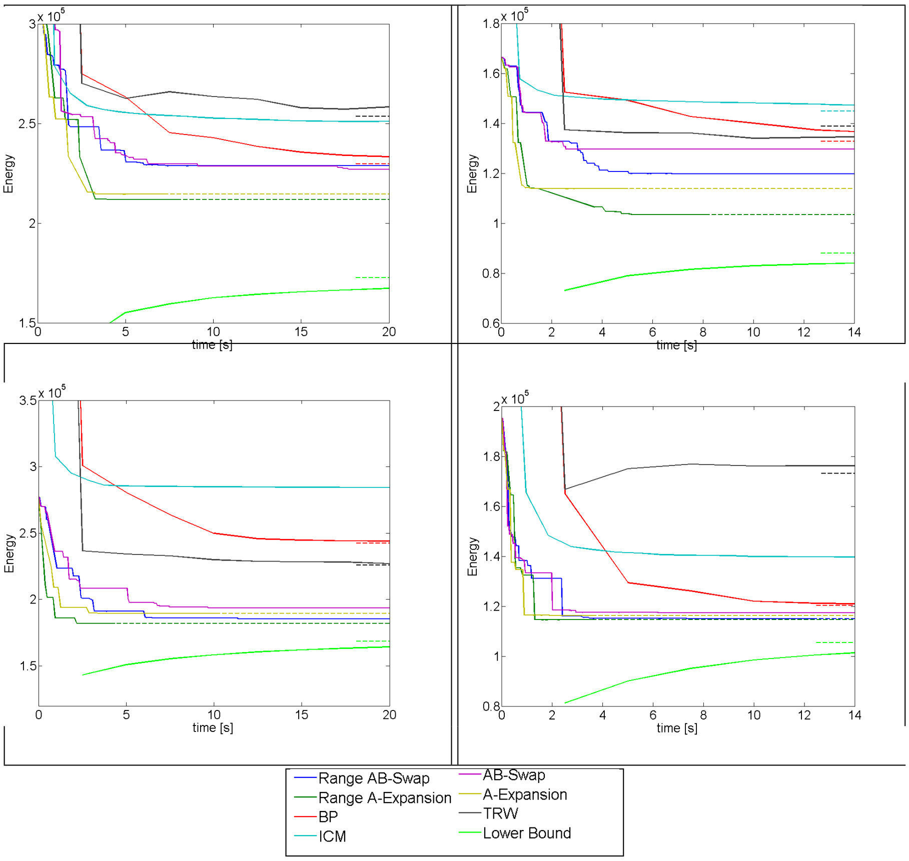

Figure 3.2: **Best Viewed in Colour.** *This figure shows additional quantitative results taken from the* MSRC *data set (Shotton et al., 2006). Dashed lines indicate the final converged solution. The slow convergence and poor solutions found by* TRW *and* BP *are to be expected given the large number of cycles present in the graph. Of the remaining move making schemes, the relatively weak performance of $\alpha\beta$-swap and* ICM *is in line with the restricted space of moves available to them. While the three methods derived in this chapter significantly outperform all other approaches, range α expansion reliably dominates. Over the page, qualitative results are shown.*

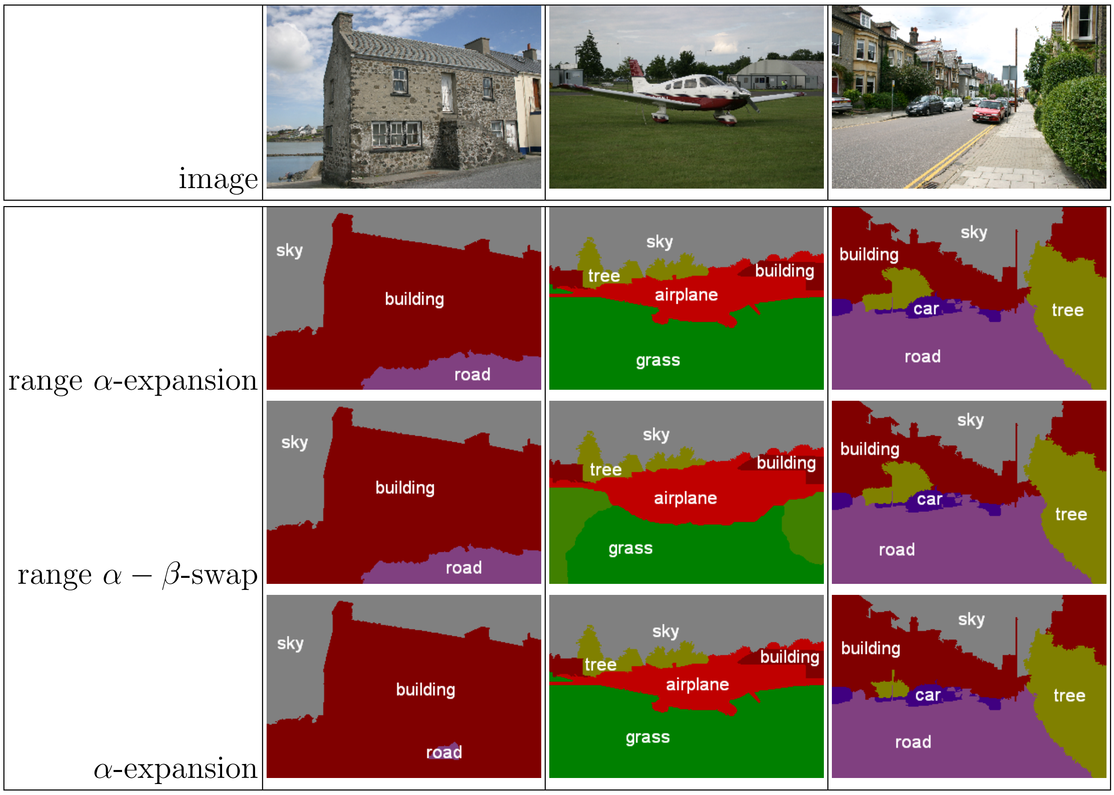

Figure 3.3: **Best Viewed in Colour.** *Qualitative results on typical images from the* MSRC *data set (Shotton et al., 2006). The improvements provided by the three approaches proposed in this chapter can be seen above, and the other approaches over the page. As with the quantitative results, the new approaches are a significant improvement over the old methods, both in correct object boundaries, and in the elimination of small classes from the image. As before, range α expansion provides a consistent improvement over the other two approaches.*

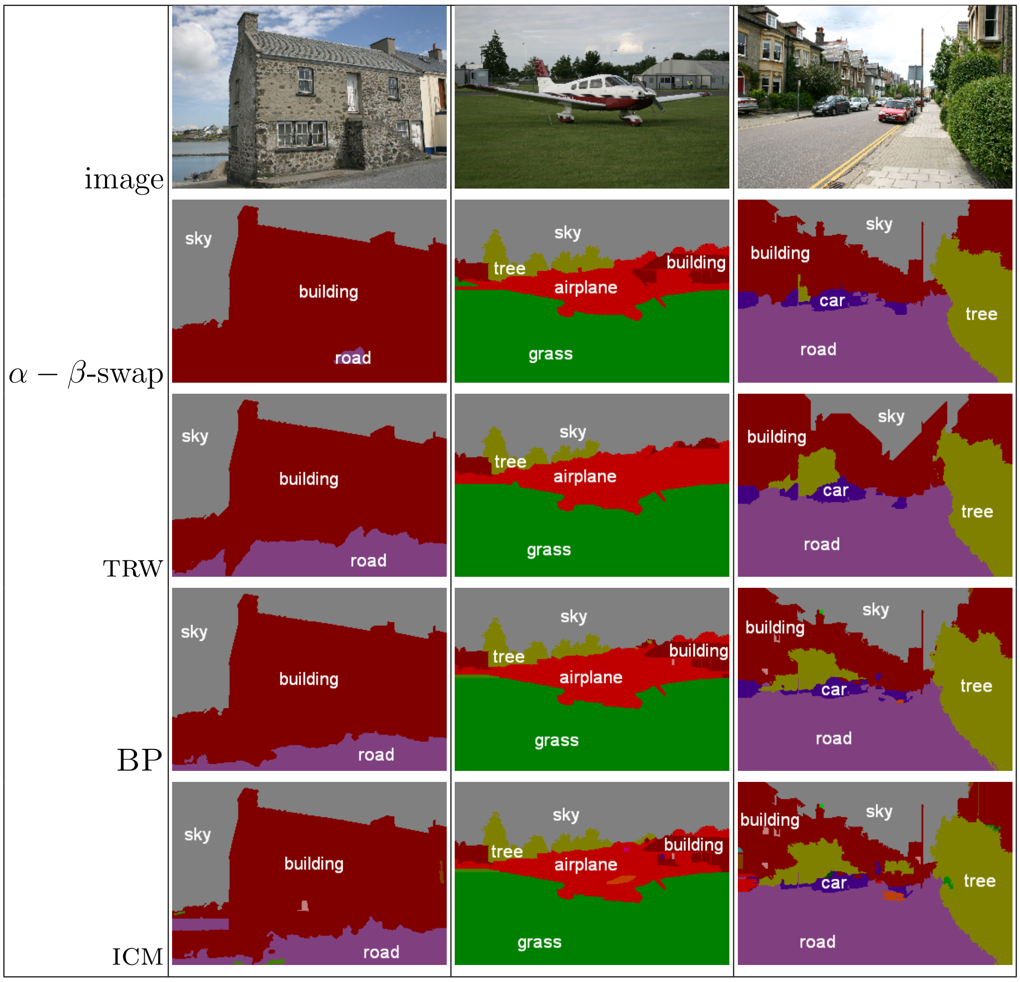

Figure 3.4: **Best Viewed in Colour.** *Qualitative results on typical images from the* MSRC *data set (Shotton et al., 2006) using pre-existing approaches.Please compare with figure 3.3, to see the benefits of our approach.*

3.7 Conclusion

This chapter shows that higher order AMNs are intimately related to pairwise hierarchical networks. This observation allowed us to characterise higher order potentials which can be solved under a novel reparameterisation using conventional move making expansion and swap algorithms, and derive bounds for such approaches. We also gave a new transformationally optimal family of algorithms for performing efficient inference in higher order AMN that inherits such bounds.

We have demonstrated the usefulness of our algorithms on the problem of object class segmentation where they have been shown to outperform state of the art approaches over challenging data sets (Ladicky et al., 2009) both in speed and accuracy. We believe that similar improvements can be achieved for many other higher order labelling problems both in computer vision and machine learning in general.

Chapter 4

Inference with Co-occurrence Statistics

4.1 Overview

The Markov and Conditional random fields (CRFs) used in computer vision typically model only local interactions between variables, as this is generally thought to be the only case that is computationally tractable. In this paper we consider a class of global potentials defined over all variables in the CRF. We show how they can be readily optimised using standard graph cut algorithms at little extra expense compared to a standard pairwise field.

This result can be directly used for the problem of *class based image segmentation* which has seen increasing recent interest within computer vision. Here the aim is to assign a label to each pixel of a given image from a set of possible object classes. Typically these methods use random fields to model local interactions between pixels or super-pixels. One of the cues that helps recognition is global *object co-occurrence statistics*, a measure of which classes (such as chair or motorbike) are likely to occur in the same image together. There have been several approaches proposed to exploit this property, but all of them suffer from different limitations and typically carry a high computational cost, preventing their application on large images. We find that the new model we propose produces an improvement in the labelling compared to just using a pairwise model.

4.2 Introduction

Class based image segmentation is a highly active area of computer vision research as shown by a spate of recent publications (Heitz, 2008; Rabinovich et al., 2007; Shotton et al., 2006; Torralba et al., 2003; Yang et al., 2007). In this problem, every pixel of the image is assigned a choice of object class label, such as grass, person, or dining table. Formulating this problem as a likelihood, in order to perform inference, is a difficult problem, as the cost or energy associated with any labelling of the image should take into account a variety of cues at different scales. A good labelling should take account of: low-level cues such as colour or texture (Shotton et al., 2006), that govern the labelling of single pixels; mid-level cues such as region continuity, symmetry (Ren et al., 2005) or shape (Borenstein and Malik, 2006) that govern the assignment of regions within the image; and high-level statistics that encode inter-object relationships, such as which objects can occur together in a scene. This combination of cues makes for a multi-scale cost function that is difficult to optimise.

Current state of the art low-level approaches typically follow the methodology proposed in *Texton-boost* (Shotton et al., 2006), in which weakly predictive features such as colour, location, and texton response are used to learn a classifier which provides costs for a single pixel taking a particular label. These costs are combined in a contrast sensitive Conditional Random Field CRF (Lafferty et al., 2001).

The majority of mid-level inference schemes (Russell et al., 2006; Larlus and Jurie, 2008) do not consider pixels directly, rather they assume that the image has been segmented into super-pixels (Comaniciu and Meer, 2002; Felzenszwalb and Huttenlocher, 2004; Shi and Malik, 2000). A labelling problem is then defined over the set of regions. A significant disadvantage of such approaches is that mistakes in the initial over-segmentation, in which regions span multiple object classes, cannot be recovered from.

These approaches can be improved by the inclusion of costs based on high level statistics, including object class co-occurrence, which capture knowledge of scene semantics that humans often take for granted: for example the knowledge that cows and crocodiles are not kept together and less likely to appear in the same image; or that motorbikes are unlikely to occur near televisions. In this paper we consider object class co-occurrence to be a measure of how likely it is for a given set of object classes to occur together in an image. They can also be used to encode scene specific information such as the facts that computer monitors and stationery are more likely to occur in offices, or that trees and grass occur outside. The use of such costs can help prevent some of the most glaring failures in object class segmentation, such as the labelling of a boat surrounded by water mislabelled as a book.

As well as penalising strange combinations of objects appearing in an image, co-occurrence potentials can also be used to impose a minimum description length (MDL) prior, that encourages a parsimonious description of an image using fewer labels. As discussed eloquently in the recent work (Choi et al., 2010), the need for a bias towards parsimony becomes increasingly important as the number of classes to be considered increases. Figure 4.1 illustrates the importance of co-occurrence statistics in image labelling.

The promise of co-occurrence statistics has not been ignored by the vision community. Rabinovich et al. (2007) proposed the integration of such co-occurrence costs that characterise the relationship between two classes. Similarly Torralba et al. (2003) proposed scene-based costs that penalised the existence of particular

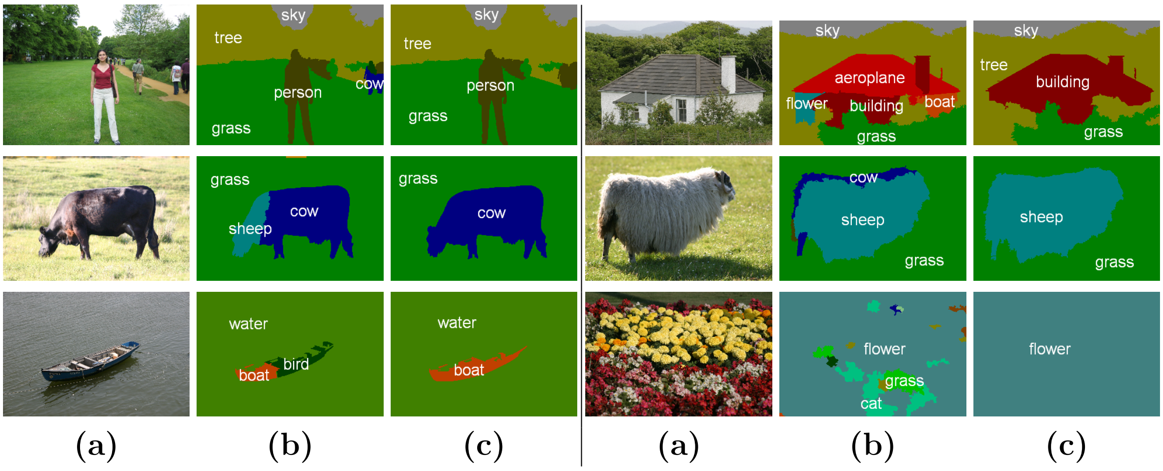

Figure 4.1: **Best viewed in colour:** *Qualitative results of object co-occurrence statistics.* **(a)** *Typical images taken from the MSRC data set (Shotton et al., 2006);* **(b)** *A labelling based upon a pixel based random field model (Ladicky et al., 2009) that does not take into account co-occurrence;* **(c)** *A labelling of the same model using co-occurrence statistics. The use of co-occurrence statistics to guide the segmentation results in a labelling that is more parsimonious and more likely to be correct. These co-occurrence statistics suppress the appearance of small unexpected classes in the labelling.* **Top left:** *a mistaken hypothesis of a cow is suppressed* **Top right:** *Many small classes are suppressed in the image of a building. Note that the use of co-occurrence typically changes labels, but does not alter silhouettes.*

classes in a context dependent manner. We shall discuss these approaches, and some problems with them in the next section.

4.3 CRFs and Co-occurrence

To model object class co-occurrence statistics a new term $K(\mathbf{x})$ is added to the cost function of 2 (2.8) :

$$E(\mathbf{x}) = \sum \psi_c(\mathbf{x}_c) + K(\mathbf{x}). \tag{4.1}$$

The question naturally arises as to what form an energy involving co-occurrence terms should take. We now list a set of desiderata that we believe are intuitive for any co-occurrence cost.

(i) Global Energy: We would like a formulation of co-occurrence that allows us to estimate the segmentation using all the data directly, by minimising a *single* cost function of the form (4.1). Rather than any sort of two stage process in which a hard decision is made of which objects are present in the scene *a priori* as in (Torralba et al., 2003).

(ii) Invariance: The co-occurrence cost should depend only on the labels present in an image, it should be invariant to the number and location of pixels that object occupies. To reuse an example from (Toyoda and Hasegawa, 2008), the surprise at seeing a polar bear in a street scene should not not vary with the number of pixels that represent the bear in the image.

(iii) Efficiency: Inference should be tractable, *i.e.* the use of co-occurrence should not be the bottle-neck preventing inference. As the memory requirements of any conventional inference algorithm (Szeliski et al., 2006) is typically $O(|\mathcal{V}|)$ for vision problems, the memory requirements of a formulation incorporating co-occurrence potentials should also be $O(|\mathcal{V}|)$.

(iv) Parsimony: The cost should follow the principle of parsimony in the following way : if several solutions are almost equally likely then the solution that can describe the image using the fewest distinct labels should be chosen. Whilst this might not seem important when classifying pixels into a few classes, as the set of putative labels for an image increases the chance of speckle noise due to mis-classification will increase unless a parsimonious solution is encouraged.

While these properties seem uncontroversial, no prior work exhibits property *(ii)*. Similarly, no approaches satisfy properties *(i)* and *(iii)* simultaneously. In order to satisfy condition *(ii)* the co-occurrence cost $K(\mathbf{x})$ defined over $\mathbf{x}$ must be a function defined on the set $L(\mathbf{x}) = \{l \in \mathcal{L} : \exists x_i = l\}$ of labels used in the labelling $\mathbf{x}$; this guarantees invariance to the size of an object:

$$K(\mathbf{x}) = C(L(\mathbf{x})) \tag{4.2}$$

Adding the co-occurrence term to the CRF cost function (1.22), we have:

$$E(\mathbf{x}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) + C(L(\mathbf{x})). \tag{4.3}$$

To satisfy the parsimony condition *(iv)* potentials must act to penalise the unexpected appearance of combinations of labels in a labelling. This observation can be formalised as the statement that the cost $C(L)$ is monotonically increasing with respect to the label set L *i.e.*:

$$L_1 \subset L_2 \implies C(L_1) \leq C(L_2). \tag{4.4}$$

The new potential $C(L(\mathbf{x}))$ can be seen as a particular higher order potential defined over a clique which includes the whole of $\mathcal{V}$, *i.e.* $\psi_{\mathcal{V}}(\mathbf{x})$.

4.3.1 Prior Work

There are two existing approaches to co-occurrence potentials, neither of which use potentials defined over a clique of size greater than two. The first makes an initial hard estimate of the type of scene, and updates the unary potentials associated with each pixel to encourage or discourage particular choices of label, on the basis of how likely they are to occur in the scene. The second approach models object co-occurrence as a pairwise potential between regions of the image.

Torralba et al. (2003) proposed the use of additional unary potentials to capture scene based occurrence priors. Their costs took the form:

$$K(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi(x_i). \tag{4.5}$$

While the complexity of inference over such potentials scales linearly with the size of the graph, they are prone to over counting costs, violating *(ii)*, and require an initial hard decision of scene type before inference, which violates *(i)*. As it encourages the appearance of all labels which are common to a scene, it does not necessarily encourage parsimony *(iv)*.

A similar approach was seen in the Pascal VOC2008 object segmentation challenge, where the best performing method, by (Csurka and Perronnin, 2008), worked in two stages. Initially the set of object labels present in the image was estimated, and in the second stage, a label from the estimated label set was assigned to each image pixel. As no cost function $K(\cdot)$ was proposed, it is open to debate if it satisfied *(ii)* or *(iv)*.

Rabinovich et al. (2007); Galleguillos et al. (2008), and independently Toyoda and Hasegawa (2008), proposed co-occurrence as a soft constraint that approximated $C(L(\mathbf{x}))$ as a pairwise cost defined over a *fully connected graph* that took

Method	Global energy	Invariance	Efficiency	Parsimony
	(i)	*(ii)*	*(iii)*	*(iv)*
Unary	✓	✗	✓	✗
Pairwise	✓	✗	✗	✓
Hard decisions	✗	—	✓	—
Our approach	✓	✓	✓	✓

Figure 4.2: *A comparison of the capabilities of existing image co-occurrence formulations (Unary (Torralba et al., 2003), Pairwise (Rabinovich et al., 2007; Galleguillos et al., 2008; Toyoda and Hasegawa, 2008), Hard decision (Csurka and Perronnin, 2008)) against our new approach. See section 4.3.1 for details.*

the form:

$$K(\mathbf{x}) = \sum_{i,j \in \mathcal{V}} \phi(x_i, x_j), \tag{4.6}$$

where ϕ was some potential which penalised labels that should not occur together in an image. Unlike our model (4.3) the penalty cost for the presence of pairs of labels, that rarely occur together, appearing in the same image grows with the number of random variables taking these labels, violating assumption *(ii)*. While this serves as a functional penalty that prevents the occurrence of many classes in the same labelling, it does not accurately model the co-occurrence costs we described earlier. The memory requirements of inference scales badly with the size of a fully connected graph. It grows with complexity $O(|\mathcal{V}|^2)$ rather than $O(|\mathcal{V}|)$ with the size of the graph, violating constraint *(iii)*. Providing the pairwise potentials are semi-metric[1] (Boykov et al., 2001), it does satisfy the parsimony condition *(iv)*.

To minimise these difficulties, previous approaches defined variables over segments rather than pixels. Such segment based methods work under the assumption that some segments share boundaries with objects in the image. This is not always the case, and this assumption may result in dramatic errors in the

[1] Recall from chapter 1, that this a prerequisite for using standard graph-cuts algorithms such as $\alpha\beta$-swap for inference.

labelling. The relationship between previous approaches and the desiderata can be seen in figure 4.2.

Two efficient schemes (Delong et al., 2010; Hoiem et al., 2007) have been proposed for the minimisation of the number of classes or objects present in a scene. While neither of them directly models class based co-occurrence relationships, their optimisation approaches satisfy the desiderata proposed in 2.1.

Hoiem et al. (2007), proposed a cost based on the number of objects in the scene, in which the presence of any instance of any object incurs a uniform penalty cost. For example, the presence of both a motorbike and a bus in a single image is penalised as much as the presence of two buses. Minimising the number of objects in a scene is a good method of encouraging consistent labellings, but does not capture any co-occurrence relationship between object classes.

If we view Hoiem's work as assigning a different label to every instance of an object class, their label set costs take the form:

$$C(L(\mathbf{x})) = k||L(\mathbf{x})|| \tag{4.7}$$

In a recent work, appearing at the same time as ours, Delong et al. (2010) also proposed the use of a soft cost over the number of labels present. In general their approach allowed the imposition of a penalty cost if any elements of certain subset are present in the image. They proposed using this cost to combine probabilistic formulations such as Akaike's Information Criterion, or the Bayesian Information Criterion (Torr, 1998) with efficient graph cut based label assignment. The general form of their costs is:

$$C(L(\mathbf{x})) = \sum_{L \subseteq \mathcal{L}} k_L \Delta(L(\mathbf{x}) \cap L \neq \emptyset) \tag{4.8}$$

Note that the costs of Delong et al. (2010) and Hoiem et al. (2007) both satisfy the inequality:

$$C(L_1 \cup L_2) \leq C(L_1) + C(L_2), \tag{4.9}$$

where L_1 and L_2 are any subsets of labels of $\mathcal{L}$. Consequently, their models are unable to express to co-occurrence potentials which say that certain classes, such as the previously mentioned example of polar bear and street, are less likely to occur together than in separate images.

4.4 Inference on global co-occurrence potentials

Consider the energy (4.3) defined in section 4.3. The inference problem becomes:

$$\begin{aligned} \mathbf{x}^* = \arg\min\nolimits_{\mathbf{x}\in\mathcal{L}^{|\mathcal{V}|}} \textstyle\sum_{c\in\mathcal{C}} \psi_c(\mathbf{x}_c) + C(L(\mathbf{x})) \\ \text{s.t. } \mathbf{x} \in \mathcal{L}^{|\mathcal{V}|},\ L(\mathbf{x}) = \{l \in \mathcal{L} : \exists x_i = l\}. \end{aligned} \tag{4.10}$$

In this section we show that the problem of minimising this energy can be reformulated as an integer program and solved using LP-relaxation. We will also show how it can be transformed into pairwise energy by adding one auxiliary variable connected to all pixels in the image and solved using Belief Propagation (Weiss and Freeman, 2001) or TRW-S (Kolmogorov, 2006). However, reparameterisation methods such as these perform badly on densely connected graphs (see previous chapter, section 3.6 and Kolmogorov and Rother (2006)). We show that the problem can be solved efficiently using move-making $\alpha\beta$-swap and α-expansion moves (Boykov et al., 2001), where the number of additional edges of the graph grows linearly with the number of variables in the graph. In contrast to (Rabinovich et al., 2007), these algorithms can be applied to large graphs with more than $200,000$ variables.

4.4.1 The Integer Programming formulation, and its Linear Relaxation

In the following two subsections, we make the simplifying assumption that the cost (1.22) is currently represented as a pairwise energy. The minimisation of the energy function (4.3) can be formulated as an Integer Program (IP) (Wainwright et al., 2002; Schlesinger, 1976). A vector $\mathbf{z}$ of binary indicator variables is used to represent the assignment of labels. $\mathbf{z}$ is composed of two sets of variables *(i)* $z_{i;a} \forall a \in \mathcal{L}, \forall i \in \mathcal{V}$, and *(ii)* $z_{ij;ab} \forall a, b \in \mathcal{L}, (i,j) \in \mathcal{E}$ where $\mathcal{E}$ is the set of edges, to represent the state of variables x_i, x_j such that:

$$z_{i;a} = \begin{cases} 1 & \text{if } x_i = a \\ 0 & \text{otherwise} \end{cases}, \qquad z_{ij;ab} = \begin{cases} 1 & \text{if } x_i = a \text{ and } x_j = b \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

In addition $\mathbf{z}$ contains a further set z_L, indicator variables that show which subset of labels $L(\mathbf{x})$ is used in the assignment. There are $2^{|\mathcal{L}|}$ such variables in total, one variable z_L for every $L \subseteq \mathcal{L}$. We write:

$$z_L = \begin{cases} 1 & \text{if } L = L(\mathbf{x}) \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

Thus, $\mathbf{z}$ is a binary vector of length $|\mathcal{V}| \cdot |\mathcal{L}| + |\mathcal{E}| \cdot |\mathcal{L}|^2 + 2^{|\mathcal{L}|}$. The resulting IP can be written as:

$$\min_{\mathbf{z}} \sum_{i \in \mathcal{V}, a \in \mathcal{L}} \psi_i(a) z_{i;a} + \sum_{(i,j) \in \mathcal{E}, a, b \in \mathcal{L}} \psi_{i,j}(a,b) z_{ij;ab} + \sum_{L \subseteq \mathcal{L}} C(L) z_L \tag{4.13}$$

such that:

$$\sum_a z_{ij;ab} = z_{j;b}, \qquad \forall (i,j) \in \mathcal{E}, b \in \mathcal{L}, \quad (4.14)$$

$$\sum_b z_{ij;ab} = z_{i;a}, \qquad \forall (i,j) \in \mathcal{E}, a \in \mathcal{L}, \quad (4.15)$$

$$\sum_a z_{i;a} = 1, \qquad \forall i \in \mathcal{V}, \quad (4.16)$$

$$\sum_{L \ni a} z_L \geq z_{i;a}, \qquad \forall i \in \mathcal{V}, a \in \mathcal{L}, L \subseteq \mathcal{L} \quad (4.17)$$

$$\sum_{L \in \mathcal{L}} z_L = 1 \qquad (4.18)$$

$$z_{i;a}, z_{ij;ab}, z_L \in \{0,1\} \qquad \forall i \in \mathcal{V}, \forall (i,j) \in \mathcal{E}, \forall a, b \in \mathcal{L}, \forall L \subseteq \mathcal{L}. \quad (4.19)$$

The marginal consistency and uniqueness constraints (4.14 - 4.16) are well-known and used in the standard IP formulation of the labelling problem (Komodakis et al., 2007; Kumar and Torr, 2008a; Wainwright et al., 2005; Werner, 2005). To enforce the consistency between labelling and the label set indicator variables z_L (4.12), two new properties which we refer to as “inclusion” and “exclusion” properties must be satisfied. The exclusion property which ensures that if $z_L = 1$, no variable takes a label not present in L, is enforced by the exclusion constraints (4.17). While, the inclusion property guarantees that if $z_L = 1$, then for each label $l \in L$ there exists at least one variable $z_{i;l}$ such that $z_{i;l} = 1$, is enforced by parsimony. To see why this is the case, consider a contra-positive solution where there is a label $l \in L$ not present in the solution. In this case, the solution $\mathbf{z}$ altered by $z_L = 0$ and $z_{L \setminus \{l\}} = 1$ would also satisfy all constraints (4.25 - 4.17) and due to the parsimony property would have the same or lower cost function 4.13). Thus, there exists a global optimum satisfying $z_L = 1$ such that $L(\mathbf{x}) = L$. The final constraint (4.19) ensures that all indicator variables are binary.

The inclusion property can also be explicitly enforced by the set of constraints:

$$\sum_{i \in \mathcal{V}} z_{i;a} \geq z_L, \forall a \in L \subseteq \boldsymbol{\mathcal{L}}. \tag{4.20}$$

In this case the formulation would be applicable also to co-occurrence potentials not satisfying the parsimony property. However, this would simply encourage degenerate solutions in which only one pixel takes a particular label.

The IP can be converted to a linear program (LP) by relaxing the integral constraints (4.19) to

$$z_{i;a}, z_{ij;ab}, z_L \in [0,1] \, \forall i \in \mathcal{V}, \forall (i,j) \in \mathcal{E}, \forall a, b \in \mathcal{L}, \forall L \subseteq \boldsymbol{\mathcal{L}}. \tag{4.21}$$

The resulting linear program can be solved using any general purpose LP solver, and an integer solution can be induced using rounding schemes such as those of Kleinberg and Tardos (1999). While this approach allows co-occurrence to be computed effectively for small images, over large images the memory and time requirements of standard LP solvers make this approach infeasible.

In many practical cases the co-occurrence cost $C(L)$ is defined as the sum over costs k_L for co-occurrence of subsets of labels, for example all pairs of labels. The cost k_L for each subset is taken if all the labels L are present in an image.

$$C(L) = \sum_{B \subseteq L} k_B, \tag{4.22}$$

where $k_B \geq 0$. In general any cost $C(L)$ can be decomposed uniquely into the sum over subsets recursively as:

$$k_B = C(B) - \sum_{B' \subset B} k_{B'}, \tag{4.23}$$

however some coefficients k_B may become negative. We show, that in the case of the co-occurrence cost defined as a sum over costs for low-order subset of labels, we can remove the exponential complexity of the linear program on the number of labels. In this case we will need one variable z_L for each subset, which is either of the cardinality 1 or has a nonzero cost $k_L > 0$. The variable z_L will be set if all the labels in L are present in an image (See 4.12 for formal definition). In this case, the linear program becomes:

$$\min_{\mathbf{z}} \sum_{i \in \mathcal{V}, a \in \mathcal{L}} \psi_i(a) z_{i;a} + \sum_{(i,j) \in \mathcal{E}, a,b \in \mathcal{L}} \psi_{i,j}(a,b) z_{ij;ab} + \sum_{L \subseteq \mathcal{L}} k(L) z_L \tag{4.24}$$

such that the standard LP constraints (4.14 - 4.16) hold:

$$\begin{aligned}
\textstyle\sum_a z_{ij;ab} &= z_{j;b}, & \forall (i,j) \in \mathcal{E}, b \in \mathcal{L}, \quad (4.25)\\
\textstyle\sum_b z_{ij;ab} &= z_{i;a}, & \forall (i,j) \in \mathcal{E}, a \in \mathcal{L}, \quad (4.26)\\
\textstyle\sum_a z_{i;a} &= 1, & \forall i \in \mathcal{V}, \quad (4.27)\\
&\text{and} & \\
z_{\{a\}} &\geq z_{i;a}, & \forall i \in \mathcal{V}, a \in \mathcal{L} \quad (4.28)\\
z_L \geq \textstyle\sum_{a \in L} z_{\{a\}} &- |L| + 1, & \forall L \subseteq \mathcal{L}, |L| \geq 2 \quad (4.29)\\
z_{i;a}, z_{ij;ab}, z_L &\in [0,1] & \forall i \in \mathcal{V}, \forall (i,j) \in \mathcal{E}, \\
\forall a, b &\in \mathcal{L}, & \forall L \subseteq \mathcal{L}. \quad (4.30)
\end{aligned}$$

The constraints (4.28) guarantee that $z_{\{a\}} = 1$ if the label a is present in an image. The constraints (4.29) enforce that for all L with cardinality larger than two, $z_L = 1$ if all labels in L are present in an image. In many practical cases, when the cost is defined as a sum over costs for each label as in (Delong et al., 2010), or each pair of labels, this LP program becomes feasible for standard LP solvers.

We next show that, the higher order energy (4.3) can be transformed into a pairwise energy function with the addition of a single auxiliary variable L that takes $2^{|\mathcal{L}|}$ states.

4.4.2 Pairwise Representation of Co-occurrence Potentials

The optimisation of the energy (4.3) is equivalent to the pairwise energy function with co-occurrence cost represented using one auxiliary variable z that takes a label from the set of subsets: $z \in 2^{\mathcal{L}}$. The unary potential for this auxiliary variable is equal to the corresponding co-occurrence cost:

$$\psi_u(z) = C(z) \qquad\qquad \forall z \in 2^{\mathcal{L}}. \tag{4.31}$$

The exclusion property is enforced by using a sufficiently large pairwise cost $K \to \infty$ for each pair of inconsistent labelling of pixel $x_i \in \mathbf{x}$ and z as:

$$\psi_p(x_i, z) = K\Delta(x_i \notin z) \qquad\qquad \forall x_i \in \mathbf{x}. \tag{4.32}$$

The inclusion property is implicitly encoded in a similar way to the IP formulation as it arises naturally in the usual solutions due to the parsimony. If $z = L$ and there was a label $l \in L$ such that $\forall x_i \in \mathbf{x} : x_i \neq l$, then the solution with $z = L \setminus \{l\}$ would have the same or lower cost $E(\mathbf{x})$.

This formulation allows us to use any approach from the wide body of standard inference techniques (Boykov et al., 2001; Kolmogorov, 2006; Szeliski et al., 2006) to minimise this function. However, the complexity grows exponentially with the size of the label set. In the case in which the costs can also be decomposed into the sum of positive co-occurrence costs for low-order subsets, the exponential dependency on the size of label set can be removed. The new pairwise formulation contains one variable z_L for each subset with non-zero cost $k_L > 0$. It takes any label $l \in L$, which is currently not present in an image, or label $\emptyset$ if all labels

$l \in L$ are present in an image. The unary potential for all auxiliary variables is equal to the corresponding co-occurrence cost, if all labels $l \in L$ are present in an image:

$$\psi_u(z_L) = k(L)\Delta(z_L = \emptyset) \qquad \forall L \subseteq \mathcal{L}. \tag{4.33}$$

The consistency of the state of z_L with the labelling on an image is enforced by using a sufficiently large pairwise cost $K \rightarrow \infty$ for each pair of inconsistent labelling of pixel $x_i \in \mathbf{x}$ and z_L as:

$$\psi_p(x_i, z_L) = K\Delta(z_L = l)\Delta(x_i = l) \qquad \forall x_i \in \mathbf{x}, L \subseteq \mathcal{L}. \tag{4.34}$$

Local minima of this pairwise graph can be found using the message passing algorithms designed for general pairwise graphs, described in chapter 3. However, in our experiments (section 4.5) such message passing algorithms did not perform as effectively as the graph cut based algorithm we describe next.

4.4.3 $\alpha\beta$-Swap Moves

We now prove that optimal swap moves can be computed for the relaxed energy (4.21), where the swap moves considered will exchange α and β in $\mathbf{x}$ and simultaneously find the optimal choice of z_L. We will use the notation $A \setminus B$ to refer to the subset of A that contains all the elements of A which are not in B.

Consider a swap move over the labels α and β, and starting from an initial set of labels $L(\mathbf{x})$. We write

$$L_{\alpha,\beta} = L(\mathbf{x}) \cup \{\alpha, \beta\}. \tag{4.35}$$

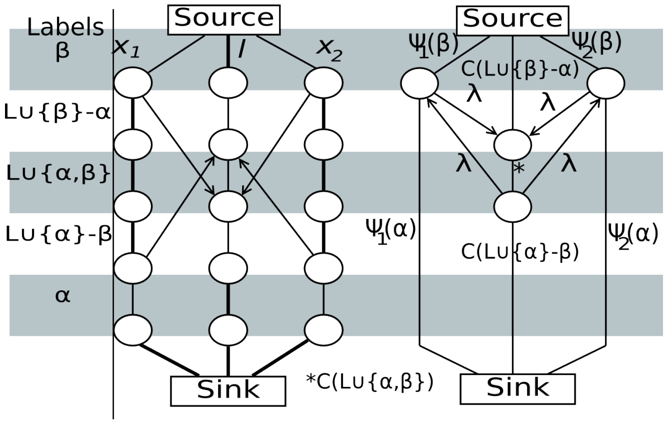

Figure 4.3: *The graph constructs used in swap inference. Each column represents the state of a different random variable.* Left*: The Ishikawa construction associated with swap inference, where the cutting of certain edges correspond to proposed moves. Bi-directional edges that can not be broken in either direction are marking in bold.* Right*: The reduced graph built by merging nodes connected by unbreakable edges in both directions, and the final weights. This is the graph ultimately solved to compute a swap move. See section 4.4.3.*

We assume that either α or β is present in the image[2]. Then, after a swap move the set of labels present must be an element of S which we define as

$$S = \{L_{\alpha,\beta} \setminus \{\beta\}, L_{\alpha,\beta} \setminus \{\alpha\}, L_{\alpha,\beta}\}\,. \tag{4.36}$$

Consequently, if we can represent a move that swaps the labels belonging to the variables in $\mathbf{x}$ that currently take labels α or β, while allowing z_L to vary over S, we can exactly compute optimal swap moves in the relaxation of the original energy (4.3).

Following the work of Ishikawa (2003) we note that the cost of the moves considered can be exactly represented as a submodular energy if there is an ordering o of the set $S' = S \cup \{\alpha, \beta\}$, such that the pairwise cost ϕ between any two nodes is convex, *i.e.* it satisfies $\phi(x_1, x_2) = g(o(x_1) - o(x_2))$, where $x_1, x_2 \in \mathbf{X} \cup z_L$, o is a function mapping from S' to $\mathbb{N}$ and g is a convex function.

[2] If this is not the case, no swap move is possible.

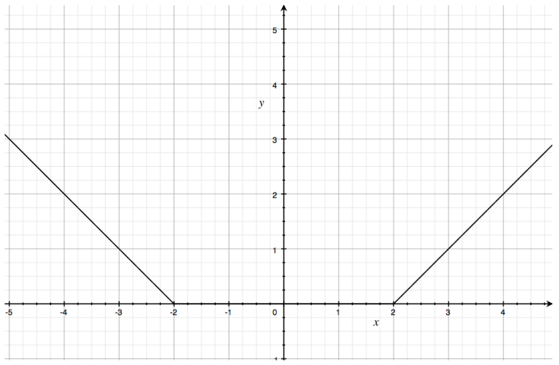

Figure 4.4: Illustration of the function $g(\cdot)$.

This approach has been used in the works (Kumar and Torr, 2008a; Veksler, 2007) that made use of an explicitly defined order over the set of all labels. However, unlike their approaches, our choice of ordering will change with every considered move, in the same way as we performed transformationally optimal moves in the previous chapter.

Such an ordering, and convex function exist, as we now show: To model the pairwise edges between the auxiliary variable z_L and the variables denoting pixels in the image, we define o as follows

$$o(x) = \begin{cases} 0 & \text{if } x = \alpha \\ 1 & \text{if } x = L_{\alpha,\beta} \setminus \{\beta\} \\ 2 & \text{if } x = L_{\alpha,\beta} \\ 3 & \text{if } x = L_{\alpha,\beta} \setminus \{\alpha\} \\ 4 & \text{if } x = \beta \end{cases} \tag{4.37}$$

and g as

$$g(y) = \begin{cases} 0 & \text{if } |y| \leq 2 \\ \lambda(|y| - 2) & \text{otherwise.} \end{cases} \tag{4.38}$$

By inspection g is convex (see figure 4.4), and

$$g(o(z_L) - o(x_i)) = \begin{cases} \lambda & \text{if } x_i \notin l \\ 0 & \text{otherwise.} \end{cases} \tag{4.39}$$

These are the costs we associated with the pairwise formulation in section 4.4.2. As the Ishikawa construction supports arbitrary unary costs we penalise the states that should not occur (for example $z_L = \beta$) with infinite cost unary potentials.

Theorem 1. *The resulting cost function is a pairwise submodular energy that exactly specifies what moves are possible, their costs, and can be efficiently solved using graph-cuts. Consequently the problem of proposing an optimal $\alpha\beta$ swap is exactly solvable.* □

The full graph construct can be seen in the left of figure 4.3.

To simplify inference we perform an additional *reduction step* after constructing the graph. In the Ishikawa construct many nodes are connected by unbreakable edges that prevent certain states from occurring. By simply merging any nodes connected in both directions by an unbreakable edge, we are able to significantly reduce the computational overhead, both in terms of time taken and memory requirements. Using this reduced form, if we let

$$C_\alpha = C(L_{\alpha,\beta} \setminus \{\beta\}) \tag{4.40}$$

$$C_{\alpha,\beta} = C(L_{\alpha,\beta}) \tag{4.41}$$

$$C_\beta = C(L_{\alpha,\beta} \setminus \{\alpha\}) \tag{4.42}$$

we can write the binary submodular function, from an initial state $\mathbf{x}$, that gives the cost of a swap move $\mathbf{t}$, where t is a binary vector such that $t_i = 0$ if $x_i = \alpha$

and $t_i = 1$ if $x_i = \beta$ as:[3]

$$E(\mathbf{t}) = \Phi(\mathbf{t}) + \min_{z_1,z_2\in\{0,1\}} \Big[C_\beta z_1 + C_\alpha(1-z_2) + C_{\alpha\beta}(1-z_1)z_2 \\ + \sum_{i\in\mathcal{V}:x_i\in\{\alpha,\beta\}} \lambda t_i(1-z_2) + \sum_{i\in\mathcal{V}:x_i\in\{\alpha,\beta\}} \lambda(1-t_i)z_1\Big] \tag{4.43}$$

where $\Phi(\mathbf{x})$ is the cost of a standard swap move. This formulation, along with it's edge costs can be seen in the right of figure 4.3.

4.4.4 α Expansion

A similar approach, using an over-estimation of move costs can be used to define α expansion type moves.

The cost $C(L(\mathbf{x}))$ is, in general, a higher-order non-submodular energy, and intractable. However, when proposing moves we can use the convex/concave procedure (see figure 4.5) as described in (Narasimhan and Bilmes, 2005; Rother et al., 2005) and over-estimate the cost of moving from the current solution.

We define a transformation vector t, which maps from a current labelling $\mathbf{x}$ to a new labelling $\mathbf{x}'$ and takes the value $t_i = 1$ if $x_i = \alpha$ and $t_i = 0$ if $x_i = x'_i$. Then our over-estimation $Q(\mathbf{t})$ of the cost of a move needs two properties;

$$C(\mathbf{x}) = Q(\mathbf{0}) \tag{4.44}$$

and

$$C(\mathbf{x_t}) \leq Q(\mathbf{t}).\ \forall \mathbf{t} \tag{4.45}$$

where $\mathbf{x}_t$ is the new labelling induced by move $\mathbf{t}$. That is $Q(\cdot)$ must be an upper bound of $C(\cdot)$ over the range of moves considered, and this upper bound must

[3] There are two subtleties to be aware of here: *(i)* Variables in $\mathbf{x}$ which are not taking either label α or β are unable to change label, and ignored in the transformation vector, and *(ii)* as we are always able to find the optimal label of z_L using only the current labelling of $\mathbf{x}$, and without prior knowledge of it's previous state, we do not need to track what label it takes.

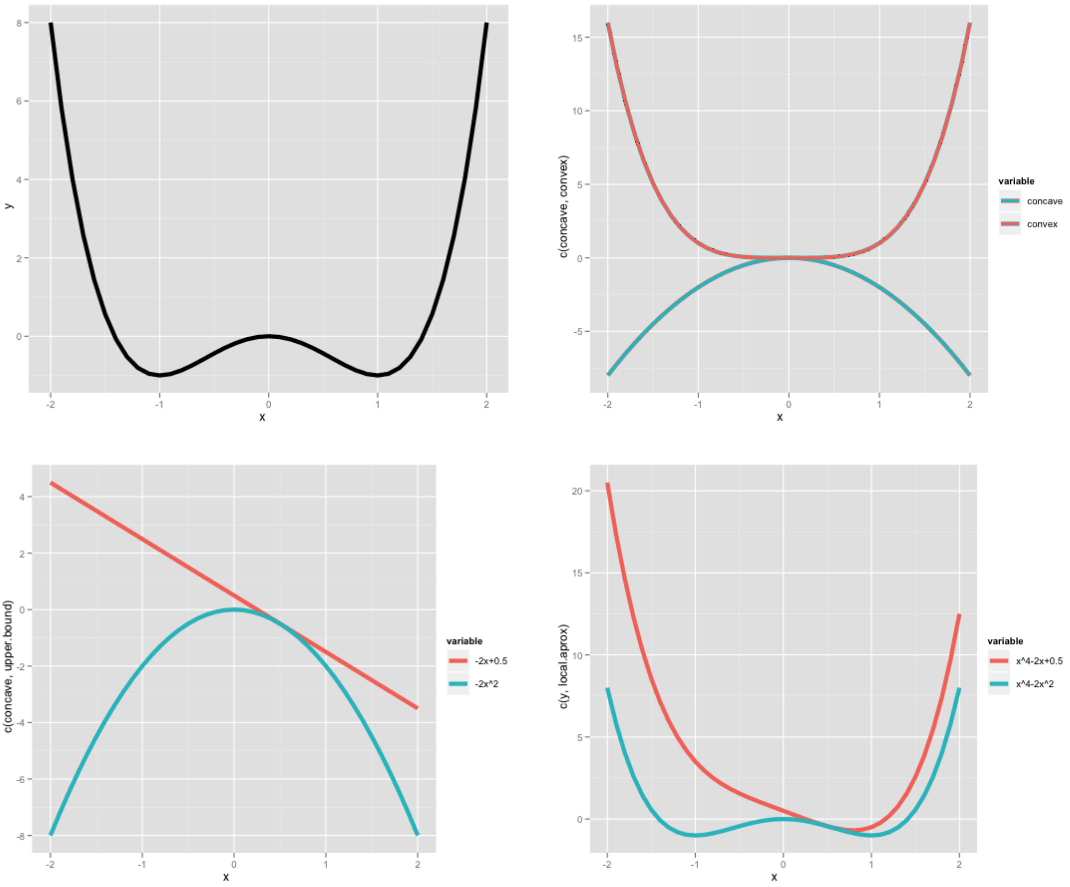

Figure 4.5: **The convex concave procedure** (Yuille et al., 2002) is a move making strategy for the optimisation of a non-convex or intractable function. Given a function such as $f(x) = x^4 - 2x^2$ (top left), and a current location $x = 0.5$, the function is decomposed into concave $(-2x^2)$ and convex (x^4) components (top right). We then replace the concave component with it's tangent at $x = 0.5$ (bottom left) This tangent $t(x)$ serves as an upper bound to the concave function that is *tight* at $x = 0.5$. As such it satisfies the general inequality $f(x) = t(x) + x^4 \geq t(x') + x'^4 \geq f(x')$ where x' is a global minima of the convex function $t(x) + x^4$. This means that moves made by minimising the convex approximation $t(x)+x^4$ of $f(x)$ must also decrease the cost of $f(x)$ and eventually converge to a local minima. The same strategy is used in our formulation of α-expansion where we conservatively over-estimate the cost of introducing a new label α (assuming it is not already present) and under-estimate the benefit of removing any other label. This conservative estimation forms an upper-bound of the true cost function, guaranteeing convergence.

be tight with respect to the current location $\mathbf{x}$. If this is the case, then given the optimal move $\mathbf{t}^*$, of $Q(\cdot)$ the following inequality holds

$$C(\mathbf{x}) = Q(\mathbf{0}) \geq Q(\mathbf{t}^*) \geq C(\mathbf{x}_{\mathbf{t}^*}). \tag{4.46}$$

If this property holds, repeatedly optimising over $\mathbf{t}$ for different choices of α must decrease the cost $C(\mathbf{x})$ and eventually converges. We choose our cost $Q(\mathbf{t})$ as a a sum of $\Phi(\mathbf{x})$, being the α-expansion moves proposed in the previous chapter, and a new term $P(\mathbf{t})$ which serves as an over-estimation of the label set cost $C(L(\mathbf{x}))$.

$$k_\beta = \min_{l \subseteq L(\mathbf{x})} C(l) - C(l \setminus \{\beta\}) \tag{4.47}$$

$$k'_\alpha = \begin{cases} 0 & \text{if } \alpha \in L(\mathbf{x}) \\ \max_{l \subseteq L(\mathbf{x})} C(l) \cup \{\alpha\}) - C(l) & \text{otherwise.} \end{cases} \tag{4.48}$$

$$P(\mathbf{t}) = \sum_{\beta \in L(\mathbf{x})} [k_\beta \Delta(\beta \in L(\mathbf{x}_t)] + k'_\alpha \Delta(\alpha \in \mathbf{x}_t) \tag{4.49}$$

If k_l is non-negative, the pairwise submodular energy is

$$E'(\mathbf{t}) = \Phi(\mathbf{t}) + \min_{\mathbf{z}} \sum_{\beta \in A \setminus \{\alpha\}} \left[k'_\beta(1 - z_\beta) + \sum_{i \in \mathcal{V}} \lambda(1 - t_i) z_\beta \right] \tag{4.50}$$

$$+ k'_\alpha z_\alpha \sum_{i \in \mathcal{V}: x_i = \beta} \lambda t_i (1 - z_\alpha)$$

where λ is a sufficiently large positive value.

4.5 Experiments

We performed a controlled test evaluating the performance of CRF models both with and without co-occurrence potentials. As a base line we used the segment-based CRF and the associative hierarchical random field (AHRF) model proposed

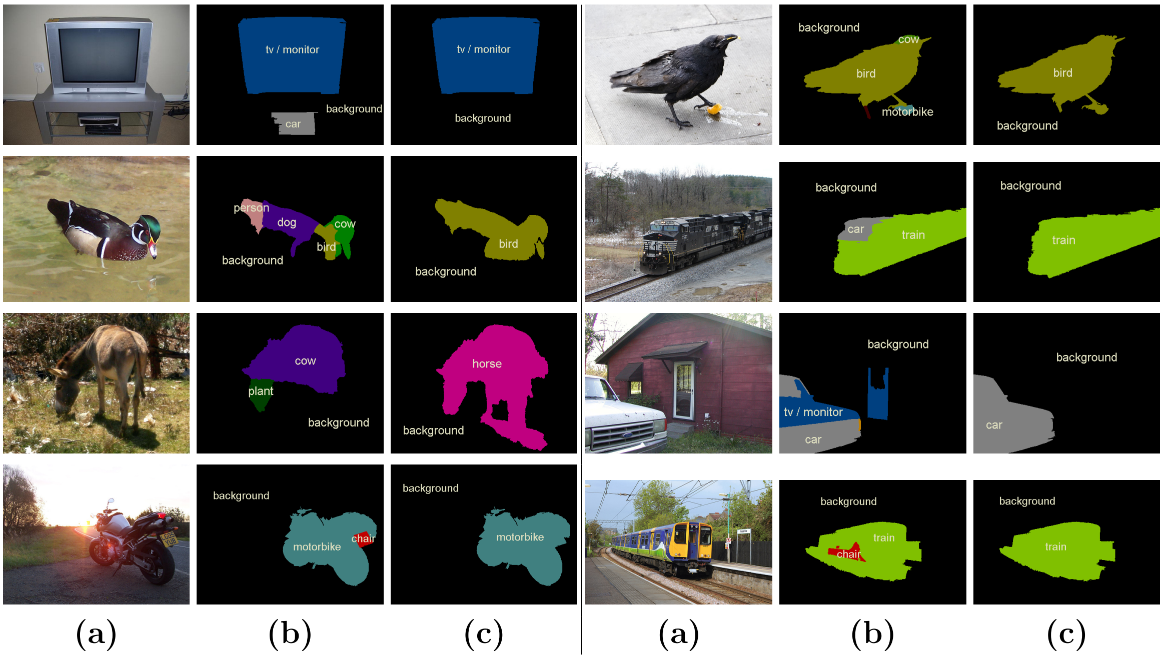

Figure 4.6: **Best viewed in colour: (a)** *Typical images taken from the* VOC-*2009 data set (Shotton et al., 2006);* **(b)** *A labelling based upon a pixel based random field model (Ladicky et al., 2009) that does not take into account co-occurrence;* **(c)** *A labelling of the same model using co-occurrence statistics. Note that the co-occurrence potentials perform in a similar way across different data sets, suppressing the smaller classes (see also figure 4.1) if they appear together in an uncommon combination with other classes such as a car with a monitor, a train with a chair or a dog with a bird. This results in a qualitative rather than quantitative difference.*

in discussed in chapter 2 and the inference method of chapter 3, which currently offers state of the art performance on the MSRC data set (Shotton et al., 2006). On the VOC data set, the baseline also makes use of the detector potentials of (Ladicky et al., 2010b) also discussed in chapter 2. The costs $C(L)$ were created from the training set as follows: let M be the number of images, $\mathbf{x}^{(m)}$ the ground truth labelling of an image m and

$$z_l^{(m)} = \Delta(l \in L(\mathbf{x}^{(m)})) \tag{4.51}$$

an indicator function for label l appearing in an image m. The associated cost was trained as:

$$C(L) = -w \log \frac{1}{M}\left(1 + \sum_{m=1}^{M} \prod_{l \in L} z_l^{(m)}\right), \tag{4.52}$$

where w is an arbitrary weighting of the co-occurrence potential, tuned via cross-validation. The form guarantees, that $C(L)$ is monotonically increasing with respect to L. To avoid over-fitting we approximated the potential $C(L)$ as a second order function:

$$C'(L) = \sum_{l \in L} c_l + \sum_{k,l \in L, k<l} c_{kl}, \tag{4.53}$$

where c_l and c_{lk} minimise the mean-squared error

$$\sum_{L \in \mathcal{L}} (C(L) - C'(L))^2 \tag{4.54}$$

between $C(L)$ and $C'(L)$.

On the MSRC data set we observed a 3% overall and 4% average per class increase in the recall and 6% in the intersection vs. union measure with the of the segment-based CRF and a 1% overall, 2% average per class and 2% in the intersection vs. union measure with the AHRF 4.5. The comparison on the VOC2009 data set was performed on the validation set, as the test set is not published and the number of permitted submissions is limited. Performance improved by 3.5% in the intersection vs. union measure used in the challenge (see table 4.2). The performance on the test set was 32.11% which is comparable with current state-of-the-art methods.

By adding a co-occurrence cost into the CRF we observe constant improvement in pixel classification for almost all classes in all measures. In accordance with desideratum *(iv)*, the co-occurrence potentials tend to suppress uncommon combination of classes and produce more coherent images in the labels space. This

	Global	Average	Building	Grass	Tree	Cow	Sheep	Sky	Aeroplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Segment CRF	77	64	70	95	78	55	76	95	63	81	76	67	72	73	82	**35**	72	17	88	29	62	45	17
Segment CRF with CO	80	68	77	**96**	80	69	82	98	69	82	79	75	75	81	85	**35**	76	17	89	25	61	50	**22**
Hierarchical CRF	86	75	81	**96**	87	72	84	**100**	77	**92**	86	**87**	87	95	95	27	**85**	33	**93**	43	**80**	62	17
Hierarchical CRF with CO	**87**	**77**	**82**	95	**88**	**73**	**88**	**100**	**83**	**92**	**88**	**87**	**88**	**96**	**96**	27	**85**	**37**	**93**	**49**	**80**	**65**	20

Table 4.1: *Quantitative results on the MSRC data set, average per class recall measure, defined as* $\frac{\textit{True Positives}}{\textit{True Positives + False Negatives}}$. *Incorporation of co-occurrence potentials led to a constant improvement for almost every class.*

results in a qualitative rather than quantitative difference. Although the unary potentials already capture textural context (Shotton et al., 2006), the incorporation of co-occurrence potentials leads to a significant improvement in accuracy.

It is not computationally feasible to perform a direct comparison between the work (Rabinovich et al., 2007) and our potentials, as the AHRF model is defined over individual pixels, and it is not possible to minimise the resulting fully connected graph which would contain approximately 4×10^{10} edges. Similarly, without their scene classification potentials it was not possible to do a like for like comparison with (Torralba et al., 2003).

Average running time on the MSRC data set without co-occurrence was $5.1s$ in comparison to $16.1s$ with co-occurrence cost. On the VOC2009 data set the average times were $107s$ and $388s$ for inference without respectively with co-occurrence costs. We compared the performance of α-expansion with LP relaxation using the solver given in Benson and Shanno (2007) for general co-occurrence potential on the sub-sampled images. Both methods produced identical results in terms of energy, however α-expansion was approximately $42,000$ times faster.

Table 4.2: *Quantitative analysis of VOC2009 results on validation set, intersection vs. union measure, defined as* $\frac{\textit{True Positive}}{\textit{True Positive + False Negative + False Positive}}$. *Incorporation of co-occurrence potential led to labellings, which visually look more coherent, but are not necessarily correct. Quantitatively the performance improved significantly, on average by 3.5% per class.*

	Average	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	TV/monitor
AHN	27.3	77.7	38.3	9.6	**24.0**	35.8	**31.0**	59.2	36.5	21.2	8.3	**1.7**	22.7	**14.3**	17.0	26.7	21.1	15.5	**16.3**	14.6	48.5	33.1
AHN with CO	**30.8**	**82.3**	**49.3**	**11.8**	19.3	**37.7**	30.8	**63.2**	**46.0**	**23.7**	**10.0**	0.5	**23.1**	14.1	**22.4**	**33.9**	**35.7**	**18.4**	12.1	**22.5**	**53.1**	**37.5**

4.6 Conclusion

The importance of co-occurrence statistics is well established (Torralba et al., 2003; Rabinovich et al., 2007; Csurka and Perronnin, 2008). In this work we examined the use of co-occurrence statistics and how they can be efficiently incorporated into a global energy or probabilistic model such as a conditional random field. We have shown how they can naturally be encoded by the use of higher order cliques, without a significant computational overhead. Whilst the performance improvements on current data sets are slight, we believe encoding co-occurrence will become increasingly important in the future when, rather than attempting to classify 20 classes in an image we have to classify $20,000$. Even with a false positive rate of 1% this would still give 200 false positives per image, co-occurrence information gives a natural way to tackle this problem.

Chapter 5

Efficient Minimisation of Higher Order Submodular Functions using Monotonic Boolean Functions

5.1 Overview

Submodular function minimisation is a key problem in a wide variety of applications in machine learning, economics, game theory, computer vision and many others. The general solver has a complexity of $O(n^6+n^5L)$ where L is the time required to evaluate the function and n is the number of variables (Orlin, 2007). On the other hand, many useful applications in computer vision and machine learning applications are defined over a special subclasses of submodular functions in which that can be written as the sum of many submodular cost functions defined over cliques containing few variables. In such functions, the pseudo-Boolean (or polynomial) representation (Boros and Hammer, 2002) of these subclasses are of degree (or order, or clique size) k where $k \ll n$. In this work, we develop efficient algorithms for the minimisation of this useful subclass of submodular functions. To do this, we define novel mapping that transform submodular functions of order k into quadratic ones, which can be efficiently minimised in $O(n^3)$ time using a max-flow algorithm. The underlying idea is to use auxiliary variables to model the higher order terms and the transformation is found using a carefully constructed linear program. In particular, we model the auxiliary variables as monotonic Boolean functions, allowing us to obtain a compact transformation using as few auxiliary variables as possible. Specifically, we show that our approach for fourth order function requires only 2 auxiliary variables in contrast to 30 or more variables used in existing approaches. In the general case, we give an upper bound for the number or auxiliary variables required to transform a function of order k using Dedekind number, which is substantially lower than the existing bound of 2^{2^k}.

5.2 Introduction

Many optimisation problems in several domains such as operations research, computer vision, machine learning, and computational biology involve *submodular* function minimisation. Submodular functions (See Definition 1) are discrete analogues of convex functions (Lovász, 1983). Examples of such functions include cut capacity functions, matroid rank functions and entropy functions. Submodular function minimisation techniques may be broadly classified into two categories: efficient algorithms for general submodular functions and more efficient algorithms for subclasses of submodular functions. This chapter falls under the second category.

General solvers: The role of submodular functions in optimisation was first discovered by Edmonds when he gave several important results on the related poly-matroids (Edmonds, 2003). Grötschel, Lovász and Schrijver first gave a polynomial-time algorithm for minimisation of submodular function using ellipsoid method (Grötschel et al., 1981). Recently several combinatoric and strongly polynomial algorithms (Fleischer and Iwata, 2003; Iwata, 2002; Iwata et al., 2001; Schrijver, 2000) have been developed based on the work of Cunningham (Cunningham, 1985). The current best strongly polynomial algorithm for minimising general submodular functions (Orlin, 2007) has a run-time complexity of $O(n^5L + n^6)$, where L is the time taken to evaluate the function and n is the number of variables. Weakly polynomial time algorithms with a smaller dependence on n also exist. For example, to minimise the submodular function $f(x)$ the scaling algorithm of Iwata (Iwata, 2003) has a run-time complexity of $O(n^4L + n^5) \log M$. As before, L refers to the time required to compute the function f and M refers to the maximum absolute value of the function f.

Specialised solvers: There has been much recent interest in the use of higher order submodular functions for better modelling of computer vision and machine learning problems (Kohli et al., 2007; Lan et al., 2006; Ishikawa, 2009). Such problems typical involve millions of pixels making the use of general solvers highly infeasible. Further, each pixel may take multiple discrete values and the conversion of such a problem to a Boolean one introduces further variables. On the other hand, the cost functions for many such optimisation algorithms belong to a small subclass of submodular functions. The goal of this chapter is to provide an efficient approach for minimising these subclasses of submodular functions using a max-flow algorithm.

Definition 1. *Submodular functions map $f : \mathbb{B}^{\mathbf{V}} \to \mathbb{R}$ and satisfy the following condition:*

$$f(X) + f(Y) \geq f(X \vee Y) + f(X \wedge Y) \tag{5.1}$$

where X and Y are elements of $\mathbb{B}^n$

In this chapter, we use a pseudo-Boolean polynomial representation for denoting submodular functions.

Definition 2. *Pseudo-Boolean functions (*PBF*) take a Boolean vector as argument and return a real number, i.e. $f : \mathbb{B}^n \rightarrow \mathbb{R}$. These can be uniquely expressed as multi-linear polynomials i.e. for all f there exists a unique set of real numbers $\{a_S : S \in \mathbb{B}^N\}$:*

$$f(x_1, ..., x_n) = \sum_{S \subseteq V} a_S(\prod_{j \in S} x_j), a_S \in \mathbb{R}, \tag{5.2}$$

where $a_\emptyset$ is said to be the constant term.

The term *order* refers to the maximum degree of the polynomial. A submodular function of second order involving Boolean variables can be easily represented using a graph such that the minimum cut, computed using a max-flow algorithm, also efficiently minimises the function. However, max-flow algorithms can not exactly minimise non-submodular functions or some submodular ones of an order greater than 3 (Živný et al., 2009). There is a long history of research in solving subclasses of submodular functions both exactly and efficiently using max-flow algorithms (Billionnet and Minoux, 1985; Kolmogorov and Zabih, 2004; Hammer, 1965; Zalesky, 2003; Queyranne, 2002). In this chapter we propose a novel linear programming formulation that is capable of definitively answering this question: given any pseudo Boolean function, it can derive a quadratic submodular formulation of the same cost, should one exist, suitable for solving with graph-cuts. Where such a quadratic submodular formulation does not exist, it will find the *closest* quadratic submodular function.

Let $\mathcal{F}^k$ denote the class of submodular Boolean functions of order k. It was first shown in (Hammer, 1965) that any function in $\mathcal{F}^2$ can be minimised exactly using a max-flow algorithm. In (Billionnet and Minoux, 1985; Kolmogorov and

Zabih, 2004), showed that any function in $\mathcal{F}^3$ can be transformed into functions in $\mathcal{F}^2$ and thereby minimised efficiently using max-flow algorithms. The underlying idea is to transform the third order function to a function in $\mathcal{F}^2$ using extra variables, which we refer to as *auxiliary variables* (AV). In the course of this chapter, you will see that these AVs are often more difficult to handle than variables in the original function and our algorithms are driven by the quest to understand the role of these auxiliary variables and to eliminate the unnecessary ones.

Recently, Zivny et al. made substantial progress in characterising the class of functions that can be transformed to $\mathcal{F}^2$. Their most notable result is to show that not all functions in $\mathcal{F}^4$ can be transformed to a function in $\mathcal{F}^2$. This result stands in strong contrast to the third order case that was positively resolved more than two decades earlier (Billionnet and Minoux, 1985). Using Theorem 5.2 from (Promislow and Young, 2005) it is possible to decompose a given submodular function in $\mathcal{F}^4$ into 10 different groups $\mathcal{G}_i, i = \{1..10\}$ where each $\mathcal{G}_i$ is shown in Table 5.1. Zivny et al. showed that one of these groups can not be expressed using any function in $\mathcal{F}^2$ employing any number of AVs. Most of these results were obtained by mapping the problem of minimising submodular functions to a valued constraint satisfaction problem.

5.2.1 Problem Statement and main contributions

Largest subclass of submodular functions We are interested in transforming a given function in $\mathcal{F}^k$ into a function in $\mathcal{F}^2$ using AVs. As such a transformation is not possible for all submodular functions of order four or more (Živný et al., 2009), our goal is to implicitly map the largest subclass $\mathcal{F}_2^k$ that can be transformed into $\mathcal{F}^2$. This distinction between the two classes $\mathcal{F}_2^k$ and $\mathcal{F}^k$ will be crucial in the remainder of the chapter.

Definition 3. *The class $\mathcal{F}_2^k$ is the largest subclass of $\mathcal{F}^k$ such that every function*

$f(\mathbf{x}) \in \mathcal{F}_2^k$ *has an equivalent quadratic function* $h(\mathbf{x}, \mathbf{z}) \in \mathcal{F}^2$ *using* AV*s* $\mathbf{z} = z_1, z_2, ..., z_m \in \mathbb{B}^m$ *satisfying the following condition:*

$$f(\mathbf{x}) = \min_{\mathbf{z} \in \mathbb{B}^m} h(\mathbf{x}, \mathbf{z}), \quad \forall \mathbf{x}. \tag{5.3}$$

In this chapter, we are interested in developing an algorithm to transform every function in this class $\mathcal{F}_2^k$ to a function in $\mathcal{F}^2$.

Efficient transformation of higher order functions: We propose a principled framework to transform higher order submodular functions to quadratic ones using a combination of monotonic Boolean functions(MBF) and linear programming. This framework provides several advantages. First we show that the state of an AV in a minimum cost labeling is equivalent to an MBFdefined over the original variables. This provides an upper bound on the number of AVgiven by the Dedekind number (Korshunov, 1981), which is defined as the total number of MBFs over a set of n binary variables. In the case of fourth order functions, there are 168 such functions. Using the properties of MBFs and the nature of these AVs in our transformation, we prove that these 168 AVs can be replaced by two AVs.

Minimal use of AVs: One of our goals is to use a minimum number(m) of AVs in performing the transformation of (5.3). Although, given a fixed choice of $\mathcal{F}^k$, reducing the value of m does not change the complexity of the resulting min/cut algorithm asymptotically, it is crucial in several machine learning and computer vision problems. In general, most image based labeling problems involve millions of pixels and in typical problems, the number of fourth order priors is linearly proportional to the number of pixels. Such problems may be infeasible for large values of m. A recent work shows that the transformation of functions in $\mathcal{F}_2^4$ using about 30 additional nodes (Živný and Jeavons, 2008). On the other hand, we show that we can transform the same class of functions using only 2 additional nodes. Note that this reduction is applicable to every fourth order term in the function. A typical vision problem may involve functions having 10000 $\mathcal{F}_2^4$ terms for an image of size 100 × 100. Under these parameters, our algorithm will use 20000 AVs, whereas the existing approach (Živný and Jeavons, 2008) would use as large as 300000 AVs. In several practical problems, this improvement will make a significant difference in the running time of the algorithm.

5.2.2 Limitations of Current Approaches and Open Problems

Decomposition of submodular functions: Many existing algorithms for transforming higher order functions target the minimisation of a single k-variable k^{th} order function. However, the transformation framework is incomplete without showing that a given n-variable submodular function of k^{th} order can be decomposed into several individual k-variable k^{th} order sub-functions. Billionet proved that it is possible to decompose a function in $\mathcal{F}^3$ involving several variables into 3-variable functions in $\mathcal{F}^3$ (Billionnet and Minoux, 1985). To the best of our knowledge, the decomposition of fourth or higher order functions is still an open problem. We believe that this problem will be to resolve as, in general, determining if a fourth order function is submodular is co-NP complete (Gallo et al., 1989). Given this, it is likely that specialised solvers based on max-flow algorithms may never solve the general class of submodular functions. However, this decomposition problem is not a critical issue in machine learning and vision problems. This is because the higher order priors from natural statistics already occur in different sub-functions of k nodes - in other words, the decomposition is known a priori. This chapter only focuses on the transformation of a single k-variable function in $\mathcal{F}^k$. As mentioned above, the solution to this problem is still sufficient to solve large functions with hundreds of nodes and higher order priors in machine learning and vision applications.

Non-Boolean problems: The results in this chapter are applicable only to set or pseudo-Boolean functions. Many real world problems involve variables that can take multiple discrete values. It is possible to convert any submodular multi-label second order function to their corresponding QBF (Ishikawa, 2003; Schlesinger and Flach, 2006). One can also transform any multi-labelled higher order function (both submodular and non-submodular) to their corresponding QBF by encoding each multi-label variable using several Boolean variables (Ramalingam et al., 2008).

Excess AVs: The complexity of an efficient max-flow algorithm is $O((n+m)^3)$ where n is the number of variables in the original higher order function and m is the number of AVs. Typically in imaging problems, the number of higher order terms is of $O(n)$ and the order k is less than 10. Thus the minimisation of the function corresponding to an entire image with $O(n)$ higher order terms will still have a complexity of $O((n+n)^3)$. However when m becomes at least quadratic in n, for example, if a higher-order term is defined over every triple of variables in V, the complexity of the max-flow algorithm will exceed that of a general solver being $O((n+n^3)^3)$. Thus in applications involving a very large number of higher order terms, a general solver may be more appropriate.

5.3 Notation and preliminaries

In what follows, we use a vector $\mathbf{x}$ to denote $\{x_1, x_2, x_3, ..., x_n\}$. Let $\mathbb{B}$ denote the Boolean set $\{0, 1\}$ and $\mathbb{R}$ the set of reals. Let the vector $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{B}^n$, and $\mathbf{V} = \{1, 2, ..., n\}$ be the set of indices of $\mathbf{x}$. Let $\mathbf{z} = (z_1, z_2, ..., z_k) \in \mathbb{B}^k$ denote the AVs. We introduce a *set representation* to denote the labellings of $\mathbf{x}$.

Let $S_4 = \{1, 2, 3, 4\}$ and let $\mathcal{P}$ be the power set of S_4. For example a labeling $\{x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1)$ is denoted by the set $\{1, 3, 4\}$.

Definition 4. *The (discrete) derivative of a function $f(x_1, \dots, x_n)$ with respect to x_i is given by:*

$$\frac{\delta f}{\delta x_i}(x_1, \dots, x_n) = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \tag{5.4}$$

Definition 5. *The second discrete derivative of a function $\Delta_{i,j}(\mathbf{x})$ is given by*

$$\begin{aligned}\Delta_{i,j}(\mathbf{x}) &= \frac{\delta}{\delta x_j}\frac{\delta f}{\delta x_i}(x_1, \dots, x_n) \qquad (5.5)\\ &= \Big(f(x_1,\dots,x_{i-1},1,x_{i+1},x_{j-1},1,x_{j+1}\dots,x_n) - f(x_1,\dots,x_{i-1},0,x_{i+1},x_{j-1},1,x_{j+1}\dots,x_n)\Big)\\ &- \Big(f(x_1,\dots,x_{i-1},1,x_{i+1},x_{j-1},0,x_{j+1}\dots,x_n) - f(x_1,\dots,x_{i-1},0,x_{i+1},x_{j-1},0,x_{j+1}\dots,x_n)\Big).\end{aligned}$$

Note that it follows from the definition of submodular functions (5.1)*, that their second derivative is always non-positive for all* $\mathbf{x}$

5.4 Transforming functions in $\mathcal{F}_2^n$ to $\mathcal{F}^2$

Consider the following submodular function $f(\mathbf{x}) \in \mathcal{F}_2^n$ represented as a multilinear polynomial:

$$f(\mathbf{x}) = \sum_{S \in B^n} a_S(\prod_{j \in S} x_j), a_S \in \mathbb{R} \tag{5.6}$$

Let us consider a function $h(\mathbf{x}, \mathbf{z}) \in \mathcal{F}^2$ where $\mathbf{z}$ is a set of AVs used to model functions in $\mathcal{F}_2^n$. Any general function in $\mathcal{F}^2$ can be represented as a multi-linear polynomial (consisting of linear and bi-linear terms involving all variables):

$$h(\mathbf{x}, \mathbf{z}) = \sum_i a_i\, x_i - \sum_{i,j:i>j} a_{i,j}\, x_i x_j + \sum_l a_l\, z_l - \sum_{l,m:l>m} a_{l,m}\, z_l z_m - \sum_{i,l} a_{i,l}\, x_i z_l \tag{5.7}$$

The negative signs in front of the bi-linear terms $(x_i x_j, z_l x_i, z_l z_m)$ emphasise that their coefficients $(-a_{ij}, -a_{il}, -a_{lm})$ must be non-positive if the function is submodular. We are seeking a function h such that:

$$f(\mathbf{x}) = \min_{\mathbf{z} \in \mathbb{B}^n} h(\mathbf{x}, \mathbf{z}), \forall \mathbf{x}. \tag{5.8}$$

Here the function $f(\mathbf{x})$ is known. We are interested in computing the coefficients $\mathbf{a}$, and in determining the number of auxiliary variables required to express a function as a pairwise submodular function. The problem is extremely challenging due to the inherent instability and dependencies within the problem – different choices of parameters cause auxiliary variables to take different states. To explore the space of possible solutions fully, we must characterise what states an AV takes.

5.4.1 Auxiliary Variables as Monotonic Boolean Functions

Definition 6. *A monotonic (increasing) Boolean function (*MBF*) $m : \mathbb{B}^n \to \mathbb{B}$ takes a Boolean vector as argument and returns a Boolean, s.t if $y_i \leq x_i, \forall i \implies m(\mathbf{y}) \leq m(\mathbf{x})$*

Lemma 2. *The function $z_S(\mathbf{x})$ defined as $\mathbf{x}$ by*

$$z_s(\mathbf{x}) = \arg\min_{z_s} \left(\min_{\mathbf{z}'} h(\mathbf{x}, \mathbf{z}', z_s) \right). \tag{5.9}$$

i.e. that maps from $\mathbf{x}$ to the Boolean state of z_s is an MBF *(See Definition 6), where $\mathbf{z}'$ is the set of all auxiliary variables except z_s.*

Proof. We consider a current labeling $\mathbf{x}$ with an induced labeling of $z_s = z_s(\mathbf{x})$. We first note

$$h'(\mathbf{x}, z_s) = \min_{\mathbf{z}'} h(\mathbf{x}, \mathbf{z}', z_s) \tag{5.10}$$

is a submodular function i.e. it satisfies (5.1). We now consider *increasing* the value of $\mathbf{x}$, that is given a current labeling $\mathbf{x}$ we consider a new labeling $\mathbf{x}^{(i)}$ such that

$$x_j^{(i)} = \begin{cases} 1 & \text{if } j = i \\ \\ x_j & \text{otherwise.} \end{cases} \tag{5.11}$$

We wish to prove

$$z_s(\mathbf{x}^{(i)}) \geq z_s(\mathbf{x}) \ \forall \mathbf{x}, i. \tag{5.12}$$

Note that if $z_s(\mathbf{x}) = 0$ or $x_i = 1$ this result is trivial. This leaves the case: $z_s(\mathbf{x}) = 1$ and $x_i = 0$. It follows from (5.5) that:

$$\begin{aligned} & h'(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, 0) - h'(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, 1) \geq \\ & h'(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, 1) - h'(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, 0). \end{aligned} \tag{5.13}$$

As, by hypothesis, $z_s(\mathbf{x}) = 1$ and $x_i = 0$ we have:

$$h'(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, 0) \geq h'(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, 1). \tag{5.14}$$

Hence

$$\begin{aligned} & h'(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, 0) - h'(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, 0) \geq \\ & h'(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, 1) - h'(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, 0), \end{aligned} \tag{5.15}$$

and

$$h'(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, 0) \geq h'(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, 1). \tag{5.16}$$

Therefore $z_s(\mathbf{x}^{(i)}) = 1$. Repeated application of the statement gives $y_i \leq x_i, \forall i \implies z_s(\mathbf{y}) \leq z_s(\mathbf{x})$ as required □ □

Definition 7. *The Dedekind number $M(n)$ is the number of* MBF*s of n variables.*

Finding a closed-form expression for $M(n)$ is known as the Dedekind problem (Kleitman, 1969; Korshunov, 1981).

The Dedekind number of known values are shown below: $M(1) = 3$, this corresponds to the set of functions:

$$M_1(x_1) \in \{\mathbf{0}, \mathbf{1}, x_1\}, \tag{5.17}$$

where $\mathbf{0}$ and $\mathbf{1}$ are the functions that take any input and return 0 or 1 respectively. $M(2) = 6$ corresponding to the set of functions:

$$M_2(x_1, x_2) = \{\mathbf{0}, \mathbf{1}, x_1, x_2, x_1 \vee x_2, x_1 \wedge x_2\} \tag{5.18}$$

Similarly, $M(3) = 20$, $M(4) = 168$, $M(5) = 7581$, $M(6) \approx 7.8 \times 10^6$, $M(7) \approx 2.4 \times 10^{12}$, and $M(8) \approx 5.6 \times 10^{23}$. For larger values of n, $M(n)$ remains unknown, and the development of a closed form solution remains an active area of research.

Lemma 3. *On transforming the largest graph-representable subclass of k^{th} order function to pairwise Boolean function, the upper bound on the maximal number of required* AV*s is given by the Dedekind number $M(k)$.*

Proof. The proof is straightforward. Consider a general multinomial, of similar form to equation (5.6), with more than $M(k)$ AVs. It follows from lemma 2 that at least 2 of the AVs must correspond to the same MBF, and always take the same values. Hence, all references to one of these AV in the pseudo-Boolean representation can be replaced with references to the other, without changing the associated costs. Repeated application of this process will leave us with a solution with at most $M(k)$ AVs. □ □

Although this upper bound is large for even small values of k, it is much tighter than the existing upper bound of $S(k) = 2^{2^k}$ (See Proposition 24 in (Zivny and Jeavons, 2008)). For even small values of $k = \{3, ..., 8\}$ the upper bound using Dedekind's number is much smaller: $(M(3) = 20, S(3) = 256)(M(4) = 168, S(4) = 65536), (M(5) = 7581, S(5) \approx 4.29 \times 10^9), (M(6) \approx 7.8 \times 10^6, S(6) \approx 1.85 \times 10^{19}), (M(7) \approx 2.4 \times 10^{12}, S(7) \approx 3.4 \times 10^{38}$ and $(M(8) \approx 5.6 \times 10^{23}, S(8) \approx 1.156 \times 10^{77})$. Zivny et.al. have emphasised the importance of improving this upper bound. In section 5.6, we will further tighten the bound for fourth order functions.

Note that this representation of AVs as MBF is over-complete, for example if the MBF of a auxiliary variable z_i is the constant function $z_i(\mathbf{x}) = \mathbf{1}$ we can replace $\min_{\mathbf{z}, z_i} h(\mathbf{x}, \mathbf{z}, z_i)$ with the simpler (i.e. one containing less auxiliary variables) function $\min_{\mathbf{z}} h(\mathbf{x}, \mathbf{z}, 1)$. Despite this, this is sufficient preliminary work for our main result:

Theorem 2. *Given any function f in $\mathcal{F}_2^k$, the equivalent pairwise form $f' \in \mathcal{F}^2$ can be found by solving a linear program.*

The construction of the linear program is given in the following section.

5.5 The Linear Program

A sketch of the formulation can be given as follows: In general, the presence of AVs of indeterminate state, given a labeling $\mathbf{x}$ makes the minimising an LP non-convex and challenging to solve directly. Instead of optimising this problem containing AVs of unspecified state, we create an auxiliary variable associated with every MBF. Hence given any labeling $\mathbf{x}$ the state of every auxiliary variable is fixed a priori, making the problem convex. We show how the constraints that a particular AV must conform to a given MBF can be formulated as linear constraints, and that consequently the problem of finding the closest member of $f' \in \mathcal{F}^2$ to any pseudo Boolean function is a linear program.

This program will make use of the max-flow linear program formulation to guarantee that the minimum cost labeling of the AVs corresponds to their MBFs. To do this we must first rewrite the cost of equation (5.7), in a slightly different form. We write:

$$\begin{aligned} f(\mathbf{x},\mathbf{z}) = c_\emptyset &+ \sum_i c_{i,s}\,(1-x_i) + \sum_i c_{t,i}\,x_i + \sum_{i,j:i>j} c_{i,j}\,x_i(1-x_j) \\ &+ \sum_l c_{l,s}\,(1-z_l) + \sum_l c_{t,l}\,(1-z_l) + \sum_{l,m:l>m} c_{l,m}\,z_l\,(1-z_m) + \sum_{i,l} c_{i,l}\,x_i\,(1-z_l) \end{aligned} \tag{5.19}$$

where $c_\emptyset$ is a constant that may be either positive or negative and all other c are non-negative values referred to as the *capacity* of an edge. By (Kolmogorov and Zabih, 2004; Billionnet and Minoux, 1985), this form is equivalent to that of (5.7), in that any function that can be written in form (5.7), can also be written as (5.19) and visa versa.

5.5.1 The Max-flow Linear Program

Under the assumption that $\mathbf{x}$ is fixed, we are interested in finding a minima of the equation:

$$\begin{aligned} f_\mathbf{x}(\mathbf{z}) = c_\emptyset &+ \sum_i c_{i,s}\,(1-x_i) + \sum_i c_{t,i}\,x_i + \sum_{i,j:i>j} c_{i,j}\,x_i(1-x_j) \\ &+ \sum_l c_{l,s}\,(1-z_l) + \sum_l c_{t,l}\,(1-z_l) + \sum_{l,m:l>m} c_{l,m}\,z_l\,(1-z_m) + \sum_{i,l} c_{i,l}\,x_i\,(1-z_l) \\ &= d_{\mathbf{x},\emptyset} + \sum_l d_{\mathbf{x},l,s}\,(1-z_l) + \sum_l d_{\mathbf{x},t,l}\,(1-z_l) + \sum_{l,m:l>m} d_{\mathbf{x},l,m}\,z_l\,(1-z_m) \end{aligned} \tag{5.20}$$

where

$$d_{\mathbf{x},\emptyset} = c_\emptyset + \sum_{i:x_i=0} c_{i,s} + \sum_{i:x_i=1} c_{t,i} + \sum_{i,j:i>j\wedge x_i=1\wedge x_j=0} c_{i,j} \tag{5.21}$$

$$d_{\mathbf{x},s,l} = c_{s,l} + \sum_{i:x_i=1} c_{i,l},\ d_{\mathbf{x},l,t} = c_{l,t} \text{ and } d_{\mathbf{x},l,m} = c_{l,m}. \tag{5.22}$$

Then the minimum cost of equation (5.19) may be found by solving its dual max-flow program. Writing $\nabla_{\mathbf{x},s}$ for flow from sink, and $\nabla_{\mathbf{x},t}$ for flow to the sink, we seek

$$\max \nabla_{\mathbf{x},s} + d_{\mathbf{x},\emptyset} \tag{5.23}$$

Subject to the constraints that

$$\begin{array}{rcll} f_{\mathbf{x},ij} - d_{\mathbf{x},ij} & \leq & 0 & \forall (i,j) \in E \\ \sum_{j:(j,i)\in E} f_{\mathbf{x},ji} - \sum_{j:(i,j)\in E} f_{\mathbf{x},ij} & \leq & 0 & \quad \forall i \neq s,t \\ \nabla_{\mathbf{x},s} + \sum_{j:(j,s)\in E} f_{\mathbf{x},js} - \sum_{j:(s,j)\in E} f_{\mathbf{x},sj} & \leq & 0 & \\ \nabla_{\mathbf{x},t} + \sum_{j:(j,t)\in E} f_{\mathbf{x},jt} - \sum_{j:(t,j)\in E} f_{\mathbf{x},tj} & \leq & 0 & \\ f_{\mathbf{x},ij} & \geq & 0 & \quad (i,j) \in E \end{array} \tag{5.24}$$

where E is the set of all ordered pairs $(l,m) : \forall l > m$, $(s,l) : \forall l$ and $(l,t) : \forall t$, and $f_{\mathbf{x},i,j}$ corresponds to the flow through the edge (i,j).

We will not use this exact LP formulation, but instead rely on the fact that *$f_{\mathbf{x}}(\mathbf{z})$ is a minimal cost labeling if and only if there exists a flow satisfying constraints (5.24) such that*

$$f_{\mathbf{x}}(\mathbf{z}) - \nabla_{\mathbf{x},s} - d_{\mathbf{x},\emptyset} \leq 0. \tag{5.25}$$

5.5.2 Choice of MBF as a set of linear constraints

We are seeking minima of a quadratic pseudo Boolean function of the form (5.19), where $\mathbf{x}$ is the variables we are interested in minimising and $\mathbf{z}$ the auxiliary variables. As previously mentioned, formulations that allow the state of the auxiliary variable to vary tend to result in non-convex optimisation problems. To avoid such difficulties, we specify as the location of minima of $\mathbf{z}$ as a set hard constraints. We want that:

$$\min_{\mathbf{z}} f_{\mathbf{x}}(\mathbf{z}) = f_{\mathbf{x}}([m_1(\mathbf{x}), m_2(\mathbf{x}), \ldots m_{M(k)}(\mathbf{x})])\ \forall \mathbf{x}. \tag{5.26}$$

where $f_{\mathbf{x}}$ is defined as in (5.20), and $m_1, \ldots m_{M(k)}$ are the set of all possible MBFs defined over $\mathbf{x}$. By setting all of the capacities $d_{i,j}$ to 0, it can be seen that a solution satisfying (5.26) must exist. It follows from the reduction described in lemma 2, and that all functions that can be expressed in a pairwise form can also be expressed in a form that satisfies these restrictions.

We enforce condition (5.26) by the set of linear constraints (5.24) and (5.25) for all possible choice of $\mathbf{x}$. formally we enforce the condition

$$f_{\mathbf{x}}([m_1(\mathbf{x}), \ldots, m_{M(k)}(\mathbf{x})) - \nabla_{\mathbf{x},s} - d_{\mathbf{x},\emptyset} \leq 0. \tag{5.27}$$

Substituting in (5.20) we have 2^k sets of conditions, namely,

$$\sum_l d_{\mathbf{x},l,s}\,(1-m_l(\mathbf{x}) + \sum_l d_{\mathbf{x},t,l}\,(1-m_l(\mathbf{x})) + \sum_{l,m:l>m} d_{\mathbf{x},l,m}\, m_l(\mathbf{x})\,(1-m_m(\mathbf{x})) - \nabla_{\mathbf{x},s} \leq 0, \tag{5.28}$$

subject to the set of constraints (5.24) for all $\mathbf{x}$. Note that we make use of the max-flow formulation, and not the more obvious min-cut formulation, as this remains a linear program even if we allow the capacity of edges $\mathbf{d}$[1] to vary.

Submodularity Constraints We further require that the quadratic function is submodular or equivalently, the capacity of all edges $c_{i,j}$ is non-negative. This can be enforced by the set of linear constraints that

$$c_{i,j} \geq 0 \forall i, j. \tag{5.29}$$

[1] In itself $\mathbf{d}$ is just a notational convenience, being a sum of coefficients in $\mathbf{c}$.

5.5.3 Finding the nearest submodular Quadratic Function

We now assume that we have been given an arbitrary function $g(\mathbf{x})$ to minimise, that may or may not lie in $\mathcal{F}^k$. We are interested in finding the closest possible function in $\mathcal{F}^2$ to it. To find the closest function to it (under theL_1 norm), we minimise:

$$\min_{\mathbf{c}} \sum_{\mathbf{x}\in\mathbb{B}^k} \left| g(\mathbf{x}) - \min_{\mathbf{z}} f(\mathbf{x},\mathbf{z}) \right| = \tag{5.30}$$

$$\min_{\mathbf{c}} \sum_{\mathbf{x}\in\mathbb{B}^k} \left| g(\mathbf{x}) - f(\mathbf{x},\mathbf{m}(\mathbf{x})) \right| = \tag{5.31}$$

$$\begin{aligned}\min_{\mathbf{c}} \sum_{\mathbf{x}\in\mathbb{B}^k} \Big| g(\mathbf{x}) - \big(c_\emptyset + \sum_i c_{i,s}\,(1-x_i) + \sum_i c_{t,i}\,x_i + \sum_{i,j:i>j} c_{i,j}\,x_i(1-x_j) \qquad (5.32)\\ + \sum_l c_{l,s}\,(1-m_l(\mathbf{x})) + \sum_l c_{t,l}\,(1-m_l(\mathbf{x})) + \sum_{l,m:l>m} c_{l,m}\,m_l(\mathbf{x})\,(1-m_m(\mathbf{x}))\\ + \sum_{i,l} c_{i,l}\,x_i\,(1-m_l(\mathbf{x}))\big)\Big|\end{aligned}$$

where $\mathbf{m}(\mathbf{x}) = [m_1(\mathbf{x}),\ldots,m_{M(k)}(\mathbf{x})]$ is the vector of all MBFs over $\mathbf{x}$, and subject to the family of constraints set out in the previous subsection. Note that expressions of the form $\sum_i |g_i|$ can be written as $\sum_i h_i$ subject to the linear constraints $h_i > g_i$ and $h_i > -g_i$ and this is a linear program. □

5.5.4 Discussion

Several results follow from this. In particular, if we consider a function g of the same form as equation (5.2) the set of equations such that

$$\min_{\mathbf{c}} \sum_{\mathbf{x}\in\mathbb{B}^k} \left| g(\mathbf{x}) - \min_{\mathbf{z}} f(\mathbf{x},\mathbf{z}) \right| = 0 \tag{5.33}$$

exactly defines a linear polytope for any choice of $|\mathbf{x}| = k$, and this result holds for any choice of basis functions.

Of equal note, the convex-concave procedure (Yuille et al., 2002), is a generic move-making algorithm that finds local optima by successively minimising a sequence of convex (i.e. tractable) upper-bound functions that are tight at the current location ($\mathbf{x}'$). (Narasimhan and Bilmes, 2005) showed how this could be similarly done for quadratic Boolean functions, by decomposing them into submodular and supermodular components. In the previous chapter, in order to handle arbitrary monotone increasing co-occurrence potentials we showed that any function could be decomposed into a quadratic submodular function, and an additional overestimated term. Nevertheless, this decomposition was not optimal, and we did not address the problem of finding a optimal over-estimation. The optimal overestimation which lies in $\mathcal{F}^2$ for a cost function defined over a clique $\mathbf{g}$ may be found by solving the above LP subject to the additional requirements:

$$g(\mathbf{x}) \leq f(\mathbf{x}, \mathbf{z}) \; \forall \mathbf{x} \tag{5.34}$$

$$g(\mathbf{x}') \geq f(\mathbf{x}', \mathbf{z}) \tag{5.35}$$

Efficiency concerns As we consider larger cliques, it becomes less computationally feasible to use the techniques discussed in this section, at least without pruning the number of auxiliary variables considered. As previously mentioned, constant AVs and AVs that corresponds to that of a single variable in x i.e. $z_l = x_i$ can be safely discarded without loss of generality. In the following section, we show that a function in $\mathcal{F}^4_2$ can be represented by only two AVs, rather than 168 as suggested by the number of possible MBF. However, in the general case a minimal form representation eludes us. As a matter of pragmatism, it may be useful to attempt to solve the LP of the previous section without making use of any AV, and to successively introduce new variables, until a minimum cost solution is found.

5.6 Tighter Bounds: Transforming functions in $\mathcal{F}_2^4$ to $\mathcal{F}^2$

Consider the following submodular function $f(x_1, x_2, x_3, x_4) \in \mathcal{F}^4$ represented as a multi-linear polynomial:

$$f(x_1,x_2,x_3,x_4) = a_0+\sum_i a_ix_i+\sum_{i>j} a_{ij}x_ix_j+\sum_{i>j>k} a_{ijk}x_ix_jx_k+a_{1234}x_1x_2x_3x_4, \quad \Delta_{ij}(\mathbf{x}) \leq 0 \tag{5.36}$$

where $i, j, k = S_4$ and $\Delta_{ij}(\mathbf{x})$ is the discrete second derivative of $f(\mathbf{x})$ with respect to x_i and x_j. e

Consider a function $h(x_1, x_2, x_3, x_4, z_s) \in \mathcal{F}^2$ where z_s is an AV used to model functions in $\mathcal{F}^4$. Any general function in $\mathcal{F}^2$ can be represented as a multi-linear polynomial (consisting of linear and bilinear terms involving all five variables):

$$h(x_1,x_2,x_3,x_4,z_s) = b_0+\sum_i b_ix_i-\sum_{i>j} b_{ij}x_ix_j-(g_s-\sum_{i=1}^{4} g_{s,i}x_i)z_s, \qquad b_{ij} \geq 0, g_{s,i} \geq 0, i,j \in S_4. \tag{5.37}$$

The negative signs in front of the bilinear terms (x_ix_j, z_sx_i) emphasise that their coefficients $(-b_{ij}, -g_{s,i})$ must be non-positive to ensure submodularity. We have the following condition from equation (5.3):

$$f(x_1, x_2, x_3, x_4) = \min_{z_s \in \mathbb{B}} h(x_1, x_2, x_3, x_4, z_s), \forall \mathbf{x}. \tag{5.38}$$

Here the coefficients $(a_i, a_{ij}, a_{ijk}, a_{ijkl})$ in the function $f(\mathbf{x})$ are known. We wish to compute the coefficients $(b_i, b_{ij}, g_s, g_{s,n})$ where $i, j \in \mathbf{V}, i \neq j, n \in S_4$. If we were given $(g_s, g_{s,i})$ then from equations (5.37) and (5.38) we would have

$$z_s = \begin{cases} 1 & \text{if } g_s - \sum_{i=1}^{4} g_{s,i}x_i < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.39}$$

The value of z_s that minimises equation (5.38) is dependent both upon the assignment of $\{x_1, x_2, x_3, x_4\}$ and upon the coefficients $(g_s, g_{s,1}, g_{s,2}, g_{s,3}, g_{s,4})$. The four variables x_1, x_2, x_3 and x_4 can be assigned to 16 different labellings of (x_1, x_2, x_3, x_4) giving 16 equations in the following form:

$$f(x_1, x_2, x_3, x_4) = \underbrace{\underbrace{h(x_1, x_2, x_3, x_4, 0)}_{h_1} + \underbrace{\min_{z_s \in \mathbb{B}} (g_s - \sum_{i=1}^{4} g_{s,i}x_i) z_s}_{h_2}}_{h} \tag{5.40}$$

The function h_1 is the part of h not dependent on z_s, and h_2 is the part dependent on z_s. Our main result is to prove that any function $h \in \mathcal{F}^2$ can be transformed to a function $h'(x_1, x_2, x_3, x_4, z_{j1}, z_{j2}) \in \mathcal{F}^2$ involving only two auxiliary variables z_{j1} and z_{j2}. Using this result we can transform a given function $f(x_1, x_2, x_3, x_4) \in \mathcal{F}_2^4$, the form of which we characterise later, to a function $h'(x_1, x_2, x_3, x_4, z_{j1}, z_{j2}) \in \mathcal{F}^2$.

Let $\mathcal{A}$ be the family of sets corresponding to labellings of $\mathbf{x}$ such that:$z_s = 0 = \arg\min_{z_s} h(\mathbf{x}, z_s)$. In the same way let $\mathcal{B}$ be the family of sets corresponding to labellings of $\mathbf{x}$ such that:$z_s = 1 = \arg\min_{z_s} h(\mathbf{x}, z_s)$. These sets $\mathcal{A}$ and $\mathcal{B}$ partition $\mathbf{x}$, as defined below:

Definition 8. *A partition divides $\mathcal{P}$ into sets $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} = \{\mathcal{S}(\mathbf{x}) : 0 = \arg\min_{z \in \mathbb{B}} h(\mathbf{x}, z), \mathbf{x} \in \mathbb{B}^4\}$ and $\mathcal{B} = \mathcal{P} \backslash \mathcal{A}$. Note that $\emptyset \in \mathcal{A}$.*

For the rest of the chapter, we say that the AV z_s is associated with $[\mathcal{A}, \mathcal{B}]$ or denote it by $z_s : [\mathcal{A}, \mathcal{B}]$. We illustrate the concept of a *partition* in figure 5.1.

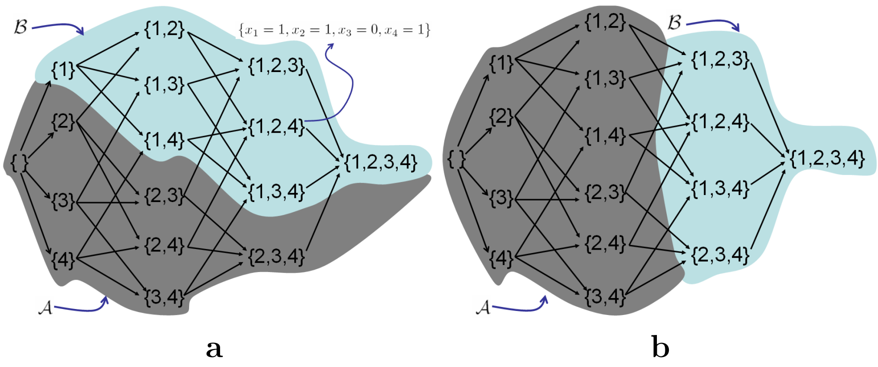

Figure 5.1: Hasse diagrams sample partitions. Here, we use set representation for denoting the labellings of (x_1, x_2, x_3, x_4). For example the set $\{1, 2, 4\}$ is equivalent to the labeling $\{x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1\}$. In (a), $\mathcal{A} = \{\{\}, \{2\}, \{3\}, \{4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{2, 3, 4\}\}$ and $\mathcal{B} = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, S_4\}$. (a) and (b) are examples of partitions. On searching the space of all possible partitions (2^{16}) we found that only 168 partitions belong to this class. These are the only partitions which will be useful in our analysis because any arbitrary AV must be associated with one of these 168 partitions.(See text for the relation between these partitions and MBF s).

From lemma 3, we could use 168 different AVs in our transformation. However, we show that the same class can be represented using only two AVs. In other words, all existing partitions could be converted to these two reference partitions represented by two AVs taking the states shown below.

Definition 9. *The* **forward reference partition** $[\mathcal{A}_f, \mathcal{B}_f]$ *takes the form:*

$$B \in \mathcal{B}_f \iff |B| \geq 3, \mathcal{A}_f = \mathcal{P} \backslash \mathcal{B}_f \tag{5.41}$$

On the other hand, a **backward reference partition** $[\mathcal{A}_b, \mathcal{B}_b]$ *is shown below:*

$$B \in \mathcal{B}_b \iff |B| \geq 2, \mathcal{A}_b = \mathcal{P} \backslash \mathcal{B}_b \tag{5.42}$$

The forward and backward reference partitions are shown in figure 5.2. Note that these reference partitions satisfy the properties of a matroid. Here we treat $\mathcal{A}$ as the family of subsets of the ground set S_4. More specifically, these reference *partitions satisfy the conditions of a uniform matroid (see appendix).*

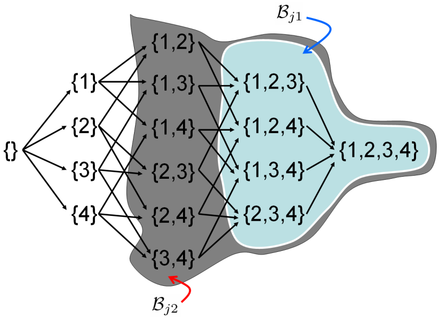

Figure 5.2: *The two* matroidal generators *used to represent all functions in* $\mathcal{F}_2^4$*. Note that the bilinear term* $z_{j1}z_{j2}$ *is active, i.e.* $z_{j1}z_{j2} = 1$*, in the region of overlap.*

We approach this problem by first considering the simplified case in which no interactions between AVs are allowed. This is covered in section 5.6.1, while section 5.6.2 builds on these results to handle the case of pairwise interactions between AV.

5.6.1 Non-interacting AVs

Here we study the role of AV independently. In other words, we don't consider the interaction of AVs that involve bilinear terms such as $z_i z_j$. The following lemmas and theorems enable the replacement of AVs with other AVs closer to the reference partitions. By successively applying replacement algorithms, we gradually replace all the AVs using with the two AVs in forward and backward reference partitions.

Lemma 4. *Let* $z_s : [\mathcal{A}_s, \mathcal{B}_s]$ *be an* AV *in a function* $h(\mathbf{x}, z_s)$ *in* $\mathcal{F}^2$ *, then* h *can be transformed to some function* $h'(\mathbf{x}, z_t)$ *in* $\mathcal{F}^2$ *involving* $z_t : [\mathcal{A}_t, \mathcal{B}_t]$*, such that for all* $B \in \mathcal{B}_t$*,* $|B| \geq 2$.

Proof. We say that a function h can be transformed to h' if $\min_{z_s} h(\mathbf{x}, z_s) = \min_{z_t} h'(\mathbf{x}, z_t), \forall \mathbf{x}$. It does not imply that $h(\mathbf{x}, z_s) = h'(\mathbf{x}, z_t), \forall \mathbf{x}$. We first consider the case where $\emptyset \in \mathcal{B}_s$. If this is the case, $\arg\min_{z_t} h'(\mathbf{x}, z_t) = 0 \ \forall \mathbf{x}$. Hence we can transform $h(\mathbf{x}, z_s)$ to $h'(\mathbf{x})$ and the lemma holds trivially. Next we assume that there exists a singleton $\{e\} \in \mathcal{B}_s$, i.e. $\{e\}$ is $\{1\}$,$\{2\}$,$\{3\}$ or $\{4\}$. We decompose h as:

$$\min_{z_s} h(x_1, x_2, x_3, x_4, z_s) = h_1(x_1, x_2, x_3, x_4) + \min_{z_s} \underbrace{(g_s - \sum_{i=1}^{4} g_{s,i} x_i) z_s}_{h_2}$$

where h_2 is the part of h dependent on z_s.

$$\min_{z_s} h_2 = \min_{z_s}((g_s - g_{s,e}) x_e z_s + (g_s - g_s x_e - \sum_{i=S_4 \backslash e} g_{s,i} x_i) z_s).$$

As $(e) \in \mathcal{B}_s$, $g_s - g_{s,e} \leq 0$. As a result, $z_s = 1$ when $x_e = 1$, i.e. $x_e \implies z_s$ or $x_e z_s = x_e$. In the above equation we replace $x_e z_s$ using simply x_e to obtain the following:

$$\min_{z_s} h_2 = \min_{z_s}((g_s - g_{s,e}) x_e + (g_s - g_s x_e - \sum_{i=S_4 \backslash e} g_{s,i} x_i) z_s).$$

The decomposition of the original function can then be written, replacing z_s by z_t:

$$h' = \underbrace{h_1 + (g_s - g_{s,e}) x_e}_{h'_1} + \underbrace{(g_s - g_s x_e - \sum_{i=S_4 \backslash e} g_{s,i} x_i) z_t}_{h'_2} .$$

A sample reduction for this lemma is shown in figure 5.3. Note that h'_2 equals 0 for the singleton $\{e\}$. Similarly any other singleton $\{e'\}$ can also be removed from $\mathcal{B}_s$ using the same approach. After repeated application, our final partition, $\mathcal{B}_t$ does not contain any singletons. □ □

Lemma 5. *Any function $h(\mathbf{x}, z_s)$ in $\mathcal{F}^2$ with z_s associated with the partition $[\mathcal{A}_s, \mathcal{B}_s]$ satisfying the condition $\mathcal{B}_s \subseteq \mathcal{B}_f$ can be transformed to some function $h'(\mathbf{x}, z_f)$ in $\mathcal{F}^2$ with z_f belonging to the forward reference partition $[\mathcal{A}_f, \mathcal{B}_f]$. The*

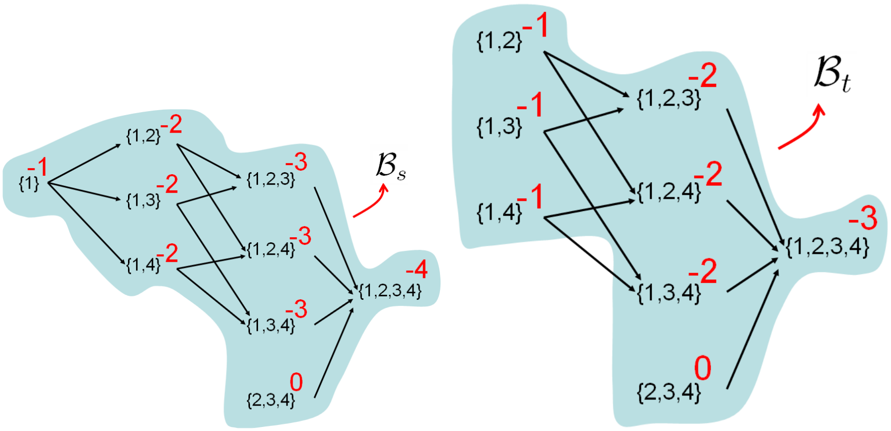

Figure 5.3: *An example of lemma 4. The* AV z_s *is replaced by* z_t *and the associated partitions* $[\mathcal{A}_s, \mathcal{B}_s]$ *and* $[\mathcal{A}_t, \mathcal{B}_t]$ *are shown in (a) and (b) respectively. The initial and the final set of parameters are given by:*$(g_s = 3, g_{s,1} = 4, g_{s,2} = 1, g_{s,3} = 1, g_{s,4} = 1), (g_t = 3, g_{t,1} = 3, g_{t,2} = 1, g_{t,3} = 1, g_{t,4} = 1)$. *In the initial partition we have the singleton* $\{1\} \in \mathcal{B}_s$. *After the transformation all the singletons* $\{e\} \in \mathcal{A}_t$.

same result holds for backward partition.

Proof. The proof is by construction. Let the parameters of the partition $[\mathcal{A}_s, \mathcal{B}_s]$ be

$(g_s, g_{s,1}, g_{s,2}, g_{s,3}, g_{s,4})$. Our goal is to compute a new set of parameters $(g_f, g_{f,1}, g_{f,2}, g_{f,3}, g_{f,4})$ corresponding to the forward reference partition such that the associated functions keep the same value at the minimum:

$$\min_{z_f} h'(\mathbf{x}, z_f) = \min_{z_s} h(\mathbf{x}, z_s), \forall \mathbf{x} \tag{5.43}$$

$$\min_{z_f}(h'_1(\mathbf{x}) + h'_2(\mathbf{x}, z_f)) = \min_{z_s}(h_1(\mathbf{x}) + h_2(\mathbf{x}, z_s)), \forall \mathbf{x} \tag{5.44}$$

$$\min_{z_f}(h'_2(\mathbf{x}, z_f)) = \min_{z_s}(h_2(\mathbf{x}, z_s)), \forall \mathbf{x} \tag{5.45}$$

We can rewrite h_2 and h'_2 using κ function:

$$\min_{z_f} \kappa(f, S) z_f = \min_{z_s} \kappa(s, S) z_s, \forall S \in \mathcal{P} \tag{5.46}$$

By substituting the values of z_s and z_f for all $S \in \mathcal{P}$ we obtain five equations with five unknowns $(g_f, g_{f,1}, g_{f,2}, g_{f,3}, g_{f,4})$. We rewrite the equations as:

$$\underbrace{\begin{pmatrix} 1 & -1 & -1 & 0 & -1 \\ 1 & -1 & -1 & -1 & 0 \\ 1 & 0 & -1 & -1 & -1 \\ 1 & -1 & 0 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 \end{pmatrix}}_{\mathcal{H}} \begin{pmatrix} g_f \\ g_{f,1} \\ g_{f,2} \\ g_{f,3} \\ g_{f,4} \\ g_{f,5} \end{pmatrix} = \begin{pmatrix} \min(0, \kappa(s, \{2,3,4\})) \\ \min(0, \kappa(s, \{1,3,4\})) \\ \min(0, \kappa(s, \{1,2,4\})) \\ \min(0, \kappa(s, \{1,2,3\})) \\ \min(0, \kappa(s, S_4)) \end{pmatrix} \tag{5.47}$$

The solution to the above linear system is unique because $\mathcal{H}$ is of rank 5. Now we show that the solution satisfies submodularity condition and corresponds to the forward reference partition. Submodularity is ensured by the constraint that the parameters $(g_{f,1}, g_{f,2}, g_{f,3}, g_{f,4})$ are all non-negative. Using equation (5.47) and the non-negativity of original variables $(g_{s,i})$ we obtain the following:

$$g_{f,i} = \min(0, \kappa(s, S_4 \backslash i)) - \min(0, \kappa(s, S_4)) \tag{5.48}$$

$$\kappa(s, S_4) \leq \kappa(s, S_4 \backslash i) \tag{5.49}$$

From these equations we can show that $g_{f,i}$ is always non-negative:

$$g_{f,i} = \begin{cases} 0 & \text{if } \kappa(s, S_4) \geq 0 \text{ and } \kappa(s, S_4 \backslash i) \geq 0 \\ -\kappa(s, S_4) & \text{if } \kappa(s, S_4) \leq 0 \text{ and } \kappa(s, S_4 \backslash i) = 0 \\ \kappa(s, S_4 \backslash i) - \kappa(s, S_4) & \text{if } \kappa(s, S_4) \leq 0 \text{ and } \kappa(s, S_4 \backslash i) \leq 0 \end{cases} \tag{5.50}$$

We now prove that the computed parameters correspond to the forward reference partition:

$$S \in \begin{cases} \mathcal{B}_f & \text{if } |S| \geq 3 \\ \mathcal{A}_f & \text{otherwise} \end{cases} \tag{5.51}$$

From equation (5.47) it follows that any set S, such that $|S| \geq 3$, exists in $\mathcal{B}_f$. We need to prove the remaining case where $|S| \leq 3$. To do this, we consider

$S = \{i,j\} = S_4 \backslash \{k,l\}$ and examine its partition coefficients:

$$\begin{aligned}
\kappa(f,\{i,j\}) &= \kappa(f,\{i,j,k\}) + g_{f,k} \\
\kappa(f,\{i,j\}) &= \kappa(f,\{i,j,k\}) + ((\kappa(f,\{i,j,l\}) - \kappa(f,\{i,j,k,l\}) \\
\kappa(f,\{i,j\}) &= \min(0,\kappa(s,\{i,j,k\})) + \min(0,\kappa(s,\{i,j,l\})) - \min(0,\kappa(s,\{i,j,k,l\}))
\end{aligned}$$

As in table 5.2 (see appendix), $\kappa(f,\{i,j\})$ has four possible values and $\kappa(f,\{i,j\}) \geq 0$ in all. As each set $S : |S| = 2$ exist in $\mathcal{A}_f$, every other set with a cardinality less than two must also exist in $\mathcal{A}_f$. Hence, for every partition $\mathcal{A}_s, \mathcal{B}_s$ satisfying $\mathcal{B}_s \subseteq \mathcal{B}_f$, we can compute an equivalent reference partition $[\mathcal{A}_f, \mathcal{B}_f]$. □ □

Lemma 6. *Let $P = \{i,j,k,l\} = S_4$ and let z_s be the auxiliary variable in $h(\mathbf{x}, z_s)$ associated with the partition $[\mathcal{A}_s, \mathcal{B}_s]$. If both A and $B = P\backslash A$ are elements of $\mathcal{B}_s$, then it is not possible to have both C and $D = P\backslash C$ in $\mathcal{A}_s$.*

Proof. The statement follows by contradiction. Let $\{A, B\}$, where $B = P\backslash A$, exist in $\mathcal{B}_s$. The partition coefficients of A and B with respect to z_1 are shown below:

$$\kappa(s, A) = g_s - \sum_{i=1}^{4} \mathbf{1}_i^A \leq \quad 0 \tag{5.52}$$

$$\kappa(s, B) = g_s - \sum_{i=1}^{4} \mathbf{1}_i^B \leq \quad 0 \tag{5.53}$$

Note that $A \cup B = \{i,j,k,l\}$ and $A \cap B = \emptyset$. Hence by summing the above equations we get the following:

$$2g_s - g_{s,i} - g_{s,j} - g_{s,k} - g_{s,l} \leq 0 \tag{5.54}$$

Assume now that a different pair $\{C, D\}$, where $D = P\backslash C$ exist in $\mathcal{A}_s$. By

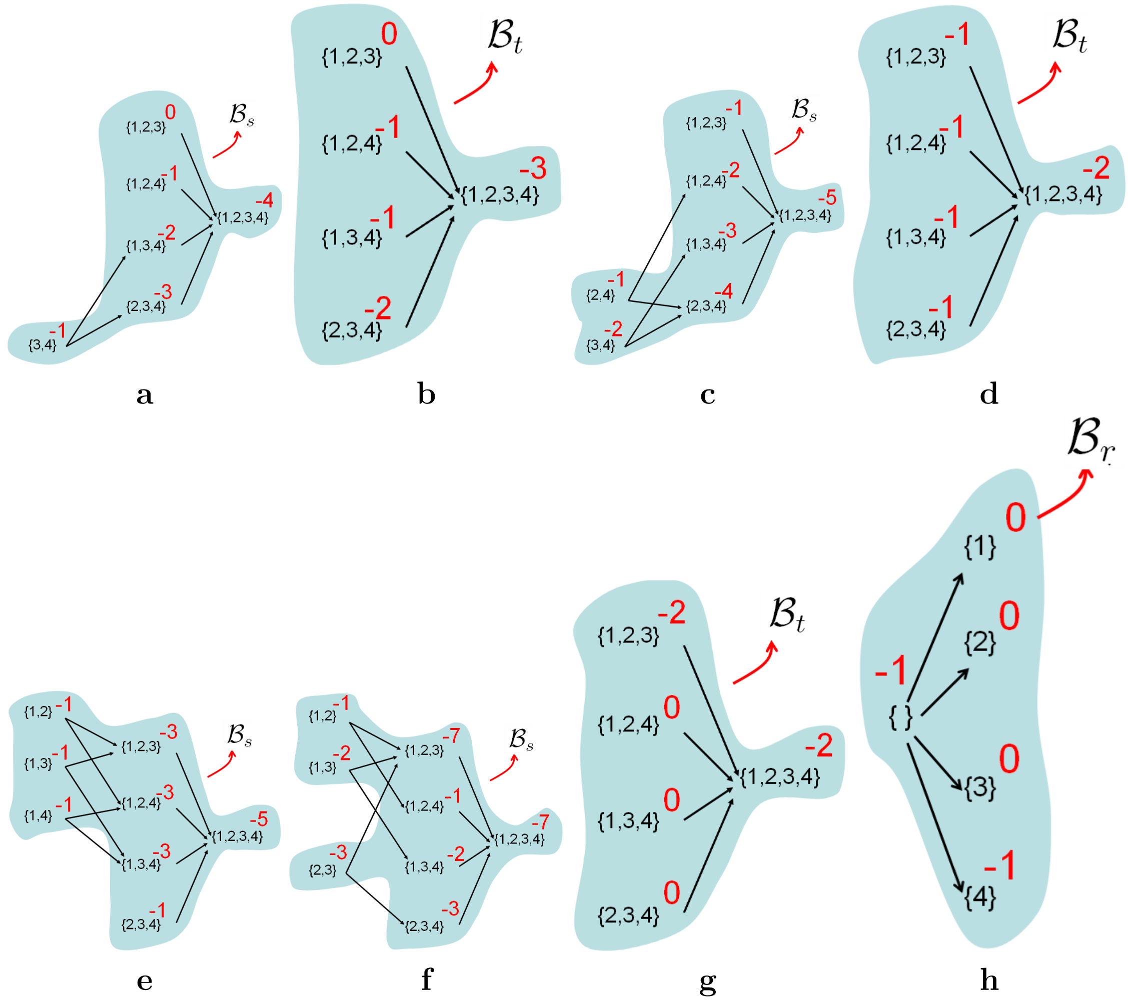

Figure 5.4: *Examples for the four cases in tables 5.3, 5.4, 5.5 and 5.6. In the first case the transition in (a) is mapped to that in (b) and the associated parameters are given by:* $((g_s = 6, g_{s,1} = 1, g_{s,2} = 1, g_{s,3} = 1, g_{s,4} = 1), (g_t = 5, g_{t,1} = 1, g_{t,2} = 2, g_{t,3} = 2, g_{t,4} = 3))$. *The generated pairwise term, independent of* AV*s, is* $-x_3x_4$. *The second case is in (c) and (d) with the parameters* $((5, 1, 2, 3, 4, 5), (2, 1, 1, 1, 1))$ *(shown in the same order as the earlier one) and the pairwise function is* $-x_2x_4 - 2x_3x_4$. *The third case is in (e) and (d) with the parameters* $((5, 4, 1, 1, 1), (2, 1, 1, 1, 1))$ *along with the pairwise function* $-x_1x_2 - x_1x_3 - x_1x_4$. *The final case is in (f), (g) and (h), as the final function has two* AV*s* z_2 *and* z_3. *The function consisting of unary and pairwise terms independent of* AV*s is given by* $1 - x_1 - x_2 - x_3 - x_1x_3 - 2x_2x_3$. *Corresponding parameters are given by:* $((g_s = 8, g_{s,1} = 4, g_{s,2} = 5, g_{s,3} = 6, g_{s,4} = 0), (g_t = 4, g_{t,1} = 2, g_{t,2} = 2, g_{t,3} = 2, g_{t,4} = 0), (g_r = 2, g_{r,1} = 1, g_{r,2} = 1, g_{r,3} = 1, g_{r,4} = 0))$

summing their corresponding partition coefficients we get the following equation:

$$2g_s - g_{s,i} - g_{s,j} - g_{s,k} - g_{s,l} \geq 0, \tag{5.55}$$

Equations 5.54 and 5.55 lead to a contradiction, therefore the lemma holds . □ □

Theorem 3. *Any function $h(\mathbf{x}, z_s)$ in $\mathcal{F}^2$ with z_s associated with $[\mathcal{A}_s, \mathcal{B}_s]$, such that $\forall B \in \mathcal{B}_s$, $|B| \geq 2$, can be transformed to another function $h''(\mathbf{x}, z_f, z_b)$ in $\mathcal{F}^2$ without any $z_f z_b$ terms, where z_f and z_b are* AV *correspond to the forward and backward reference partitions respectively.*

Proof. Our proof by construction takes the form of a two-step procedure. In the first stage every function $h(\mathbf{x}, z_s)$ is transformed to $h'(\mathbf{x}, z_t, z_r)$ where z_t and z_r are associated with the partition $[\mathcal{A}_t, \mathcal{B}_t]$ and the backward partition $[\mathcal{A}_r, \mathcal{B}_r]$ respectively and satisfy the conditions $\mathcal{B}_t \subseteq \mathcal{B}_f$ and $\mathcal{B}_r \subseteq \mathcal{B}_b$. In the second step we use lemma 5 to transform $h'(\mathbf{x}, z_t, z_s)$ to $h''(\mathbf{x}, z_f, z_b)$. In most cases only one partition, either the forward or the backward, is used.

$$\min_{z_s} h_2(\mathbf{x}, z_s) = \min_{z_s} \kappa(s, S) z_s, \forall S \in \mathcal{P} \tag{5.56}$$

$$\min_{z_s} h(\mathbf{x}, z_s) = \sum_{i=1}^{4} a_i x_i + \sum_{i=1}^{4} \sum_{j,i \neq j}^{4} a_{i,j} x_i x_j + \min_{z_t} \kappa(t, S) z_t + \min_{z_r} \kappa(r, S) z_r, \forall S \in \mathcal{P} \tag{5.57}$$

The key idea is to decompose h_2 into functions of unary and pairwise terms involving only $\mathbf{x}$ and functions involving new auxiliary variables z_t and z_r. Consider the condition $|B| \geq 2$. A degenerate case occurs where $|B| \geq 3$; here we can directly use lemma 5 to obtain our desired result. We now consider the cases where at least one set $S \in \mathcal{B}_s$ has cardinality two and show a transformation similar to the general one of (5.57). Tables 5.3, 5.4, 5.5 and 5.6 in the appendix contain details of the decomposition.

After the decomposition the new partitions $[\mathcal{A}_t, \mathcal{B}_t]$ and $[\mathcal{A}_r, \mathcal{B}_r]$ satisfy the conditions $\mathcal{B}_t \subseteq \mathcal{B}_f$ and $\mathcal{B}_r \subseteq \mathcal{B}_b$. To show this, we first consider the case where exactly one set $S \in \mathcal{B}_s$ has a cardinality of 2. There are six such occurrences, and all of them are symmetrical. The transformation for this case is in table 5.3.

Next, consider the case where exactly two sets of cardinality two exist in $\mathcal{B}_s$. Although there are 15 ($\binom{6}{2}$) possible cases, they must all be of the form $\{\{i,j\},\{k,l\}\}$ or $\{\{i,j\},\{j,k\}\}$. The first sub-case is prohibited because the presence of the mutually exclusive pair $\{\{i,j\},\{k,l\}\}$ would not permit any other mutually exclusive pair $\{\{i,k\},\{j,l\}\}$ to exist in $\mathcal{A}_s$ as per lemma 6. The transformation for the latter case is in table 5.4.

Finally, consider the case where exactly three sets of cardinality two exist in $\mathcal{B}_s$. The 20 different occurrences ($\binom{6}{3}$) can be expanded to three different scenarios:$\{\{i,j\},\{i,k\},\{i,l\}\}$, $\{\{i,j\},\{k,l\},\{i,k\}\}$ and $\{\{i,j\},\{j,k\},\{i,k\}\}$. Again, lemma 6 prevents the second scenario $\{\{i,j\},\{k,l\},\{i,k\}\}$ from occurring. The transformations of the first and the third cases are in table 5.5 and 5.6. Example transformations are shown in figure 5.4. □

□

Theorem 4. *Any function $h(\mathbf{x}, z_1, z_2, ...z_k)$ in $\mathcal{F}^2$ that is linear in $\mathbf{z}$ can be transformed to some function $h'(\mathbf{x}, z_f, z_b)$ in $\mathcal{F}^2$ where z_f and z_b correspond to the forward and backward reference partitions respectively.*

Proof. Every z_i is independent of every other z_j due to the absence of bilinear terms $z_i z_j$. Hence, the minimisation under $\mathbf{z}$ can be carried out in any order.

$$\min_{z_i, z_j} h(\mathbf{x}, z_i, z_j) = \min_{z_i} \min_{z_j} h(\mathbf{x}, z_i, z_j) = \min_{z_j} \min_{z_i} h(\mathbf{x}, z_i, z_j) \tag{5.58}$$

Applying lemma 4, followed by theorem 3, for every AV, the function $h(\mathbf{x}, z_1, z_2, z_3, ..., z_k)$ can be transformed into $\hat{h}(\mathbf{x}, \hat{z}_1, \hat{z}'_1, ..., \hat{z}_k, \hat{z}'_k)$ where $\hat{z}_i$ and $\hat{z}'_k$ correspond to the forward and backward reference partitions respectively. In other words, every z_i in the original function is replaced by $\hat{z}_i$ and $\hat{z}'_i$. Note that one reference partition may be sufficient in some cases. Finally we remove all constant AVs to obtain $h'(x_1, x_2, x_3, x_4, z_f, z_b)$ from $\hat{h}$. □ □

5.6.2 Interacting AVs

The earlier theorem shows the transformation when the original function h has no bilinear terms $z_i z_j$. The problem becomes more intricate in the presence of these terms. In the earlier case, we could define partitions using a single variable. Here, it is necessary to consider the partitions using two or more variables. Below, we show the joint partition that can solve the transformation with interactions between the AVs. We refer to this as the *matroidal generators*, since the associated partitions satisfy matroid constraints(See appendix).

Definition 10. *The matroidal generators associated with two* AV*s z_{j1} and z_{j2} for expressing all graph-representable fourth order functions is given below:*

$$B \in \mathcal{B}_{j1} \iff |B| \geq 3, \qquad \mathcal{A}_{j1} = \mathcal{P} \backslash \mathcal{B}_{j1} \tag{5.59}$$

$$B \in \mathcal{B}_{j2} \iff |B| \geq 2, \qquad \mathcal{A}_{j2} = \mathcal{P} \backslash \mathcal{B}_{j2} \tag{5.60}$$

In Figure 5.2 we show the matroidal generators for fourth order functions. These partitions are same as the reference partitions studied earlier. The expressive power of these AVs are enhanced by interaction or the usage of the bilinear term $z_{j1} z_{j2}$.

Theorem 5. *Any function $h(\mathbf{x}, z_1, z_2, ...z_k)$ in $\mathcal{F}^2$ that has bilinear terms $z_i z_j$ can be transformed to some function $h'(\mathbf{x}, z_{j1}, z_{j1})$ in $\mathcal{F}^2$.*

Proof. The basic idea of the proof is to decompose a given fourth order function using the result of (Promislow and Young, 2005) and show that all the spawned MBFs can be expressed by the matroidal generators. Using Theorem 5.2 from (Promislow and Young, 2005) we can decompose a given submodular function in $\mathcal{F}^4$ into 10 different groups $\mathcal{G}_i, i = \{1..10\}$ where each $\mathcal{G}_i$ is in Table 5.1.

Each group $\mathcal{G}_i$ contains three or four functions giving rise to a total of 30 or more different functions. Prior work uses one auxiliary variable for every function, whereas we will show that the two AVs corresponding to the matroidal generators are sufficient to simultaneously model all these functions. As shown in (Živný and Jeavons, 2008) the functions in $\mathcal{G}_{10}$ are not graph-representable. Note that the functions in $\mathcal{G}_{10}$ does not become graph-representable when combined with other generators of $\mathcal{F}^4$ according to Theorem 16(3) in (Živný and Jeavons, 2008). We also observe that these functions are not representable by both non-interacting and interacting AVs. Thus the largest subclass $\mathcal{F}_2^k$ should be composed of functions in the remaining 9 groups.

As the functions present in the groups $\mathcal{G}_i, i = \{1..8\}$ do not require bilinear AVterms, any sum of functions in $\mathcal{G}_i, i = \{1..8\}$ can be expressed with only two AVs z_f and z_b according to Theorem 4. We consider the functions in $\mathcal{G}_9$. The sum of functions in this group may lead to two alternatives. The union of functions in $\mathcal{G}_9$ may either result in a function in $\mathcal{G}_9$ or a function that uses the AVs z_f and z_b. Any function in $\mathcal{G}_9$ can be expressed using two AVs z_{91} and z_{92} (Živný et al., 2009). As a result, the sum of functions in $\mathcal{G}_i, i = \{1..9\}$ can be expressed using four AVs $(z_f, z_b, z_{91}, z_{91})$. These four AVs could be merged into two AVs z_{j1} and z_{j2} in the matroidal generators as shown in Figure 5.2.

Hence, all functions in $\mathcal{G}_i, i = \{1..9\}$ can be expressed by the matroidal generators. □ □

Group	$f(\mathbf{x})$	$\min_{z_1,z_2} h(\mathbf{x},z_1,z_2)$ where $h(\mathbf{x},z_1,z_2) \in \mathcal{F}$
$\mathcal{G}_1$	$-x_ix_j$	$-x_ix_j$
$\mathcal{G}_2$	$-x_ix_jx_k$	$\min_z(2 - x_i - x_j - x_k)$
$\mathcal{G}_3$	$-x_1x_2x_3x_4$	$\min_z(3 - x_1 - x_2 - x_3 - x_4)$
$\mathcal{G}_4$	$-x_1x_2x_3x_4 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 - x_1x_2 - x_1x_3 - x_1x_4 - x_2x_3 - x_2x_4 - x_3x_4$	$\min_z(z(1 - x_1 - x_2 - x_3 - x_4))$
$\mathcal{G}_5$	$x_ix_jx_kx_l - x_ix_jx_k - x_ix_l - x_jx_l - x_kx_l$	$\min_z(z(2 - x_i - x_j - x_k - 2x_l)$
$\mathcal{G}_6$	$x_ix_jx_k - x_ix_j - x_ix_k - x_jx_k$	$\min_z(z(1 - x_i - x_j - x_k))$
$\mathcal{G}_7$	$x_ix_jx_kx_l - x_ix_jx_k - x_ix_jx_l - x_ix_kx_l$	$\min_z(z(3 - 2x_i - x_j - x_k - x_l))$
$\mathcal{G}_8$	$2x_1x_2x_3x_4 - x_1x_2x_3 - x_1x_2x_4 - x_1x_3x_4 - x_2x_3x_4$	$\min_z(z(2 - x_1 - x_2 - x_3 - x_4))$
$\mathcal{G}_9$	$x_ix_jx_kx_l - x_ix_j - x_ix_k - x_ix_kx_l - x_jx_kx_l$	$\min_{z_1,z_2}(z_1 + 2z_2 - z_1z_2 - z_1x_i - z_1x_j - z_2x_k - z_2x_l)$
$\mathcal{G}_{10}$	$-x_ix_jx_kx_l + x_ix_kx_l + x_jx_kx_l - x_ix_k - x_ix_l - x_jx_k - x_jx_l - x_kx_l$	$f(\mathbf{x}) \ni \mathcal{F}_2^4$ as shown in (Živný and Jeavons, 2

Table 5.1: *The above table is adapted from Figure 2 of (Zivny and Jeavons, 2008) where $\{i,j,k,l\} = S_4$. Each group has several terms depending on the values of $\{i,j,k,l\}$. As the groups $\mathcal{G}_4$ and $\mathcal{G}_8$ are symmetric with respect to $\{i,j,k,l\}$; they contain one function each.*

5.7 Linear Programming solution

For a given function $f(x_1,x_2,x_3,x_4)$ in $\mathcal{F}_s^4$, our goal is to compute a function $h(\mathbf{x},\mathbf{z})$ in $\mathcal{F}^2$. As a result of theorem 5 we only need to solve the case with two AVs (z_{j1}, z_{j2}) associated with the matroidal generators. The required function $h(\mathbf{x},\mathbf{z})$ is:

$$h(\mathbf{x}, z_{j1}, z_{j2}) = b_0 + \sum_i b_i x_i - \sum_{i>j} b_{ij} x_i x_j - (g_{j1} - \sum_{i=1}^{4} g_{j1,i} x_i) z_{j1} + (g_{j2} - \sum_{i=1}^{4} g_{j2,i} x_i) z_{j2} - j_{12} z_{j1} z_{j2}. \tag{5.61}$$

such that $b_{ij}, g_{j1,i}, g_{j2,i}, j_{12} \geq 0$ and $i,j \in S_4$. As we know the partition of (z_{j1}, z_{j2}) we know their Boolean values for all labellings of $\mathbf{x}$. We need the coefficients $(b_i, b_{ij}, j_{12}, g_{j1}, g_{j2}, g_{j1,i}, g_{j2,i}), i = S_4$ to compute $h(x_1,x_2,x_3,x_4,z_{j1},z_{j2})$. These coefficients satisfy both submodularity constraints(that the coefficients of all bilinear terms $(x_ix_j, x_iz_{j1}, x_jz_{j2}, z_{j1}z_{j2})$ are less than or equal to zero) and those imposed by the reference partitions. First we list these conditions below:

$$\underbrace{\begin{pmatrix} b_{ij} \\ g_{j1,i} \\ g_{j2,i} \\ j_{12} \end{pmatrix}^T}_{\mathcal{S}_p} \geq \mathbf{0}, i, j = S_4, i \neq j \tag{5.62}$$

where **0** refers of a vector composed 0's of appropriate length. Next we list the conditions which guarantee $f(\mathbf{x}) = \min_{z_{j1},z_{j2}} h(\mathbf{x}, z_{j1}, z_{j2})$ for all $\mathbf{x}$. Let $\forall S \in \mathcal{P}$, and let the value of $z_{j1}z_{j2}$ for different subsets S be given by $\eta(S)$. As we know the partition functions of both z_{j1} and z_{j2} it is easy to find this. Let $\mathcal{G}$ and $\mathcal{H}$ denote values of f and h for different S:

$$\mathcal{G} = f(\mathbf{1}_1^S, \mathbf{1}_2^S, \mathbf{1}_3^S, \mathbf{1}_4^S) \tag{5.63}$$

$$\mathcal{H} = h(\mathbf{1}_1^S, \mathbf{1}_2^S, \mathbf{1}_3^S, \mathbf{1}_4^S, 0, 0) - (g_{j1} - \sum_{i=1}^{4} g_{j1,i}\mathbf{1}_i^S) - (g_{j2} - \sum_{i=1}^{4} g_{j2,i}\mathbf{1}_i^S) - j_{12}\eta(S) \tag{5.64}$$

As a result we have the following 16 linear equations (N.B. there are $2^4(16)$ different S):

$$\mathcal{G} = \mathcal{H}, \forall S \in \mathcal{P} \tag{5.65}$$

Note that as with section 5.6 we do not make use of either auxiliary variables or the min operator over $\mathcal{H}$. Again, this because we already know the partition of (z_{j1}, z_{j2}) and their appropriate values a priori. This can be seen as (5.65) need not hold if z_{j1} and z_{j2} do not lie in the reference partitions.

$$\underbrace{\begin{pmatrix} g_f - \sum_{i=1}^{4} g_{f,i}\mathbf{1}_i^S \\ g_b - \sum_{i=1}^{4} g_{b,i}\mathbf{1}_i^D \end{pmatrix}}_{\mathcal{G}_g} \geq \mathbf{0}, S \in \mathcal{A}_{j1}, D \in \mathcal{A}_{j2}$$

$$\underbrace{\begin{pmatrix} g_f - \sum_{i=1}^{4} g_{f,i}\mathbf{1}_i^S \\ g_b - \sum_{i=1}^{4} g_{b,i}\mathbf{1}_i^D \end{pmatrix}}_{\mathcal{G}_l} \leq \mathbf{0}, S \in \mathcal{B}_{j1}, D \in \mathcal{B}_{j2}.$$

Essentially we need to compute the coefficients $(b_{ij}, g_{j1}, g_{j1,i}, g_{j2}, g_{j2,i}, j_{12})$ that satisfy the equations (5.62,5.65,5.66) This is equivalent to finding a feasible point in a linear programming problem:

$$\min\ const \tag{5.66}$$

$$s.t\ \mathcal{S}_p \geq \mathbf{0},\ \mathcal{G} = \mathcal{H},\ \mathcal{G}_g \geq \mathbf{0},\ \mathcal{G}_l \leq \mathbf{0} \tag{5.67}$$

As discussed in section 5.5, by using a different cost function we can formulate a problem to to compute a function in $\mathcal{F}^2$ closest to a given arbitrary fourth-order function.

5.8 Discussion and open problems

We observe that the basis MBFs corresponding to reference partitions always satisfy matroid constraints (See appendix). It can be easily shown that for $k = 3$ there is only one reference partition corresponding to a uniform matroid $\mathcal{U}_1$. When $k = 4$ we have two reference partitions corresponding to uniform matroids $\mathcal{U}_1$ and $\mathcal{U}_2$. Thus we conjecture that we can transform a large subclass, possibly the largest, of $\mathcal{F}_2^k$ using $k-2$ matroidal generators. Each of these generators correspond to uniform matroids $\mathcal{U}_1$,$\mathcal{U}_2$,$\mathcal{U}_3$, ...,$\mathcal{U}_{k-2}$. We do not have any proof for this result. However, our intuition is based on the following reasons:

- The reference partitions for $k = 3$ and $k = 4$ are symmetrical with respect all x_i variables.
- The reference partitions correspond to only distinct uniform matroids.
- We can only transform a subclass of all submodular functions of order k. Using the result of Zivny et al., we know that when $k \geq 4$, not all submodular functions can be transformed to a quadratic PBF.
- Although we use only a linear number of auxiliary variables, the underlying function is powerful as we employ all possible interactions among the auxiliary variables. Each of these intersection can be seen as the intersection of two uniform matroids.

5.9 Appendix

i	$\min(0, \kappa(s, \{i,j,k\}))$	$\min(0, \kappa(s, \{i,j,l\}))$	$\min(0, \kappa(s, \{i,j,k,l\}))$	$\kappa(f, \{i,j\})$
1	0	0	0	0
2	0	$\kappa(s, \{i,j,l\})$	$\kappa(s, S_4)$	$g_{s,k}$
3	$\kappa(s, \{i,j,k\})$	0	$\kappa(s, \{i,j,k,l\})$	$g_{s,l}$
4	$\kappa(s, \{i,j,k\})$	$\kappa(s, \{i,j,l\})$	$\kappa(s, \{i,j,k,l\})$	$\kappa(s, \{i,j\})$

Table 5.2: **See lemma 5.** *In all four cases $\kappa(f, \{i,j\})$ is non-negative. This result holds for the fourth case as $\kappa(s, \{i,j\}) \geq 0$.*

5.9.1 Definitions

Definition 11. *A matroid $\mathcal{M}$ is an ordered pair $(E, \mathcal{I})$ consisting on a finite set E and a family of subsets $\mathcal{I}$ of E satisfying the following conditions:*

1. *$\emptyset \in \mathcal{I}$.*
2. *If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$.*

Case 1:($\{i,j\} \in \mathcal{B}_s$.

$$h_2 = \kappa(s,\{i,j\})x_ix_j + (\underbrace{(2 * g_s - g_{s,i} - g_{t,j})}_{g_t} - \underbrace{(g_s - g_{s,i})}_{g_{t,j}} x_i - \underbrace{(g_s - g_{s,j})}_{g_{t,i}} x_j - \underbrace{g_{s,k}}_{g_{t,k}} x_k - \underbrace{g_{s,l}}_{g_{t,l}} x_l)z_t$$

S	$\kappa(t,S)$	$S \in \mathcal{A}_t$ or $S \in \mathcal{B}_t$
$\{i,j\}$	0	$S \in \mathcal{A}_t$
$\{i,k\}$	$\kappa(s,\{j,k\})$	$S \in \mathcal{A}_t$ since $\{j,k\} \in \mathcal{A}_s$
$\{k,l\}$	$\kappa(s,\{i,k\}) + \kappa(s,\{j,l\})$	$S \in \mathcal{A}_t$ since $\{i,k\},\{j,l\} \in \mathcal{A}_s$
$\{i,j,k\}$	$-g_{s,k}$	$S \in \mathcal{B}_t$
$\{i,k,l\}$	$\kappa(s,\{j,k,l\})$	$S \in \mathcal{B}_t$ since $\{j,k,l\} \in \mathcal{B}_s$

Table 5.3: **See theorem 3.** *Case 1: The details of the transformation (similar to one in equation (5.57)) are shown for a scenario where exactly one set ($\{i,j\}$) with cardinality two exist in $\mathcal{B}_s$. We prove that after the transformation all the sets S with $|S| = 2$ exist in $\mathcal{A}_t$ and $|S| \geq 3$ exist in $\mathcal{B}_t$. Although the reduction is illustrated for only a few cases, they are representative of the remainder.*

Case 2:$\{i,j\},\{j,k\} \in \mathcal{B}_s$.

$$h_2 = \kappa(s,\{i,j\})x_ix_j + \kappa(s,\{j,k\})x_jx_k + (\underbrace{3g_s - 2g_{s,j} - g_{s,i} - g_{s,k}}_{g_t} - \underbrace{(g_s - g_{s,j})}_{g_{t,i}} x_i - \underbrace{(2g_s - g_{s,i} - g_{s,j} - g_{s,k})}_{g_{t,j}} x_j - \underbrace{(g_s - g_{s,j})}_{g_{t,k}} x_k - (\underbrace{g_{s,l}}_{g_{t,l}} x_l)z_t$$

S	$\kappa(t,S)$	$S \in \mathcal{A}_t$ or $S \in \mathcal{B}_t$
$\{i,j\}$	0	$S \in \mathcal{A}_t$
$\{i,l\}$	$\kappa(s,\{i,k\}) + \kappa(s,\{j,l\})$	$S \in \mathcal{A}_t$ since $\{i,k\},\{j,l\} \in \mathcal{A}_s$
$\{j,l\}$	$\kappa(s,\{j\}) + g_{s,l}$	$S \in \mathcal{A}_t$ since $\{j\} \in \mathcal{A}_s$ and $g_{s,l} \geq 0$
$\{i,j,k\}$	$-\kappa(s,\{j\})$	$S \in \mathcal{B}_t$ since $\{j\} \in \mathcal{A}_s$
$\{i,k,l\}$	$\kappa(s,\{i,k,l\})$	$S \in \mathcal{B}_t$ if $\{i,k,l\} \in \mathcal{B}_s$
$\{i,j,l\}$	$-g_{s,l}$	$S \in \mathcal{B}_t$

Table 5.4: **See theorem 3.***Case 2: We study the scenario where exactly two sets with cardinality two $\{\{i,j\},\{j,k\}$) occur in $\mathcal{B}_s$. Note that all other cases either can not happen (according to lemma 6) or similar to the ones shown in this table. We also prove that after the transformation all the sets S with $|S| = 2$ exist in $\mathcal{A}_t$ and $|S| \geq 3$ exist in $\mathcal{B}_t$.*

Case 3: $\{i,j\},\{i,k\},\{i,l\} \in \mathcal{B}_s$.
$h_2 = \kappa(s,\{i,j\})x_ix_j + \kappa(s,\{i,k\})x_ix_k + \kappa(s,\{i,l\})x_ix_l + (\underbrace{\min(0,\kappa(s,\{j,k,l\}) + 3(g_s - g_{s,i})}_{g_t} - \underbrace{\min(0,\kappa(s,\{j,k,l\})) + 2(g_s - g_{s,i})}_{g_{t,i}} x_i - \underbrace{(g_s - g_{s,i})}_{g_{t,j}} x_j - \underbrace{(g_s - g_{s,i})}_{g_{t,k}} x_k - \underbrace{(g_s - g_{s,i})}_{g_{t,l}} x_l)z_t$

S	$\kappa(t,S)$	$S \in \mathcal{A}_t$ or $S \in \mathcal{B}_t$
$\{i,j\}$	0	$S \in \mathcal{A}_t$
$\{j,k\}$	$\min(0,\kappa(s,\{j,k,l\})) + (g_s - g_{s,i})$	$S \in \mathcal{A}_t$ since $\{i\} \in \mathcal{A}_s$
$\{i,j,k\}$	$-\kappa(s,\{j\})$	$S \in \mathcal{B}_t$ since $\{j\} \in \mathcal{A}_s$
$\{i,k,l\}$	$\kappa(s,\{i,k,l\})$	$S \in \mathcal{B}_t$ if $\{i,k,l\} \in \mathcal{B}_s$
$\{i,j,l\}$	$-g_{s,l}$	$S \in \mathcal{B}_t$

Table 5.5: **See theorem 3.** *Case 3: Here we study the scenario where exactly three sets with cardinality two $\{\{i,j\},\{i,k\},\{i,l\})$ exist in $\mathcal{B}_s$. The only other case where three sets can exist is shown in table 5.6. The shown cases are generalisations of all the possible cases that can occur without violating lemma (6). We prove that after the transformation all the sets S with $|S| = 2$ exist in $\mathcal{A}_t$ and $|S| \geq 3$ exist in $\mathcal{B}_t$.*

Case 4: $\{i,j\},\{i,k\},\{i,l\} \in \mathcal{B}_s$.
$h_2 = \kappa(s,\{i,j\})(1 - x_i - x_j - x_k) - (g_{s,k} - g_{s,j})x_ix_k - (g_{s,k} - g_{s,i})x_jx_k + (\underbrace{2(g_s - g_{s,k})}_{g_t} - \underbrace{(g_s - g_{s,k})}_{g_{t,i}} x_i - \underbrace{(g_s - g_{s,k})}_{g_{t,j}} x_j - \underbrace{(g_s - g_{s,k})}_{g_{t,k}} x_k - \underbrace{g_{s,l}}_{g_{t,l}} x_l)z_t + (\underbrace{-2\kappa(s,\{i,j))}_{g_r} - \underbrace{(-\kappa(s,\{i,j)))}_{g_{r,i}}(1 - x_i) - \underbrace{(-\kappa(s,\{i,j)))}_{g_{r,j}}(1 - x_j) - \underbrace{(-\kappa(s,\{i,j)))}_{g_{r,k}}(1 - x_k) - \underbrace{0}_{g_{r,l}}(1 - x_l))z_r$

S	$\kappa(t,S)$	$S \in \mathcal{A}_t$ or $S \in \mathcal{B}_t$
$\{i,j\}$	0	$S \in \mathcal{A}_t \kappa = 0$
$\{i,l\}$	$\kappa(s,\{k,l\})$	$S \in \mathcal{A}_t$ since $\{k,l\} \in \mathcal{A}_s$
$\{i,j,k\}$	$-\kappa(s,\{k\})$	$S \in \mathcal{B}_t$ since $\{k\} \in \mathcal{A}_s$
$\{i,j,l\}$	$-g_{s,l}$	$S \in \mathcal{B}_t$ since $g_{s,l} \geq 0$

S	$\kappa(r,S)$	$S \in \mathcal{A}_r$ or $S \in \mathcal{B}_r$
$\{i,l\}$	0	$S \in \mathcal{A}_r$
$\{i,j\}$	$-\kappa(s,\{i,j\})$	$S \in \mathcal{A}_r$ since $\{i,j\} \in \mathcal{B}_s$
$\{i\}$	0	$S \in \mathcal{B}_r$
$\{l\}$	$\kappa(s,\{i,j\})$	$S \in \mathcal{B}_r$ since $\{i,j\} \in \mathcal{B}_s$

Table 5.6: **See theorem 3.** *Case 4: We consider three sets $\{i,j\},\{i,k\},\{j,k\} \in \mathcal{B}_s$ which involve only three elements and all three repeating in more than one set. Without loss of generality, we assume that $\kappa(s,\{i,j\}) \geq \kappa(s,\{i,k\})$ and $\kappa(s,\{i,j\}) \geq \kappa(s,\{j,k\})$. In this case we replace the* AV *z_s using two variables z_t and z_r.*

3. *If I_1 and I_2 are in $\mathcal{I}$ and $|I_1| < |I_2|$, then there is an element e of $I_2 - I_1$ such that $I_1 \cup e \in \mathcal{I}$.*

The maximal independent set in a matroid is called the base of a matroid. All the bases of a matroid are equicardinal,i.e., they have the same number of elements.

Definition 12. *The dual matroid of $\mathcal{M}$ is given by $\mathcal{M}^*$ whose bases are the complements of the bases of $\mathcal{M}$.*

Definition 13. *In a uniform matroid $\mathcal{U}_n(E, \mathcal{I})$, all the independent sets $I_i \in \mathcal{I}$ satisfy the condition that $|I_i| \leq n$ for some fixed n.*

Chapter 6

Conclusion

> The purpose of numeric computing is not yet in sight.
>
> **RW Hamming**

This thesis covers several novel techniques of inference, which have already shown themselves to be of importance to the vision community. The analysis shown in chapter 3 provides the only finite bounds, currently published, of the P^n and robust P^n models (Kohli et al., 2007, 2009), both of which, are widely used in vision. The inference techniques described in this chapter also provide the backbone of the Associative Hierarchical Networks, which have remained state of the art on object-class segmentation data-sets including CamVid (see Sturgess et al. (2009) for details), and the MSRC-data-set (see Ladicky et al. (2009), or chapter 2 for details) which contain 'stuff' annotations, such as 'road', or 'grass', and our approach has been consistently competitive on data-sets such as VOC(Everingham et al., 2009) which lack these labels.

The work on co-occurrence potentials described in chapter 4, is already showing its importance outside of semantic segmentation. By restricting these co-occurrence potentials to local neighbourhoods, we have been able to propose a new form of MRF, defined over sets of labels. Application of this new form of model, to the problems of Non-rigid Structure from Motion (Russell et al., 2011), articulated motion (Fayad et al., 2011), and kernel learning (unpublished work), has given state of the art performance in all domains. Beyond this, its extension to more complex forms of Minimum description length (MDL) cost such as robust Geometric Information Criteria (Torr et al., 1999), is straightforward, and already proving valuable in the dense reconstruction of piecewise rigid scenes.

One of the strengths of these novel graph-cut based potentials, is their composability. The fact that texture-based potentials, detections, and co-occurrence can be integrated into a single robust framework provides a substantial advantage, and in particular the combination of detectors with co-occurrence potentials is much stronger than the sum of its parts.

One of the more surprising facts that has come to light, since the completion of my thesis is that the pairwise formulations of both AHNs, P^n type potentials, and the restricted class co-occurrence potentials as described by Delong et al. (2010), belong to the same family of pairwise cost which we call *near Potts* models. Formulation of costs in this manner allows us to tighten the approximation bound to 2. This means that much of the expanded class of potentials discussed in chapters 2 — 4 essential comes for free, with no weakening of the standard guarantees of α-expansion over the Potts model.

Despite, its less immediate applications, the techniques discussed in the final chapter open the door to the automated discovery of cost functions such as those used in earlier chapters. One future avenue for research, is the combination of this automated potential discovery, with learning techniques, such as Taskar et al. (2004); Alahari et al. (2010), which are normally used to learn the optimal weights assigned to pre-given potentials. However, finding a compact representation in the general case remains an open problem

Finally, it is worth emphasising that one of the strongest results of this thesis is a negative. Although much care was taken in chapters 3 and 4 to formulate the problem of inference in such a way that other message passing and move-making algorithms could be used; in every case graph-cuts, and in particular variants of α-expansion, strongly outperformed all other algorithms.

Bibliography

Adelson, E. H. (2001). On seeing stuff. In *Proceedings of the the International Society for Optics and Photonics*, pages 1–12. 42

Alahari, K., Russell, C., and Torr, P. H. S. (2010). Efficient piecewise learning for conditional random fields. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* 8, 136

Batra, D., Sukthankar, R., and Tsuhan, C. (2008). Learning class-specific affinities for image labelling. In *IEEE Computer Vision and Pattern Recognition.* 38, 40

Bengio, Y., Lamblin, P., Popovici, P., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems*, 19. 39

Benson, H. Y. and Shanno, D. F. (2007). An exact primal—dual penalty method approach to warmstarting interior-point methods for linear programming. *Computational Optimization and Applications*, 38(3):371–399. 95

Besag, J. (1975). Statistical Analysis of Non-Lattice Data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):179–195. 6

Besag, J. (1986). On the statisical analysis of dirty pictures. *Journal of the Royal Statistical Society B.* 6

Billionnet, A. and Minoux, M. (1985). Maximizing a supermodular pseudo-boolean function: A polynomial algorithm for supermodular cubic functions. *Discrete Applied Mathematics*, 12(1):1 – 11. 100, 101, 103, 110

Borenstein, E. and Malik, J. (2006). Shape guided object segmentation. In *IEEE Computer Vision and Pattern Recognition (1)*, pages 969–976. 72

Boros, E. and Hammer, P. (2002). Pseudo-boolean optimization. *Discrete Applied Mathematics*. 97

Boykov, Y. and Kolmogorov, V. (2004). An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137. 19

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001. 13, 15, 22, 25, 38, 47, 54, 57, 58, 78, 80, 85

Chekuri, C., Khanna, S., Naor, J., and Zosin, L. (2005). A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal of Discrete Mathematics*, 18(3):608–625. 26

Choi, M. J., Lim, J. J., Torralba, A., and Willsky, A. S. (2010). Exploiting hierarchical context on a large database of object categories. In *IEEE Computer Vision and Pattern Recognition*. 73

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 73

Csurka, G. and Perronnin, F. (2008). A simple high performance approach to semantic segmentation. In *British Machine Vision Conference*. 77, 78, 96

Cunningham, W. (1985). On submodular function minimization. *Combinatorica*, 5:185–192. 10.1007/BF02579361. 99

Dahlhaus, E., Johnson, D. S., Papadimitriou, C. H., Seymour, P. D., and Yannakakis, M. (1994). The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894. 12, 14

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Vision and Pattern Recognition*, pages 886–893. 10

Delong, A., Osokin, A., Isack, H., and Boykov, Y. (2010). Fast approximate energy minimization with label costs. *IEEE Computer Vision and Pattern Recognition*. 4, 79, 84, 136

Edmonds, J. (2003). Submodular functions, matroids, and certain polyhedra. In Jünger, M., Reinelt, G., and Rinaldi, G., editors, *Combinatorial optimization - Eureka, you shrink!*, pages 11–26. Springer-Verlag New York, Inc., New York, NY, USA. 99

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2009). The PASCAL Visual Object Classes Challenge (VOC) Results. 45, 135

Fayad, J., Russell, C., and Agapito, L. (2011). Automated articulated structure and 3d shape recovery from point correspondences. In *International Conference on Computer Vision*. 136

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *IJCV*. 73

Felzenszwalb, P. F., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE Computer Vision and Pattern Recognition*. 43, 45

Fleischer, L. and Iwata, S. (2003). A push-relabel framework for submodular function minimization and applications to parametric optimization. *Discrete Applied Mathematics*, 131:311–322. 99

Galleguillos, C., Rabinovich, A., and Belongie, S. (2008). Object categorization using co-occurrence, location and appearance. In *IEEE Computer Vision and Pattern Recognition*. 37, 38, 77, 78

Gallo, G., Grigoriadis, M., and Tarjan, R. (1989). A fast parametric maximum flow algorithm and applications. *SIAM J. on Comput.*, 18:18:30–55. 44, 103

Gould, S., Amat, F., and Koller, D. (2009a). Alphabet soup: A framework for approximate energy minimization. In *IEEE Computer Vision and Pattern Recognition*, pages 903–910. 26, 53, 57

Gould, S., Fulton, R., and Koller, D. (2009b). Decomposing a scene into geometric and semantically consistent regions. In *Proceeding of International Conference on Computer Vision (ICCV)*. 32, 34

Gould, S., Gao, T., and Koller, D. (2009c). Region-based segmentation and object detection. In *NIPS*. 34

Grötschel, M., Lovász, L., and Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197. 99

Hammer, P. (1965). Some network flow problems solved with pseudo boolean programming. *Operations Research,*, 13. 100

He, X., Zemel, R. S., and Carreira-Perpiñán, M. Á. (2004). Learning and incorporating top-down cues in image segmentation. In *IEEE Computer Vision and Pattern Recognition*, volume 2, pages 695–702. 42

Heitz, D. K. G. (2008). Learning spatial context: Using stuff to find things. In *European Conference on Computer Vision*. 72

Hinton, G. E., Osindero, S., and Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18. 39

Hoiem, D., Rother, C., and Winn, J. M. (2007). 3d layoutcrf for multi-view object class recognition and segmentation. In *IEEE Computer Vision and Pattern Recognition*. 4, 79

Ishikawa, H. (2003). Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336. 13, 58, 87, 104

Ishikawa, H. (2009). Higher-order clique reduction in binary graph cut. In *IEEE Conference on Computer Vision and Pattern Recognition*. 99

Iwata, S. (2002). A fully combinatorial algorithm for submodular function minimization. *Journal Comb. Theory Ser. B*, 84:203–212. 99

Iwata, S. (2003). A faster scaling algorithm for minimizing submodular functions. *SIAM J. Computing*. 99

Iwata, S., Fleischer, L., and Fujishige, S. (2001). A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal ACM*, 48:761–777. 99

Kleinberg, J. M. and Tardos, É. (1999). Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. In *FOCS*, pages 14–23. 26, 83

Kleitman, D. (1969). On dedekind's problem: The number of boolean functions. *Amer. Math Society*. 108

Kohli, P., Kumar, M., and Torr, P. (2007). P^3 and beyond: Solving energies with higher order cliques. In *IEEE Computer Vision and Pattern Recognition*. 17, 18, 27, 47, 50, 53, 57, 99, 135

Kohli, P., Ladicky, L., and Torr, P. (2008). Robust higher order potentials for enforcing label consistency. In *IEEE Computer Vision and Pattern Recognition*. 12, 34, 35, 47, 50, 53, 54

Kohli, P., Ladicky, L., and Torr, P. H. S. (2009). Robust higher order potentials for enforcing label consistency. *IJCV*, 82:302–324. 12, 17, 18, 27, 47, 135

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583. 47, 52, 80, 85

Kolmogorov, V. and Rother, C. (2006). C.: Comparison of energy minimization algorithms for highly connected graphs. in: European conference on computer vision. In *In Proc. European Conference on Computer Vision*, pages 1–15. 52, 80

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts?. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19, 25, 100, 110

Komodakis, N. and Paragios, N. (2008). Beyond loose lp-relaxations: Optimizing mrfs by repairing cycles. In *European Conference on Computer Vision*. 47

Komodakis, N. and Paragios, N. (2009). Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *CVPR09*, pages 2985–2992. 47, 66

Komodakis, N., Tziritas, G., and Paragios, N. (2007). Fast, approximately optimal solutions for single and dynamic mrfs. In *IEEE Computer Vision and Pattern Recognition*. 82

Korshunov, A. D. (1981). The number of monotone boolean functions. *Problemy Kibernet*. 102, 108

Kschischang, F. R., Member, S., Frey, B. J., and andrea Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519. 50

Kumar, M. and Torr, P. (2008a). Efficiently solving convex relaxations for map estimation. In *ICML*. 47, 82, 88

Kumar, M. P. and Koller, D. (2009). MAP estimation of semi-metric MRFs via hierarchical graph cuts. In *Proceedings of the Conference on Uncertainity in Artificial Intelligence*. 15, 47

Kumar, M. P. and Koller, D. (2010). Efficiently selecting regions for scene understanding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 34

Kumar, M. P. and Torr, P. H. S. (2008b). Improved moves for truncated convex models. In *Proceedings of Advances in Neural Information Processing Systems*. 15, 22, 47, 58

Kumar, S. and Hebert, M. (2005). A hierarchical field framework for unified context-based classification. In *International Conference on Computer Vision*. 38

Kumar, S. and Hebert, M. (2006.). Discriminative random fields. In *International Journal of Computer Vision (IJCV)*, volume 68(2), pages 179–201. 10

Ladicky, L., Russell, C., Kohli, P., and Torr, P. (2010a). Graph cut based inference with co-occurrence statistics. In *European Conference on Computer Vision*. springer.

Ladicky, L., Russell, C., Kohli, P., and Torr, P. H. (2009). Associative hierarchical crfs for object class image segmentation. In *International Conference on Computer Vision*. 4, 10, 12, 31, 41, 52, 65, 70, 74, 93, 135

Ladicky, L., Russell, C., Sturgess, P., Alahari, K., and Torr, P. (2010b). What, where and how many? combining object detectors and crfs. *European Conference on Computer Vision*. 4, 93

Ladicky, L., Russell, C., Sturgess, P., Alahri, K., and Torr, P. (2010c). What, where and how many? combining object detectors and crfs. In *European Conference on Computer Vision*. IEEE. 31, 47

Ladicky, L., Sturgess, P., Russell, C., Sengupta, S., Clockson, Y. B. W., and Torr, P. H. (2010d). Joint optimisation for object class segmentation and dense stereo reconstruction. *British Machine Vision Conference*. 7

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In *ICML*. 6, 12, 35, 72

Lan, X., Roth, S., Huttenlocher, D., and Black, M. (2006). Efficient belief propagation with learned higher-order markov random fields. In *European Conference on Computer Vision (2)*, pages 269–282. 47, 53, 99

Larlus, D. and Jurie, F. (2008). Combining appearance models and markov random fields for category level object segmentation. In *IEEE Computer Vision and Pattern Recognition*. 43, 72

Lim, J. J., Arbelez, P., Gu, C., and Malik, J. (2009). Context by region ancestry. In *International Conference on Computer Vision*. 38

Lovász, L. (1983). *Submodular functions and convexity*, pages 235–257. Springer, Berlin. 98

Narasimhan, M. and Bilmes, J. A. (2005). A submodular-supermodular procedure with applications to discriminative structure learning. In *Uncertainty in Artificial Intelligence*, pages 404–412. 90, 114

Nowozin, S., Gehler, P. V., and Lampert, C. H. (2010). On parameter learning in crf-based approaches to object class image segmentation. In *European Conference on Computer Vision*. 38

Orlin, J. B. (2007). A faster strongly polynomial time algorithm for submodular function minimization. In *Proceedings of the 12th international conference on Integer Programming and Combinatorial Optimization*, IPCO '07, pages 240–251, Berlin, Heidelberg. Springer-Verlag. 97, 99

Pantofaru, C., Schmid, C., and Hebert, M. (2008). Object recognition by integrating multiple image segmentations. In *European Conference on Computer Vision*. 37

Pearl, J. (1998). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann. 40

Potetz, B. and Lee, T. S. (2008). Efficient belief propegation for higher order cliques using linear constraint nodes. 47, 53, 66

Promislow, S. and Young, V. (2005). Supermodular functions on finite lattices. *Order*, 22(4):389–413. 101, 127

Queyranne, M. (2002). An introduction to submodular functions and optimization. Technical report, University of British Columbia. 100

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). Objects in context. In *International Conference on Computer Vision*, Rio de Janeiro. 37, 38, 72, 73, 77, 78, 80, 95, 96

Ramalingam, S., Kohli, P., Alahari, K., and Torr, P. H. S. (2008). Exact inference in multi-label crfs with higher order cliques. In *IEEE Computer Vision and Pattern Recognition*. 104

Ren, X., Fowlkes, C., and Malik, J. (2005). Mid-level cues improve boundary detection. Technical Report UCB/CSD-05-1382, EECS Department, University of California, Berkeley. 72

Reynolds, J. and Murphy, K. (2007). Figure-ground segmentation using a hierarchical conditional random field. In *CRV '07: Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, pages 175–182, Washington, DC, USA. IEEE Computer Society. 38

Roth, S. and Black, M. (2005). Fields of experts: A framework for learning image priors. In *IEEE Computer Vision and Pattern Recognition*, pages 860–867. 47

Rother, C., Kohli, P., Feng, W., and Jia, J. (2009). Minimizing sparse higher order energy functions of discrete variables. In *IEEE Computer Vision and Pattern Recognition09*, pages 1382–1389. 47

Rother, C., Kolmogorov, V., and Blake, A. (2004). GrabCut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, pages 309–314. 44

Rother, C., Kumar, S., Kolmogorov, V., and Blake, A. (2005). Digital tapestry. In *IEEE Computer Vision and Pattern Recognition (1)*, pages 589–596. 90

Russell, B., Freeman, W., Efros, A., Sivic, J., and Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Computer Vision and Pattern Recognition*. 72

Russell, C., Fayad, J., and Agapito, L. (2011). Energy based multiple model fitting for non-rigid structure from motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 135

Russell, C., Restif, C., Metaxas, D., and Torr, P. (2007). Using the pn potts model with learning methods to segment live cell images. In *IEEE Computer*

Society Workshop on Mathematical Methods in Biomedical Image Analysis. 10, 33

Schlesinger, D. (2007). Exact solution of permuted submodular minsum problems. Energy Minimization Methods in Computer Vision and Pattern Recognition '07, pages 28–38, Berlin, Heidelberg. Springer-Verlag. 13

Schlesinger, D. and Flach, B. (2006). Transforming an arbitrary minsum problem into a binary one. Technical Report TUD-FI06-01, Dresden University of Technology. 104

Schlesinger, M. (1976). Syntactic analysis of two-dimensional visual signals in noisy conditions. *Kibernetika*. 81

Schrijver, A. (2000). A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal Comb. Theory Ser. B*, 80:346–355. 99

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 73

Shotton, J., Johnson, M., and Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *IEEE Computer Vision and Pattern Recognition*. 40

Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2006). *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, pages 1–15. 9, 10, 35, 40, 65, 67, 68, 69, 72, 74, 93, 95

Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T., and Weiss, Y. . (2008). Tightening lp relaxations for map using message passing. In *Uncertainty in Artificial Intelligence*. 47, 52

Sturgess, P., Alahari, K., Ladicky, L., and Torr, P. H. S. (2009). Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference*. 135

Sutton, C. and McCallum, A. (2007). Piecewise pseudolikelihood for efficient training of conditional random fields. In *ICML*, pages 863–870. 6

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2006). A comparative study of energy minimization methods for markov random fields. In *European Conference on Computer Vision (2)*, pages 16–29. 47, 52, 75, 85

Tarlow, D., Givoni, I., and Zemel, R. (2010). Hop-map: Efficient message passing with high order potentials. In *Artificial Intelligence and Statistics*. 53

Tarlow, D., Zemel, R., and Frey, B. (2008). Flexible priors for exemplar-based clustering. In *Uncertainty in Artificial Intelligence (UAI)*. 53

Taskar, B. (2004). *Learning Structured Prediction Models: A Large Margin Approach. December 2004.* PhD thesis, Stanford University. 16

Taskar, B., Chatalbashev, V., and Koller, D. (2004). Learning associative markov networks. In *Proc. ICML*, page 102. ACM Press. 15, 18, 26, 47, 49, 136

Taskar, B., Guestrin, C., and Koller, D. (2003). Max-margin Markov networks. In *NIPS*. 8

Torr, P. H. S. (1998). Geometric motion segmentation and model selection [and discussion]. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*. 79

Torr, P. H. S., Fitzgibbon, A. W., and Zisserman, A. (1999). The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32:27–44. 136

Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. (2003). Context-based vision system for place and object recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 273–280 vol.1. 72, 73, 75, 77, 78, 95, 96

Toyoda, T. and Hasegawa, O. (2008). Random field model for integration of local information and global information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1483–1489. 75, 77, 78

Veksler, O. (2007). Graph cut based optimization for mrfs with truncated convex priors. In *IEEE Computer Vision and Pattern Recognition.* 15, 22, 58, 88

Vicente, S., Kolmogorov, V., and Rother, C. (2009). Joint optimization of segmentation and appearance models. In *International Conference on Computer Vision.* IEEE. 47

Villamizar, M., Scandaliaris, J., Sanfeliu, A., and Andrade-Cetto, J. (2009). Combining color-based invariant gradient detector with hog descriptors for robust image detection in scenes under cast shadows. In *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 1573–1578, Piscataway, NJ, USA. IEEE Press. 10

Živný, S., Cohen, D. A., and Jeavons, P. G. (2009). The expressive power of binary submodular functions. *Discrete Applied Mathematics*, 157(15):3347 – 3358. 100, 101, 127

Živný, S. and Jeavons, P. G. (2008). Classes of submodular constraints expressible by graph cuts. In *Proceedings of the 14th international conference on Principles and Practice of Constraint Programming*, CP '08, pages 112–127, Berlin, Heidelberg. Springer-Verlag. 102, 103, 127, 128

Wainwright, M., Jaakkola, T., and Willsky, A. (2002). Map estimation via agreement on (hyper)trees: messagepassing and linear programming approaches. 81

Wainwright, M., Jaakkola, T., and Willsky, A. (2005). Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717. 47, 82

Wainwright, M. and Jordan, M. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305. 50

Weiss, Y. and Freeman, W. (2001). On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *Transactions on Information Theory.* 47, 52, 80

Werner, T. (2005). A linear programming approach to max-sum problem: A review. Research Report CTU–CMP–2005–25, Center for Machine Perception, Czech Technical University. 82

Werner, T. (2009). High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (map-mrf). In *IEEE Computer Vision and Pattern Recognition.* 47, 52

Woodford, O., Torr, P., Reid, I., and Fitzgibbon, A. (2008). Global stereo reconstruction under second order smoothness priors. In *IEEE Computer Vision and Pattern Recognition.* 12, 47

Yang, L., Meer, P., and Foran, D. J. (2007). Multiple class segmentation using a unified framework over mean-shift patches. In *IEEE Computer Vision and Pattern Recognition.* 37, 38, 40, 42, 72

Yuille, A., Rangarajan, A., and Yuille, A. L. (2002). The concave-convex procedure (cccp. In *Advances in Neural Information Processing Systems 14*. MIT Press. 91, 114

Zalesky, B. (2003). Efficient Determination of Gibbs Estimators with Submodular Energy Functions. *ArXiv Mathematics e-prints*. 100

Zhu, L. and Yuille, A. L. (2005). A hierarchical compositional system for rapid object detection. In *NIPS*. 38

Zivny, S. and Jeavons, P. (2008). Which submodular functions are expressible using binary submodular functions? *Oxford University Computing Laboratory Researc Report CS-RR-08-08*. 108, 128