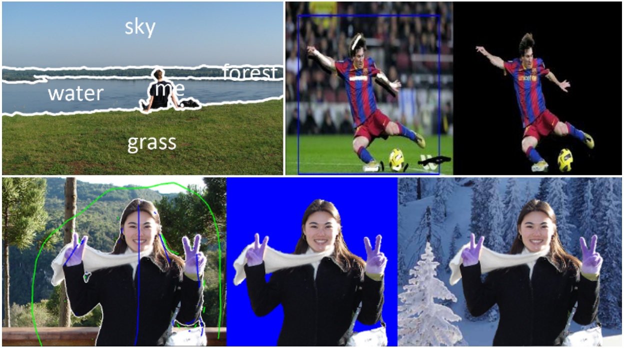
Low-Rank Approximation of MRF Energy by means of the TT-Format

A. Rodomanov A. Novikov A. Osokin D. Vetrov

Lomonosov Moscow State University

SIAM-IS14, Hong Kong, May 2014

Semantic Image Segmentation



Goal: assign a label $t_i \in \Lambda$ to each pixel of the image.

Problem: for an $M \times N$ image there are $|\Lambda|^{MN}$ possible labellings. Which one is the best?

Probabilistic Approach

- Define a probabilistic model on the set of all possible labellings.

Probabilistic Approach

- Define a probabilistic model on the set of all possible labellings.
- Let $p(T|X, W)$ measure the probability of the labelling T given the image X and the parameters of the model W.

Probabilistic Approach

- Define a probabilistic model on the set of all possible labellings.
- Let $p(T|X, W)$ measure the probability of the labelling T given the image X and the parameters of the model W.
- The goal is to find the labelling T^* that maximizes $p(T|X, W)$:
$$T^* = \arg\max_T p(T|X, W).$$
This is called the maximum a posteriori (MAP) inference.

Probabilistic Approach

- Define a probabilistic model on the set of all possible labellings.
- Let $p(T|X, W)$ measure the probability of the labelling T given the image X and the parameters of the model W.
- The goal is to find the labelling T^* that maximizes $p(T|X, W)$:
$$T^* = \arg\max_T p(T|X, W).$$
This is called the maximum a posteriori (MAP) inference.
- We will use Markov random fields (MRFs) to define the probabilistic model $p(T|X, W)$.

Markov Random Fields

- Define an undirected graph $\mathcal{G}$ with nodes corresponding to the pixels of the image.

Markov Random Fields

- Define an undirected graph $\mathcal{G}$ with nodes corresponding to the pixels of the image.
- Define some positive functions $\Psi_c(T_c; X, W)$ (called MRF factors) on the cliques of the graph $\mathcal{G}$.

Markov Random Fields

- Define an undirected graph $\mathcal{G}$ with nodes corresponding to the pixels of the image.
- Define some positive functions $\Psi_c(T_c; X, W)$ (called MRF factors) on the cliques of the graph $\mathcal{G}$.
- The model is then defined as follows:
$$p(T|X, W) = \frac{1}{Z(X, W)} \prod_{c \in \mathcal{C}} \Psi_c(T_c; X, W),$$
where $Z(X, W)$ is the normalization constant.

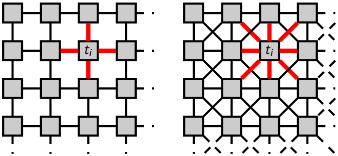
Markov Random Fields cont'd

- How to choose the graph $\mathcal{G}$?

- How to choose the factors $\Psi_c(T_c; X, W)$?

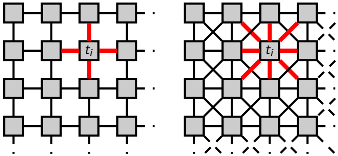
Markov Random Fields cont'd

- How to choose the graph $\mathcal{G}$?
  The structure of the graph defines relations between the pixels. E.g., adjacent pixels are likely to have the same label, so they are linked by an edge. The graph $\mathcal{G}$ is usually grid-like. Two popular variants:



- How to choose the factors $\Psi_c(T_c; X, W)$?

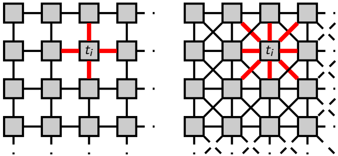
- How to choose the graph $\mathcal{G}$?
  The structure of the graph defines relations between the pixels. E.g., adjacent pixels are likely to have the same label, so they are linked by an edge. The graph $\mathcal{G}$ is usually grid-like. Two popular variants:



- How to choose the factors $\Psi_c(T_c; X, W)$?
  Determine, how likely the labelling T_c for the clique c is. E.g., there are two types of factors for the first graph:

Markov Random Fields cont'd

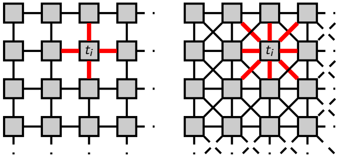
- How to choose the graph $\mathcal{G}$?
  The structure of the graph defines relations between the pixels. E.g., adjacent pixels are likely to have the same label, so they are linked by an edge. The graph $\mathcal{G}$ is usually grid-like. Two popular variants:



- How to choose the factors $\Psi_c(T_c; X, W)$?
  Determine, how likely the labelling T_c for the clique c is. E.g., there are two types of factors for the first graph:
  - unary factors $\Psi_i(t_i)$: how likely it is that the i-th pixel is labelled as t_i;

Markov Random Fields cont'd

- How to choose the graph $\mathcal{G}$?
  The structure of the graph defines relations between the pixels. E.g., adjacent pixels are likely to have the same label, so they are linked by an edge. The graph $\mathcal{G}$ is usually grid-like. Two popular variants:



- How to choose the factors $\Psi_c(T_c; X, W)$?
  Determine, how likely the labelling T_c for the clique c is. E.g., there are two types of factors for the first graph:
  - unary factors $\Psi_i(t_i)$: how likely it is that the i-th pixel is labelled as t_i;
  - pairwise factors $\Psi_{ij}(t_i, t_j)$: how likely it is that the i-th and j-th pixels are *simultaneously* labelled as t_i and t_j.

MAP-Inference

The MAP-inference problem now corresponds to the following problem:

$$\max_T p(T|X, W) = \max_T \frac{1}{Z(X, W)} \prod_{c \in \mathcal{C}} \Psi_c(T_c; X, W).$$

Further we demonstrate how one can address such a problem using the Tensor-Train (TT) framework.

MAP-Inference

The MAP-inference problem now corresponds to the following problem:

$$\max_T p(T|X,W) = \max_T \frac{1}{Z(X,W)} \prod_{c\in\mathcal{C}} \Psi_c(T_c; X, W).$$

Further we demonstrate how one can address such a problem using the Tensor-Train (TT) framework.

We will assume that

- the parameters of the model W are already chosen (how to choose them will be touched on in the next talk);

MAP-Inference

The MAP-inference problem now corresponds to the following problem:

$$\max_T p(T|X, W) = \max_T \frac{1}{Z(X, W)} \prod_{c \in \mathcal{C}} \Psi_c(T_c; X, W).$$

Further we demonstrate how one can address such a problem using the Tensor-Train (TT) framework.

We will assume that

- the parameters of the model W are already chosen (how to choose them will be touched on in the next talk);
- we are performing the MAP-inference for the concrete image X.

MAP-Inference

The MAP-inference problem now corresponds to the following problem:

$$\max_T p(T|X, W) = \max_T \frac{1}{Z(X, W)} \prod_{c \in \mathcal{C}} \Psi_c(T_c; X, W).$$

Further we demonstrate how one can address such a problem using the Tensor-Train (TT) framework.

We will assume that

- the parameters of the model W are already chosen (how to choose them will be touched on in the next talk);
- we are performing the MAP-inference for the concrete image X.

So, to simplify notation, we won't explicitly write X, W any more:

$$\max_T p(T) = \max_T \frac{1}{Z} \prod_{c \in \mathcal{C}} \Psi_c(T_c)$$

Tensor-Train Framework

- An n-dimensional tensor $\boldsymbol{A}$ is said to be represented in the TT-format if its elements can be expressed as the following matrix product:

$$\boldsymbol{A}(x_1, \ldots, x_n) = \underbrace{G_1[x_1]}_{\mathrm{r}_0 \times \mathrm{r}_1} \underbrace{G_2[x_2]}_{\mathrm{r}_1 \times \mathrm{r}_2} \ldots \underbrace{G_n[x_n]}_{\mathrm{r}_{n-1} \times \mathrm{r}_n} .$$

Tensor-Train Framework

- An n-dimensional tensor $\boldsymbol{A}$ is said to be represented in the TT-format if its elements can be expressed as the following matrix product:

$$\boldsymbol{A}(x_1, \ldots, x_n) = \underbrace{G_1[x_1]}_{\mathrm{r}_0 \times \mathrm{r}_1} \underbrace{G_2[x_2]}_{\mathrm{r}_1 \times \mathrm{r}_2} \ldots \underbrace{G_n[x_n]}_{\mathrm{r}_{n-1} \times \mathrm{r}_n} .$$

- The matrices $G_i[x_i]$ are called the TT-cores and their sizes (numbers r_i) are referred to as the TT-ranks.

Tensor-Train Framework

- An n-dimensional tensor $\boldsymbol{A}$ is said to be represented in the TT-format if its elements can be expressed as the following matrix product:

$$\boldsymbol{A}(x_1, \ldots, x_n) = \underbrace{G_1[x_1]}_{\mathrm{r}_0 \times \mathrm{r}_1} \underbrace{G_2[x_2]}_{\mathrm{r}_1 \times \mathrm{r}_2} \ldots \underbrace{G_n[x_n]}_{\mathrm{r}_{n-1} \times \mathrm{r}_n} .$$

- The matrices $G_i[x_i]$ are called the TT-cores and their sizes (numbers r_i) are referred to as the TT-ranks.
- The TT-format is very efficient provided that the TT-ranks are small.

Tensor-Train Framework

- An n-dimensional tensor $\boldsymbol{A}$ is said to be represented in the TT-format if its elements can be expressed as the following matrix product:

$$\boldsymbol{A}(x_1, \dots, x_n) = \underbrace{G_1[x_1]}_{r_0 \times r_1} \underbrace{G_2[x_2]}_{r_1 \times r_2} \dots \underbrace{G_n[x_n]}_{r_{n-1} \times r_n} .$$

- The matrices $G_i[x_i]$ are called the TT-cores and their sizes (numbers r_i) are referred to as the TT-ranks.
- The TT-format is very efficient provided that the TT-ranks are small.
- Two algorithms for converting a tensor into the TT-format:

Tensor-Train Framework

- An n-dimensional tensor $\boldsymbol{A}$ is said to be represented in the TT-format if its elements can be expressed as the following matrix product:
$$\boldsymbol{A}(x_1, \dots, x_n) = \underbrace{G_1[x_1]}_{r_0 \times r_1} \underbrace{G_2[x_2]}_{r_1 \times r_2} \dots \underbrace{G_n[x_n]}_{r_{n-1} \times r_n} .$$
- The matrices $G_i[x_i]$ are called the TT-cores and their sizes (numbers r_i) are referred to as the TT-ranks.
- The TT-format is very efficient provided that the TT-ranks are small.
- Two algorithms for converting a tensor into the TT-format:
  - TT-SVD: finds an exact TT-representation for a tensor but suitable only for low dimensionality n.

Tensor-Train Framework

- An n-dimensional tensor $\boldsymbol{A}$ is said to be represented in the TT-format if its elements can be expressed as the following matrix product:

$$\boldsymbol{A}(x_1, \dots, x_n) = \underbrace{G_1[x_1]}_{r_0 \times r_1} \underbrace{G_2[x_2]}_{r_1 \times r_2} \dots \underbrace{G_n[x_n]}_{r_{n-1} \times r_n}.$$

- The matrices $G_i[x_i]$ are called the TT-cores and their sizes (numbers r_i) are referred to as the TT-ranks.
- The TT-format is very efficient provided that the TT-ranks are small.
- Two algorithms for converting a tensor into the TT-format:
  - TT-SVD: finds an exact TT-representation for a tensor but suitable only for low dimensionality n.
  - AMEn: builds a TT-approximation of a tensor by using only a small fraction of its elements; suitable for high dimensionality n but doesn't have strong theoretical guarantees.

Problem Formulation & Notation

Suppose that:

- the MRF contains n variables denoted by $x_1, \ldots, x_n$;

Problem Formulation & Notation

Suppose that:

- the MRF contains n variables denoted by $x_1, \ldots, x_n$;
- each variable x_i takes values from the domain $\{1, \ldots, d\}$;

Problem Formulation & Notation

Suppose that:

- the MRF contains n variables denoted by $x_1, \ldots, x_n$;
- each variable x_i takes values from the domain $\{1, \ldots, d\}$;
- all the potentials are numbered from 1 to m.

Problem Formulation & Notation

Suppose that:

- the MRF contains n variables denoted by $x_1, \dots, x_n$;
- each variable x_i takes values from the domain $\{1, \dots, d\}$;
- all the potentials are numbered from 1 to m.

Denote:

- $\boldsymbol{x} = (x_1, \dots, x_n)$ — vector of all variables;

Problem Formulation & Notation

Suppose that:

- the MRF contains n variables denoted by $x_1, \dots, x_n$;
- each variable x_i takes values from the domain $\{1, \dots, d\}$;
- all the potentials are numbered from 1 to m.

Denote:

- $\boldsymbol{x} = (x_1, \dots, x_n)$ — vector of all variables;
- $\boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$ — ℓ-th factor;

Problem Formulation & Notation

Suppose that:

- the MRF contains n variables denoted by $x_1, \ldots, x_n$;
- each variable x_i takes values from the domain $\{1, \ldots, d\}$;
- all the potentials are numbered from 1 to m.

Denote:

- $\boldsymbol{x} = (x_1, \ldots, x_n)$ — vector of all variables;
- $\boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$ — ℓ-th factor;
- $\boldsymbol{x}^\ell$ — vector of variables on which the ℓ-th factor depends.

Problem Formulation & Notation

Suppose that:

- the MRF contains n variables denoted by $x_1, \ldots, x_n$;
- each variable x_i takes values from the domain $\{1, \ldots, d\}$;
- all the potentials are numbered from 1 to m.

Denote:

- $\boldsymbol{x} = (x_1, \ldots, x_n)$ — vector of all variables;
- $\boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$ — ℓ-th factor;
- $\boldsymbol{x}^\ell$ — vector of variables on which the ℓ-th factor depends.

The main problem of our interest is the MAP-inference problem:

$$\max_{\boldsymbol{x}} \boldsymbol{P}(\boldsymbol{x}) = \max_{\boldsymbol{x}} \frac{1}{Z} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell).$$

MAP-Inference & Energy Minimization

The MAP-inference problem

$$\max_{\boldsymbol{x}} \boldsymbol{P}(\boldsymbol{x}) = \max_{\boldsymbol{x}} \frac{1}{Z} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_{\ell}(\boldsymbol{x}^{\ell})$$

is equivalent to the following problem:

$$\min_{\boldsymbol{x}} \sum_{\ell=1}^{m} [-\ln \boldsymbol{\Psi}_{\ell}(\boldsymbol{x}^{\ell})].$$

MAP-Inference & Energy Minimization

The MAP-inference problem

$$\max_{\boldsymbol{x}} \boldsymbol{P}(\boldsymbol{x}) = \max_{\boldsymbol{x}} \frac{1}{Z} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$$

is equivalent to the following problem:

$$\min_{\boldsymbol{x}} \sum_{\ell=1}^{m} [-\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)].$$

Terminology:

- The terms $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell) = -\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$ are called MRF potentials.

MAP-Inference & Energy Minimization

The MAP-inference problem

$$\max_{\boldsymbol{x}} \boldsymbol{P}(\boldsymbol{x}) = \max_{\boldsymbol{x}} \frac{1}{Z} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$$

is equivalent to the following problem:

$$\min_{\boldsymbol{x}} \sum_{\ell=1}^{m} [-\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)].$$

Terminology:

- The terms $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell) = -\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$ are called MRF potentials.
- Their sum $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ is called MRF energy.

The MAP-inference problem

$$\max_{\boldsymbol{x}} \boldsymbol{P}(\boldsymbol{x}) = \max_{\boldsymbol{x}} \frac{1}{Z} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$$

is equivalent to the following problem:

$$\min_{\boldsymbol{x}} \sum_{\ell=1}^{m} [-\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)].$$

Terminology:

- The terms $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell) = -\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$ are called MRF potentials.
- Their sum $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ is called MRF energy.

So, the MAP-inference is equivalent to energy minimization:

$$\min_{\boldsymbol{x}} \boldsymbol{E}(\boldsymbol{x}) = \min_{\boldsymbol{x}} \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell).$$

Tensor Approach

- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be considered as an n-dimensional tensor:
$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{E}(x_1, \dots, x_n).$$

Tensor Approach

- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be considered as an n-dimensional tensor:

$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{E}(x_1, \ldots, x_n).$$

- Then energy minimization corresponds to finding the minimal element in the tensor $\boldsymbol{E}(\boldsymbol{x})$.

Tensor Approach

- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be considered as an n-dimensional tensor:

$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{E}(x_1, \ldots, x_n).$$

- Then energy minimization corresponds to finding the minimal element in the tensor $\boldsymbol{E}(\boldsymbol{x})$.
- If the energy $\boldsymbol{E}(\boldsymbol{x})$ were represented in the TT-format, we could use a special algorithm from the TT-framework to find the minimal element.

Tensor Approach

- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be considered as an n-dimensional tensor:
$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{E}(x_1, \ldots, x_n).$$
- Then energy minimization corresponds to finding the minimal element in the tensor $\boldsymbol{E}(\boldsymbol{x})$.
- If the energy $\boldsymbol{E}(\boldsymbol{x})$ were represented in the TT-format, we could use a special algorithm from the TT-framework to find the minimal element.
- How to convert the energy tensor into the TT-format?
  AMEn-algorithm?

Tensor Approach

- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be considered as an n-dimensional tensor:
$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{E}(x_1, \dots, x_n).$$
- Then energy minimization corresponds to finding the minimal element in the tensor $\boldsymbol{E}(\boldsymbol{x})$.
- If the energy $\boldsymbol{E}(\boldsymbol{x})$ were represented in the TT-format, we could use a special algorithm from the TT-framework to find the minimal element.
- How to convert the energy tensor into the TT-format?
AMEn-algorithm? Possible, but there is also a much better way!

The Idea of the Algorithm

- Let's try to take into account the structure of the energy tensor $\boldsymbol{E}$.
  Recall: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.

The Idea of the Algorithm

- Let's try to take into account the structure of the energy tensor $\boldsymbol{E}$. Recall: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- Each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ can be considered as an n-dimensional tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$ if we add inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ for non-existing dimensions: $\boldsymbol{\Theta}_\ell(\boldsymbol{x}) \equiv \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.

The Idea of the Algorithm

- Let's try to take into account the structure of the energy tensor $\boldsymbol{E}$. Recall: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- Each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ can be considered as an n-dimensional tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$ if we add inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ for non-existing dimensions: $\boldsymbol{\Theta}_\ell(\boldsymbol{x}) \equiv \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be expressed as a sum of the tensors $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$:

$$\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}).$$

The Idea of the Algorithm

- Let's try to take into account the structure of the energy tensor $\boldsymbol{E}$.
  Recall: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- Each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ can be considered as an n-dimensional tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$ if we add inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ for non-existing dimensions: $\boldsymbol{\Theta}_\ell(\boldsymbol{x}) \equiv \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be expressed as a sum of the tensors $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$:

$$\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}).$$

- If the tensors $\boldsymbol{\Theta}_\ell$ were represented in the TT-format, we could exploit the summation operation on tensors in the TT-format to build the TT-representation for the tensor $\boldsymbol{E}$.

The Idea of the Algorithm

- Let's try to take into account the structure of the energy tensor $\boldsymbol{E}$. Recall: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- Each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ can be considered as an n-dimensional tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$ if we add inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ for non-existing dimensions: $\boldsymbol{\Theta}_\ell(\boldsymbol{x}) \equiv \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be expressed as a sum of the tensors $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$:

$$\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}).$$

- If the tensors $\boldsymbol{\Theta}_\ell$ were represented in the TT-format, we could exploit the summation operation on tensors in the TT-format to build the TT-representation for the tensor $\boldsymbol{E}$.
- How to find the TT-decomposition for each tensor $\boldsymbol{\Theta}_\ell$?

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it n-dimensional.

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it *n*-dimensional.
- These inessential variables can be added constructively:

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it n-dimensional.
- These inessential variables can be added constructively:
  - Let $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$, $\boldsymbol{x}^\ell = (x_1, x_2, x_4)$.

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it n-dimensional.
- These inessential variables can be added constructively:
  - Let $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$, $\boldsymbol{x}^\ell = (x_1, x_2, x_4)$.
  - Suppose that after TT-SVD we have:

$$\boldsymbol{\Theta}_\ell(x_1, x_2, x_4) = G_1[x_1]G_2[x_2]G_4[x_4].$$

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it n-dimensional.
- These inessential variables can be added constructively:
    - Let $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$, $\boldsymbol{x}^\ell = (x_1, x_2, x_4)$.
    - Suppose that after TT-SVD we have:
    $$\boldsymbol{\Theta}_\ell(x_1, x_2, x_4) = G_1[x_1]G_2[x_2]G_4[x_4].$$
    - To introduce x_3, x_5, we need to define the missing cores $G_3[x_3]$, $G_5[x_5]$.

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it n-dimensional.
- These inessential variables can be added constructively:
    - Let $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$, $\boldsymbol{x}^\ell = (x_1, x_2, x_4)$.
    - Suppose that after TT-SVD we have:

$$\boldsymbol{\Theta}_\ell(x_1, x_2, x_4) = G_1[x_1] G_2[x_2] G_4[x_4].$$

    - To introduce x_3, x_5, we need to define the missing cores $G_3[x_3]$, $G_5[x_5]$.
    - Define them as identity matrices:

$$\boldsymbol{\Theta}_\ell(x_1, x_2, x_3, x_4, x_5) = G_1[x_1] G_2[x_2] \underbrace{I}_{\equiv G_3[x_3]} G_4[x_4] \underbrace{I}_{\equiv G_5[x_5]}.$$

Converting Potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it n-dimensional.
- These inessential variables can be added constructively:
  - Let $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$, $\boldsymbol{x}^\ell = (x_1, x_2, x_4)$.
  - Suppose that after TT-SVD we have:
    $$\boldsymbol{\Theta}_\ell(x_1, x_2, x_4) = G_1[x_1]G_2[x_2]G_4[x_4].$$
  - To introduce x_3, x_5, we need to define the missing cores $G_3[x_3]$, $G_5[x_5]$.
  - Define them as identity matrices:
    $$\boldsymbol{\Theta}_\ell(x_1, x_2, x_3, x_4, x_5) = G_1[x_1]G_2[x_2] \underbrace{I}_{\equiv G_3[x_3]} G_4[x_4] \underbrace{I}_{\equiv G_5[x_5]}.$$
  - The maximal TT-rank hasn't increased!

The Algorithm & Its Theoretical Guarantees

1. Compute the TT-decomposition for each individual potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.

The Algorithm & Its Theoretical Guarantees

1. Compute the TT-decomposition for each individual potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
2. Add the inessential variables to each $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ to obtain $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$.

The Algorithm & Its Theoretical Guarantees

1. Compute the TT-decomposition for each individual potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
2. Add the inessential variables to each $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ to obtain $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$.
3. Use the TT-summation to build $\boldsymbol{E}(\boldsymbol{x})$: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x})$.

The Algorithm & Its Theoretical Guarantees

1. Compute the TT-decomposition for each individual potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
2. Add the inessential variables to each $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ to obtain $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$.
3. Use the TT-summation to build $\boldsymbol{E}(\boldsymbol{x})$: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^m \boldsymbol{\Theta}_\ell(\boldsymbol{x})$.

Theorem

The maximal TT-rank of the tensor $\boldsymbol{E}$ constructed by the algorithm is polynomially bounded:

$$\mathrm{r}(\boldsymbol{E}) \leq d^{\frac{p}{2}} m,$$

where

- *d is the number of values that each variable can take;*
- *m is the total number of potentials;*
- *p is the maximal order of a potential (i.e. the maximal $|\boldsymbol{x}^\ell|$).*

The Algorithm & Its Theoretical Guarantees

1. Compute the TT-decomposition for each individual potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
2. Add the inessential variables to each $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ to obtain $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$.
3. Use the TT-summation to build $\boldsymbol{E}(\boldsymbol{x})$: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^m \boldsymbol{\Theta}_\ell(\boldsymbol{x})$.

Theorem

The maximal TT-rank of the tensor $\boldsymbol{E}$ constructed by the algorithm is polynomially bounded:

$$\mathrm{r}(\boldsymbol{E}) \le d^{\frac{p}{2}} m,$$

where

- *d is the number of values that each variable can take;*
- *m is the total number of potentials;*
- *p is the maximal order of a potential (i.e. the maximal $|\boldsymbol{x}^\ell|$).*

Consider $d = 2$, $p = 2$. Then $\mathrm{r}(\boldsymbol{E}) \le 2m$ (linear dependence on m).

High Order Potentials

- Sometimes it is convenient to use potentials of high order, i.e. those which depend on many variables.

High Order Potentials

- Sometimes it is convenient to use potentials of high order, i.e. those which depend on many variables. E.g., the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \underbrace{\left[\sum_{i=1}^{n} x_i \leq a\right]}_{\text{indicator function}},$$

which depends on all the variables, could be used to specify some preference on the minimal value of the area of foreground in the problem of segmenting an image into background/foreground.

High Order Potentials

- Sometimes it is convenient to use potentials of high order, i.e. those which depend on many variables. E.g., the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \underbrace{\left[\sum_{i=1}^{n} x_i \leq a\right]}_{\text{indicator function}},$$

  which depends on all the variables, could be used to specify some preference on the minimal value of the area of foreground in the problem of segmenting an image into background/foreground.
- We can't use the TT-SVD algorithm any more to convert such potentials into the TT-format!

High Order Potentials

- Sometimes it is convenient to use potentials of high order, i.e. those which depend on many variables. E.g., the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \underbrace{\left[\sum_{i=1}^{n} x_i \leq a\right]}_{\text{indicator function}},$$

which depends on all the variables, could be used to specify some preference on the minimal value of the area of foreground in the problem of segmenting an image into background/foreground.

- We can't use the TT-SVD algorithm any more to convert such potentials into the TT-format!
- However, for some of these potentials we can explicitly construct the TT-representation, i.e. we can derive analytical formulas for the corresponding TT-cores.

High Order Potentials

- Sometimes it is convenient to use potentials of high order, i.e. those which depend on many variables. E.g., the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \underbrace{\left[\sum_{i=1}^{n} x_i \le a\right]}_{\text{indicator function}},$$

which depends on all the variables, could be used to specify some preference on the minimal value of the area of foreground in the problem of segmenting an image into background/foreground.
- We can't use the TT-SVD algorithm any more to convert such potentials into the TT-format!
- However, for some of these potentials we can explicitly construct the TT-representation, i.e. we can derive analytical formulas for the corresponding TT-cores.
- Such TT-representations will be of low TT-rank!

Sparse Potential

- Consider a so-called sparse potential:

$$\boldsymbol{\Theta}_\ell(x_{i_1}, \dots, x_{i_w}) = [x_{i_1} = \beta_1] \dots [x_{i_w} = \beta_w].$$

  It always equals zero with the exception of only one configuration.

Sparse Potential

- Consider a so-called sparse potential:
$$\boldsymbol{\Theta}_\ell(x_{i_1}, \dots, x_{i_w}) = [x_{i_1} = \beta_1] \dots [x_{i_w} = \beta_w].$$
It always equals zero with the exception of only one configuration.
- Such a potential admits a TT-representation
$$\boldsymbol{\Theta}_\ell(x_{i_1}, \dots, x_{i_w}) = G_{i_1}[x_{i_1}] \dots G_{i_w}[x_{i_w}]$$
with the following TT-cores:
$$G_{i_v}[x_{i_v}] = [x_{i_v} = \beta_v], \quad v = 1, \dots, w.$$

Sparse Potential

- Consider a so-called sparse potential:
$$\boldsymbol{\Theta}_\ell(x_{i_1},\dots,x_{i_w}) = [x_{i_1} = \beta_1]\dots[x_{i_w} = \beta_w].$$
It always equals zero with the exception of only one configuration.
- Such a potential admits a TT-representation
$$\boldsymbol{\Theta}_\ell(x_{i_1},\dots,x_{i_w}) = G_{i_1}[x_{i_1}]\dots G_{i_w}[x_{i_w}]$$
with the following TT-cores:
$$G_{i_v}[x_{i_v}] = [x_{i_v} = \beta_v], \quad v = 1,\dots,w.$$
- In this case each TT-core is simply a number (1-by-1 matrix) for every concrete value of x_{i_v}. Hence, the maximal TT-rank equals 1.

Sparse Potential

- Consider a so-called sparse potential:
$$\Theta_\ell(x_{i_1}, \ldots, x_{i_w}) = [x_{i_1} = \beta_1] \ldots [x_{i_w} = \beta_w].$$
It always equals zero with the exception of only one configuration.
- Such a potential admits a TT-representation
$$\Theta_\ell(x_{i_1}, \ldots, x_{i_w}) = G_{i_1}[x_{i_1}] \ldots G_{i_w}[x_{i_w}]$$
with the following TT-cores:
$$G_{i_v}[x_{i_v}] = [x_{i_v} = \beta_v], \quad v = 1, \ldots, w.$$
- In this case each TT-core is simply a number (1-by-1 matrix) for every concrete value of x_{i_v}. Hence, the maximal TT-rank equals 1.
- A more general sparse potential which differs from zero on $s > 1$ configurations can be obtained as a sum of several potentials of the above type. Thus, the TT-rank of a general sparse potential is bounded above by s.

Area Potential

- Consider the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \left[\sum_{i=1}^{n} x_i \leq a\right],$$

where $x_i \in \{0, 1\}$ and $a \in \mathbb{Z}_+$.

Area Potential

- Consider the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \left[\sum_{i=1}^{n} x_i \leq a\right],$$

where $x_i \in \{0, 1\}$ and $a \in \mathbb{Z}_+$.

- This potential can be analytically represented in the TT-format with the maximal TT-rank equal to $a + 1$:

Area Potential

- Consider the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \left[\sum_{i=1}^{n} x_i \leq a\right],$$

where $x_i \in \{0, 1\}$ and $a \in \mathbb{Z}_+$.

- This potential can be analytically represented in the TT-format with the maximal TT-rank equal to $a + 1$:

$$G_i[x_i] = (S_a)^{x_i}, \quad (i = 2, \ldots, n-1),$$

$$G_1[x_1] = [\overbrace{\underbrace{0 \ldots 0}_{x_1}\ 1 \ldots 1}^{a+1}], \quad G_n[x_n] = (S_a)^{x_n}[\underbrace{0 \ldots 0}_{a}\ 1]^T,$$

Area Potential

- Consider the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \left[\sum_{i=1}^{n} x_i \le a \right],$$

where $x_i \in \{0, 1\}$ and $a \in \mathbb{Z}_+$.

- This potential can be analytically represented in the TT-format with the maximal TT-rank equal to $a + 1$:

$$G_i[x_i] = (S_a)^{x_i}, \quad (i = 2, \ldots, n-1),$$

$$G_1[x_1] = [\overbrace{\underbrace{0 \ldots 0}_{x_1} \; 1 \ldots 1}^{a+1}], \quad G_n[x_n] = (S_a)^{x_n} [\underbrace{0 \ldots 0}_{a} \; 1]^T,$$

where $S_a = \underbrace{\left[\begin{array}{c|c} O & I_a \\ \hline O & O \end{array} \right]}_{(a+1)\times(a+1)}.$

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\dots0}_{k}\;1\dots1}^{a+1}]S_a = [\overbrace{\underbrace{0\dots0}_{k+1}\;1\dots1}^{a+1}]$.

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\ldots0}_{k}\,1\ldots1}^{a+1}]S_a = [\overbrace{\underbrace{0\ldots0}_{k+1}\,1\ldots1}^{a+1}]$.

Consider, e.g., that $a = 3$. In this case

$$S_a = \left[\begin{array}{c|ccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\ldots0}_{k}\,1\ldots1}^{a+1}]S_a = [\overbrace{\underbrace{0\ldots0}_{k+1}\,1\ldots1}^{a+1}]$.

Consider, e.g., that $a = 3$. In this case

$$S_a = \left[\begin{array}{c|ccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

- $[1\,1\,1\,1]S_a = [0\,1\,1\,1]$ (the sum of all rows);

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\ldots0}_{k}\;1\ldots1}^{a+1}]S_a = [\overbrace{\underbrace{0\ldots0}_{k+1}\;1\ldots1}^{a+1}]$.

Consider, e.g., that $a = 3$. In this case

$$S_a = \left[\begin{array}{c|ccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

- $[1\,1\,1\,1]S_a = [0\,1\,1\,1]$ (the sum of all rows);
- $[0\,1\,1\,1]S_a = [0\,0\,1\,1]$ (the sum of rows 2, 3, 4);

Area Potential cont'd

Key property of S_a: $[\underbrace{\overbrace{0 \ldots 0}^{}}_{k}\!\!\overbrace{\;1 \ldots 1}^{a+1}]S_a = [\underbrace{0 \ldots 0}_{k+1}\overbrace{\;1 \ldots 1}^{a+1}]$.

Consider, e.g., that $a = 3$. In this case

$$S_a = \left[\begin{array}{c|ccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

- $[1\,1\,1\,1]S_a = [0\,1\,1\,1]$ (the sum of all rows);
- $[0\,1\,1\,1]S_a = [0\,0\,1\,1]$ (the sum of rows 2, 3, 4);
- and so on.

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\ldots0}_{k}\,1\ldots1}^{a+1}]S_a = [\overbrace{\underbrace{0\ldots0}_{k+1}\,1\ldots1}^{a+1}]$.

Then

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = G_1[x_1]G_2[x_2]G_3[x_3]\ldots G_n[x_n]$$

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\ldots0}_{k}\;1\ldots1}^{a+1}]S_a = [\overbrace{\underbrace{0\ldots0}_{k+1}\;1\ldots1}^{a+1}]$.

Then

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = G_1[x_1]G_2[x_2]G_3[x_3]\ldots G_n[x_n]$$

$$= [\overbrace{\underbrace{0\ldots0}_{x_1}\;1\ldots1}^{a+1}](S_a)^{x_2}(S_a)^{x_3}\ldots(S_a)^{x_n}[\underbrace{0\ldots0}_{a}\;1]^T$$

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0 \ldots 0}_{k} \; 1 \ldots 1}^{a+1}] S_a = [\overbrace{\underbrace{0 \ldots 0}_{k+1} \; 1 \ldots 1}^{a+1}]$.

Then

$$\begin{aligned} \Theta_\ell(\boldsymbol{x}) &= G_1[x_1] G_2[x_2] G_3[x_3] \ldots G_n[x_n] \\ &= [\overbrace{\underbrace{0 \ldots 0}_{x_1} \; 1 \ldots 1}^{a+1}] (S_a)^{x_2} (S_a)^{x_3} \ldots (S_a)^{x_n} [\underbrace{0 \ldots 0}_{a} \; 1]^T \\ &= [\overbrace{\underbrace{0 \ldots 0}_{x_1+x_2} \; 1 \ldots 1}^{a+1}] (S_a)^{x_3} \ldots (S_a)^{x_n} [\underbrace{0 \ldots 0}_{a} \; 1]^T \end{aligned}$$

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\dots0}_{k}\ 1\dots1}^{a+1}]S_a = [\overbrace{\underbrace{0\dots0}_{k+1}\ 1\dots1}^{a+1}]$.

Then

$$\begin{aligned}
\boldsymbol{\Theta}_\ell(\boldsymbol{x}) &= G_1[x_1]G_2[x_2]G_3[x_3]\dots G_n[x_n] \\
&= [\overbrace{\underbrace{0\dots0}_{x_1}\ 1\dots1}^{a+1}](S_a)^{x_2}(S_a)^{x_3}\dots(S_a)^{x_n}[\underbrace{0\dots0}_{a}\ 1]^T \\
&= [\overbrace{\underbrace{0\dots0}_{x_1+x_2}\ 1\dots1}^{a+1}](S_a)^{x_3}\dots(S_a)^{x_n}[\underbrace{0\dots0}_{a}\ 1]^T \\
&= [\overbrace{\underbrace{0\dots0}_{x_1+\dots+x_n}\ \ 1\dots1}^{a+1}][\underbrace{0\dots0}_{a}\ 1]^T
\end{aligned}$$

Area Potential cont'd

Key property of S_a: $[\underbrace{0\dots0}_{k}\overbrace{1\dots1}^{a+1}]S_a = [\underbrace{0\dots0}_{k+1}\overbrace{1\dots1}^{a+1}]$.

Then

$$\begin{aligned}
\boldsymbol{\Theta}_\ell(\boldsymbol{x}) &= G_1[x_1]G_2[x_2]G_3[x_3]\dots G_n[x_n] \\
&= [\overbrace{\underbrace{0\dots0}_{x_1}\ 1\dots1}^{a+1}](S_a)^{x_2}(S_a)^{x_3}\dots(S_a)^{x_n}[\underbrace{0\dots0}_{a}\ 1]^T \\
&= [\overbrace{\underbrace{0\dots0}_{x_1+x_2}\ 1\dots1}^{a+1}](S_a)^{x_3}\dots(S_a)^{x_n}[\underbrace{0\dots0}_{a}\ 1]^T \\
&= [\overbrace{\underbrace{0\dots0}_{x_1+\dots+x_n}\ 1\dots1}^{a+1}][\underbrace{0\dots0}_{a}\ 1]^T = \begin{cases} 1, & \sum_{i=1}^{n} x_i \le a, \\ & \end{cases}
\end{aligned}$$

Area Potential cont'd

Key property of S_a: $[\overbrace{\underbrace{0\dots0}_{k}\ 1\dots1}^{a+1}]S_a = [\overbrace{\underbrace{0\dots0}_{k+1}\ 1\dots1}^{a+1}]$.

Then

$$
\begin{aligned}
\boldsymbol{\Theta}_\ell(\boldsymbol{x}) &= G_1[x_1]G_2[x_2]G_3[x_3]\dots G_n[x_n] \\
&= [\overbrace{\underbrace{0\dots0}_{x_1}\ 1\dots1}^{a+1}](S_a)^{x_2}(S_a)^{x_3}\dots(S_a)^{x_n}[\underbrace{0\dots0}_{a}\ 1]^T \\
&= [\overbrace{\underbrace{0\dots0}_{x_1+x_2}\ 1\dots1}^{a+1}](S_a)^{x_3}\dots(S_a)^{x_n}[\underbrace{0\dots0}_{a}\ 1]^T \\
&= [\overbrace{\underbrace{0\dots0}_{x_1+\dots+x_n}\ \ 1\dots1}^{a+1}][\underbrace{0\dots0}_{a}\ 1]^T = \begin{cases} 1, & \sum_{i=1}^{n} x_i \le a, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}
$$

Experiments

The TT-method for the MAP-inference:

1. Convert the energy into the TT-format;
2. Find the minimal element in the energy tensor.

Experiments

The TT-method for the MAP-inference:

1. Convert the energy into the TT-format;
2. Find the minimal element in the energy tensor.

We compare the TT-method with the popular TRW-S algorithm on several real-world image segmentation problems from the OpenGM database.

Experiments

The TT-method for the MAP-inference:

1. Convert the energy into the TT-format;
2. Find the minimal element in the energy tensor.

We compare the TT-method with the popular TRW-S algorithm on several real-world image segmentation problems from the OpenGM database.

Problem	**Variables**	**Labels**	**TRW-S**	**TT**	**Time** (sec)

Experiments

The TT-method for the MAP-inference:

1. Convert the energy into the TT-format;
2. Find the minimal element in the energy tensor.

We compare the TT-method with the popular TRW-S algorithm on several real-world image segmentation problems from the OpenGM database.

Problem	**Variables**	**Labels**	**TRW-S**	**TT**	**Time** (sec)
gm6	320	3	45.03	43.11	637
gm29	212	3	56.81	56.21	224
gm66	198	3	75.19	74.92	172
gm105	237	3	67.81	67.71	230
gm32	100	7	150.50	289.29	257
gm70	122	7	121.78	163.60	399
gm85	143	7	168.30	228.40	1 912
gm192	99	7	114.51	174.78	180

Future Work

- Optimized implementation of the proposed method.

Future Work

- Optimized implementation of the proposed method.
- Analytical formulas of TT-representations for other types of high order potentials.

Future Work

- Optimized implementation of the proposed method.
- Analytical formulas of TT-representations for other types of high order potentials.
- Better algorithm for finding the minimal element in a tensor represented in the TT-format.

Conclusion

- We have proposed an algorithm that converts MRF energy into the TT-format exactly.
- We have derived an upper bound on the TT-ranks of the energy tensor constructed by the proposed algorithm.
- We have demonstrated how the obtained TT-representation of MRF energy can be used for solving the important problem of the MAP-inference arising in probabilistic graphical models.
- To improve the method, we need a better algorithm for finding the minimal element in a tensor represented in the TT-format.

Thank you!