**Objective**

**What:**
- Fast and parallel algorithms for dense graphical models

**Why:**
- Dense graphical models are more expressive
- CNN+CRF training
- Huge datasets and Real Time Applications

---

**MAP Inference**

\[ y^* = \arg \min_{y \in Y} E(y | \theta) = \sum_{v \in V} \theta_v(y_v) + \sum_{u \in E} \theta_{uv}(y_{uv}) \]

\[ \theta_v \rightarrow \text{node potential}, \]
\[ \theta_{uv} \rightarrow \text{pairwise potential}, \]
\[ y^* \rightarrow \text{optimal labelling}. \]

**Dual LP**

To be able to deal with arbitrary potentials, we address the dual problem:

\[ D(\phi) := \sum_{v \in V} \min_{y \in Y_v} \phi_v(y_v) + \sum_{u \in E} \min_{y \in Y_u} \phi_{uv}(y_{uv}). \]  

(1)

\[ \phi_v(s) := \theta_v(s) + \sum_{u \in \partial \setminus \partial_v} \phi_{u-v}(s) \]
\[ \phi_{uv}(s, t) := \theta_{uv}(s, t) - \phi_{v-u}(s) - \phi_{u-v}(t). \]

Dual variables \( \phi_{v \rightarrow u} \) and \( \phi_{u \rightarrow v} \) are the Lagrange multipliers.

**Motivation for Algorithm Structure**

- The subgradient for the dual is sparse. Block Coordinate Ascent methods are more efficient than sub-gradient based methods.
- TRWS [1] is the best performer for MAP inference.
- Dense graphs need no large sub-problem decompositions.

**Comparing BCA Updates**

\[ g_{uv}(s, t) = \theta_{uv}(s, t) + \theta_v(s) + \theta_u(t) \]

\[ \theta^M_v(s) := \frac{1}{2} \min_{y \in Y_v} g_{uv}(s, t), \forall s \in Y_v; \]
\[ \theta^M_t(t) := \frac{1}{2} \min_{y \in Y_t} g_{uv}(s, t), \forall t \in Y_t. \]

\[ \theta^P_v(s) := \theta_v(s), \theta^P_t(t) := \theta_t(t) \]

\[ \theta^P(0) = (\theta^M + \theta^P), \forall s \in Y_v; \]
\[ \theta^P(1) = (\theta^M + \theta^P), \forall t \in Y_t. \]

**We prove**

With the same input, at the end of the first iteration, \( D(\phi^M) \leq D(\phi^P). \)

**Parallelization: CPU & GPU**

Non-incident edges can be processed in parallel. \( \rightarrow \) We use maximum matching solvers.

---

**Conclusions and Outlook**

- Our approach is the state-of-the-art method for dense graphical models (> 10% graph density), beating even TRWS.
- We give CPU and GPU parallel implementations provided at XXXX.

---

**References**


**Acknowledgements**

This project has received funding from the ERC under the European Union’s Horizon 2020 research and innovation program (grant agreement No 649966). A. Shulchenko was supported by Czech Science Foundation grant 18-09138S.