

# Spatial Inference Machines

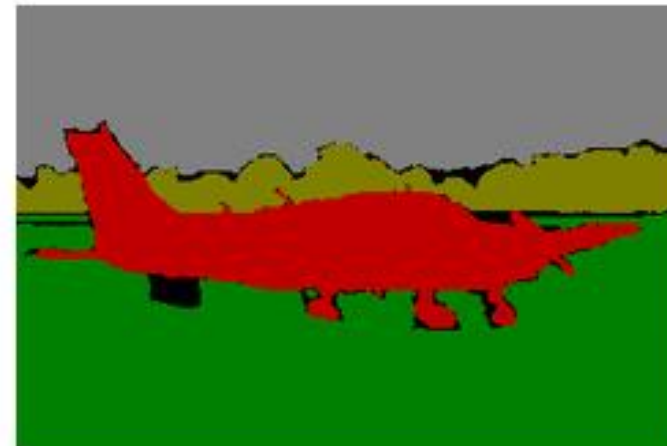
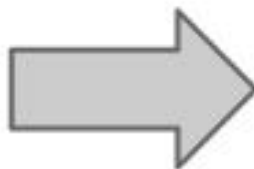
---

КУЗЬМИН АЛЕКСЕЙ

# Постановка задачи



↑ Image from the MSRC dataset



← разбиение на суперпиксели

# Постановка задачи

---

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}) =$$

← обратная энергия,  
определена на графе

$$\sum_{i \in \mathcal{V}} \sum_{k=1}^K [y_i = k] (\mathbf{x}_i^T \mathbf{w}_k^u) + \sum_{(i,j) \in \mathcal{E}} [y_i = y_j] (\mathbf{x}_{ij}^T \mathbf{w}^p)$$

метки  
↑  
суперпикселей (напр.  
1 — самолёт,  
2 — трава, и т.д.)

↑ признаки  
суперпикселей  
(цвет, текстура,  
и т.д.)

↑ признаки пар  
суперпикселей  
(обычно  
«похожесть»)

# Графические модели

---

- $f$  - фактор
- $\varphi_f$  - потенциальная функция
- $p(\mathbf{y}) = \frac{1}{Z} \prod_{f \in F} \varphi_f(\cup_{v \in f} \{y_v\})$

# Получение меток пикселей

---

- Маргинальные распределения на каждую метку

$$p(y_v) = \int p(y_v, \mathbf{y}_{\setminus v}) d\mathbf{y}_{\setminus v}$$

- Мода распределения

$$\mathbf{y} = \operatorname{argmax}_{\bar{\mathbf{y}}} p(\bar{\mathbf{y}})$$

# Мотивация

---

Зачем нам придумывать что-то новое?

- Обучение модели часто занимает очень много времени

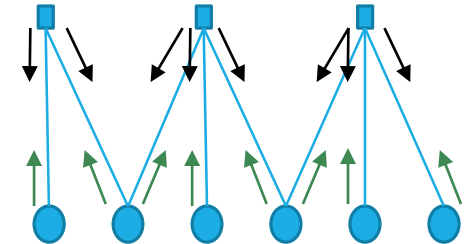
# Передача сообщений

---

$$\mu_{f \rightarrow v}^n(y_v) = \sum_{\mathbf{y}': y'_v = y_v} \phi(\mathbf{y}') \prod_{v' \in f\{v\}} \mu_{v' \rightarrow f}^n(y'_{v'})$$

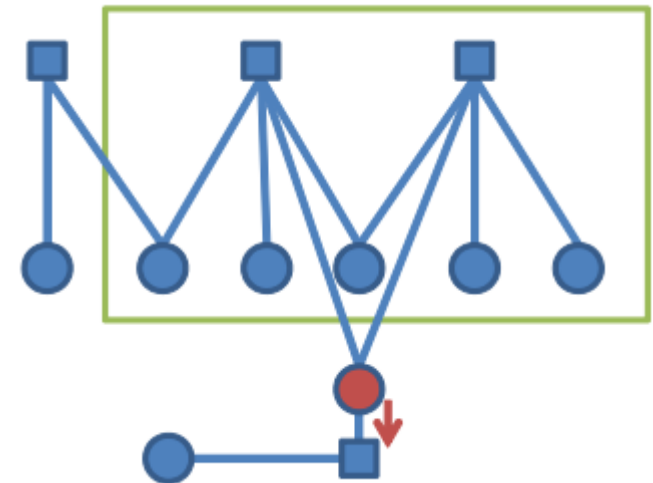
$$\mu_{v \rightarrow f}^n(y_v) = \prod_{f': v \in f', f' \neq f} \mu_{f' \rightarrow v}^{n-1}(y_v)$$

$$p(y_v) = \prod_{f: v \in f} \mu_{f \rightarrow v}^n(y_v)$$



# Ross: объединение двух сообщений

$$\mu_{v \rightarrow f}^n(y_v) = \bar{g}_n \left( \mathbf{x}_f, \bigoplus_{\substack{f': \\ \exists v': v' \in f, \\ v' \in f', f' \neq f}} \mathbf{x}_{f'}, \bigoplus_{\substack{v'' \in f', f': \\ \exists v': v' \in f, \\ v' \in f', f' \neq f}} \mu_{v'' \rightarrow f'}^{n-1}(y_{v''}) \right).$$





# Наш метод

---

# Обозначения Spartial Inference Machines

---

- D-factor – пара  $p = (d_f, S_f)$ 
  - Обобщение парных потенциалов
- $t(f)$  – тип фактора

- Belief:  $p_n(y_v) = \prod_{f:d_f=v} \left( \mu_{S_f \rightarrow v}^n \right)^{\alpha_{t(f)}^n}$

- Сообщения:

$$\mu_{S_f \rightarrow d_f}^n \left( y_{d_f} \right) = g_{n,t(f)} \left( p_{n-1} \left( y_{d_f} \right), \mathbf{x}_{d_f}, \mathbf{x}_f, \mathbf{x}_{S_f}, \mathbb{E}_{v \sim S_f} p_{n-1} \left( y_v \right) \right)$$

# Структурные факторы

---



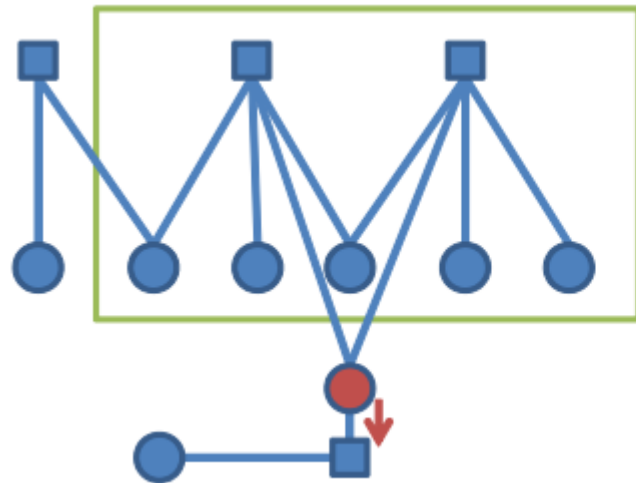
# Пространственные факторы

---

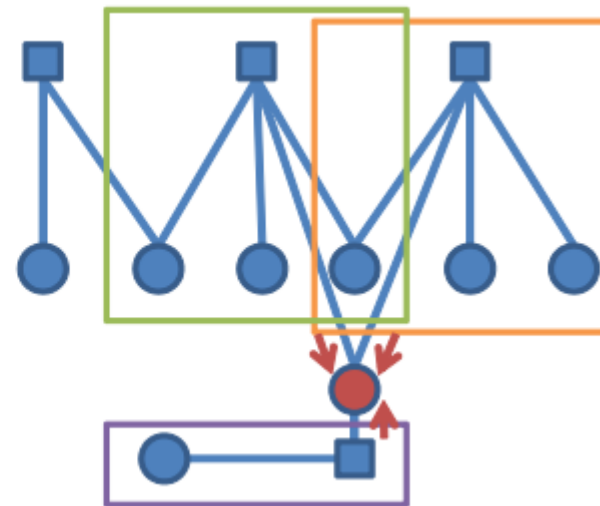


# Сравнение нашего метода и метода Ross

$$\mu_{v \rightarrow f}^n(y_v)$$



$$\mu_{S_f \rightarrow d_f}^n(y_{d_f})$$



# Обучение

---

- Попеременное обучение  $g$  и весов  $\alpha$ .
- $g$  в нашем случае RandomForest
- Обучение  $\alpha$ :

$$\max_{\alpha \geq 0} \sum_v \left( \frac{\prod_{f:d_f=v} \left( \mu_{S_{f \rightarrow v}}^n(y_v) \right)^{\alpha_{t(f)}}}{\sum_{l \in L} \prod_{f:d_f=l} \left( \mu_{S_{f \rightarrow l}}^n(y_l) \right)^{\alpha_{t(f)}}} + C \sum_{t \in T} \alpha_t \right)$$

- Для преодоления переобучения используется k-fold cross-validation.

# Алгоритм

---

На каждой итерации

Для каждого фолда

Для каждого фактора обучить  $g$  на данных всех остальных фолдов

Подсчитать сообщения этого фолда

Обучить  $g$  на всех данных

Обучить  $\alpha$

Подсчитать beliefs

# Будущие эксперименты

---

- База картинок MSRC
- Структурные факторы
- Пространственные факторы: локальный, «выше», «ниже», «правее», «левее».