



GRAPHICS & MEDIA LAB
VISION GROUP



Cutting-Plane Training of Non-associative Markov Network for 3D Point Cloud Segmentation



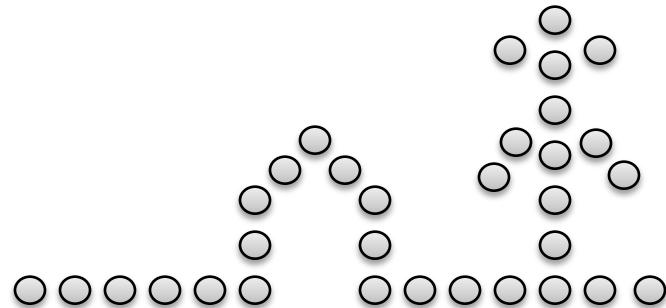
Roman Shapovalov, Alexander Velizhev
Lomonosov Moscow State University



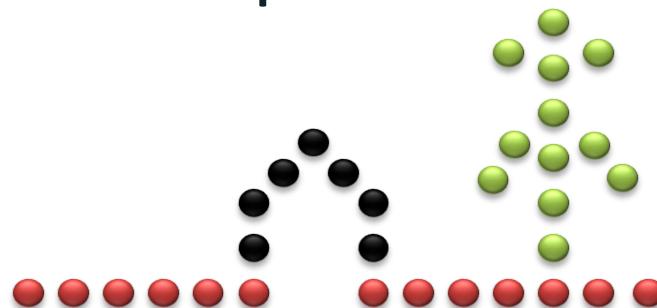
Hangzhou, May 18, 2011

Semantic segmentation of point clouds

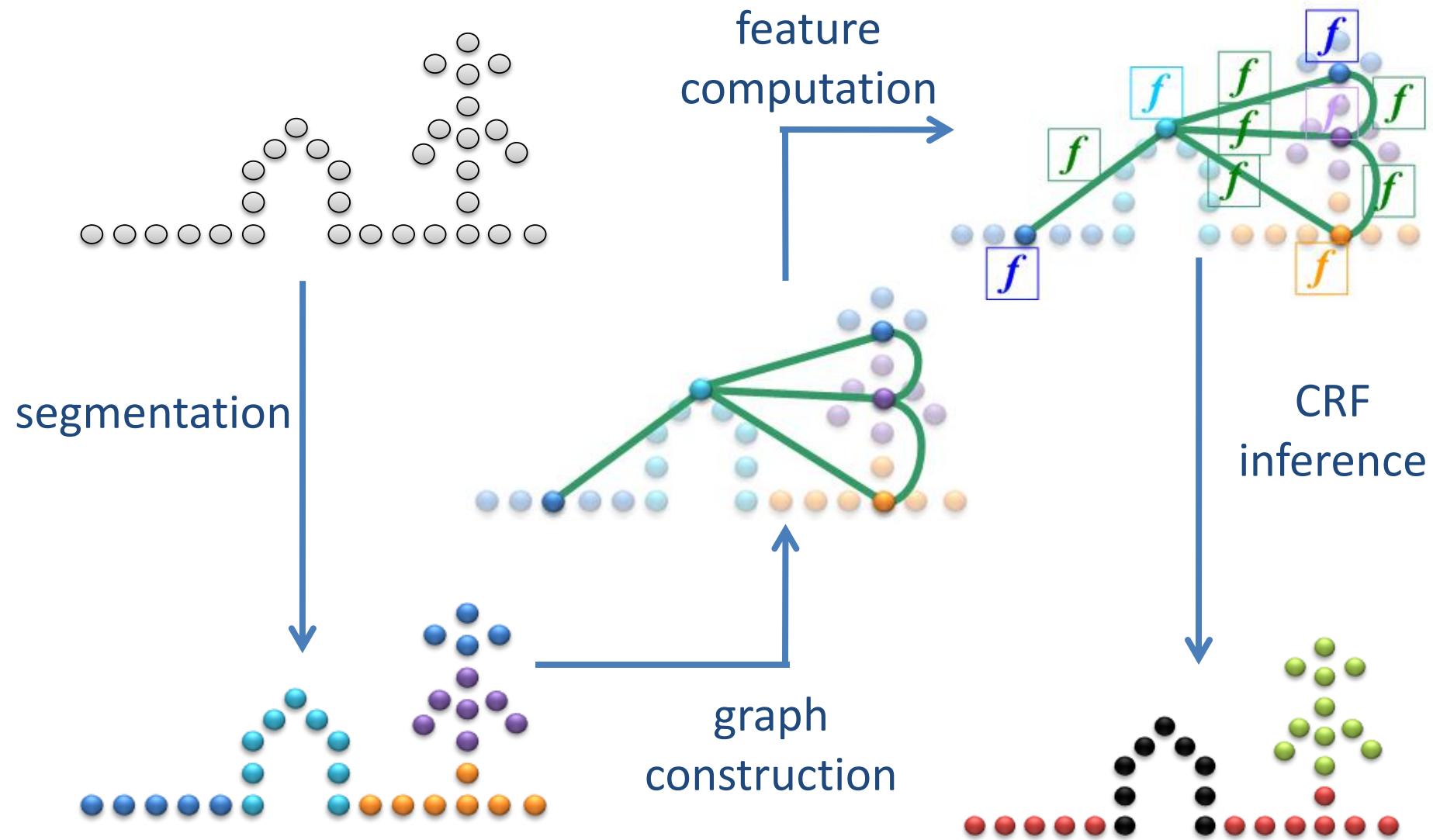
- LIDAR point cloud without color information



- Class label for each point

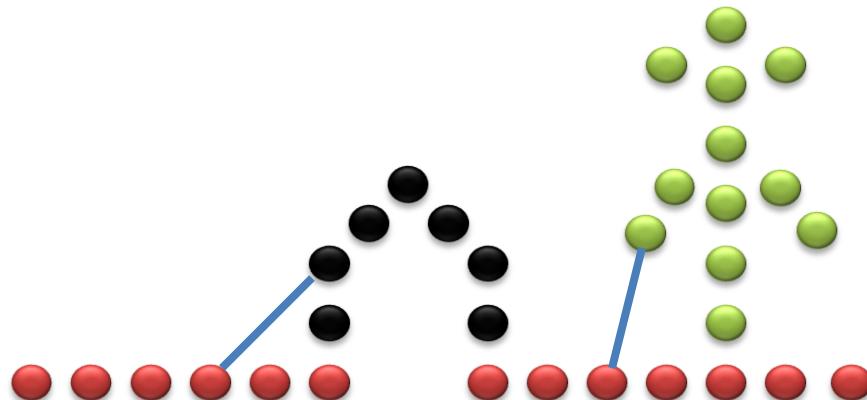


System workflow



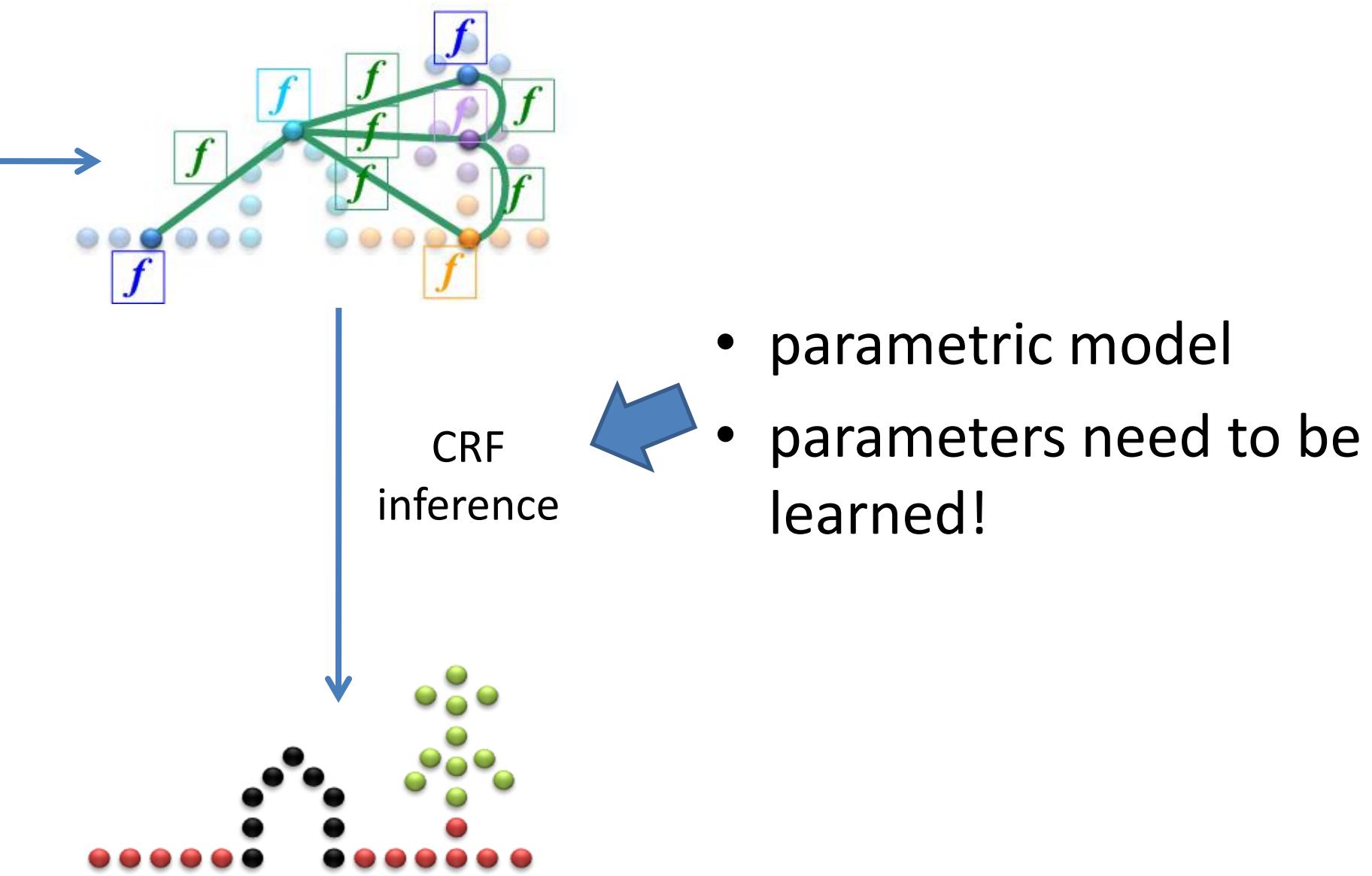
Non-associative CRF

- Associative CRF: $\phi(\mathbf{x}_{ij}, y_i, y_j) \leq \phi(\mathbf{x}_{ij}, k, k)$
 - Our model: no such constraints!



[Shapovalov et al., 2010]

CRF training



Structured learning

[Anguelov et al., 2005; and a lot more]

- Linear model: $\phi(\mathbf{x}_i, y_i) = \mathbf{w}_{n,i}^T \mathbf{x}_i y_i$
 - CRF negative energy: $\mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}) \rightarrow \max_{\mathbf{y}}$
 - Find \mathbf{w} such that

$$\mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...}) > \mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...})$$

$$\mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...}) > \mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...})$$

$$\mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...}) > \mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...})$$

$$\dots$$

$$\mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...}) > \mathbf{w}^T \Psi(\text{... } \triangle \text{ } \triangle \text{ } \triangle \text{ } \text{...}, \text{... } \circ \text{ } \circ \text{ } \circ \text{ } \text{...})$$

Structured loss

- Define $\mathbf{x} = \text{features}(\dots \triangle \triangle \dots)$
- Define structured loss, for example:

$$\Delta(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i \in N} [y_i \neq \bar{y}_i]$$

- Find \mathbf{w} such that

$$\mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) > \mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) + \Delta(\dots \triangle \triangle \dots, \dots \triangle \triangle \dots)$$
$$\mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) > \mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) + \Delta(\dots \triangle \triangle \dots, \dots \triangle \triangle \dots)$$
$$\mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) > \mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) + \Delta(\dots \triangle \triangle \dots, \dots \triangle \triangle \dots)$$

...

$$\mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) > \mathbf{w}^T \Psi(\mathbf{x}, \dots \triangle \triangle \dots) + \Delta(\dots \triangle \triangle \dots, \dots \triangle \triangle \dots)$$

Cutting-plane training

- A lot of constraints (K^n)

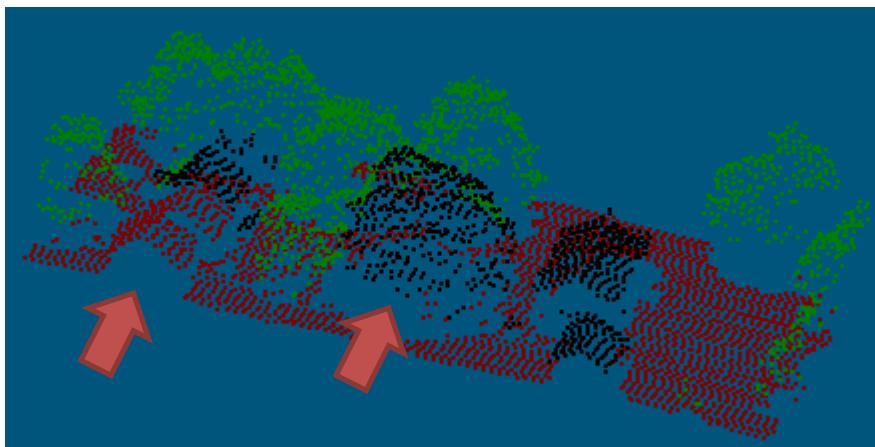
$$\mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}) > \mathbf{w}^T \Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}}), \forall \bar{\mathbf{y}}$$

- Maintain a working set
- Add iteratively the most violated one:

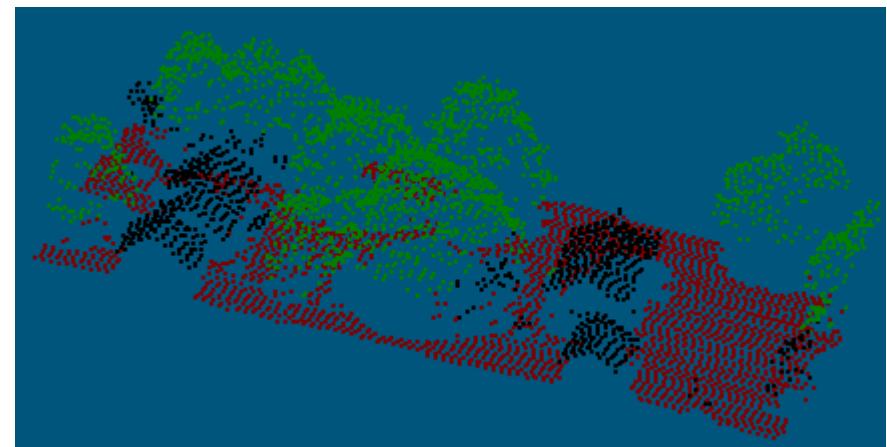
$$\bar{\mathbf{y}} = \arg \max_{\bar{\mathbf{y}}} \left[\mathbf{w}^T \Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}}) \right]$$

- Polynomial complexity
- SVM^{struct} implementation [Joachims, 2009]

Results

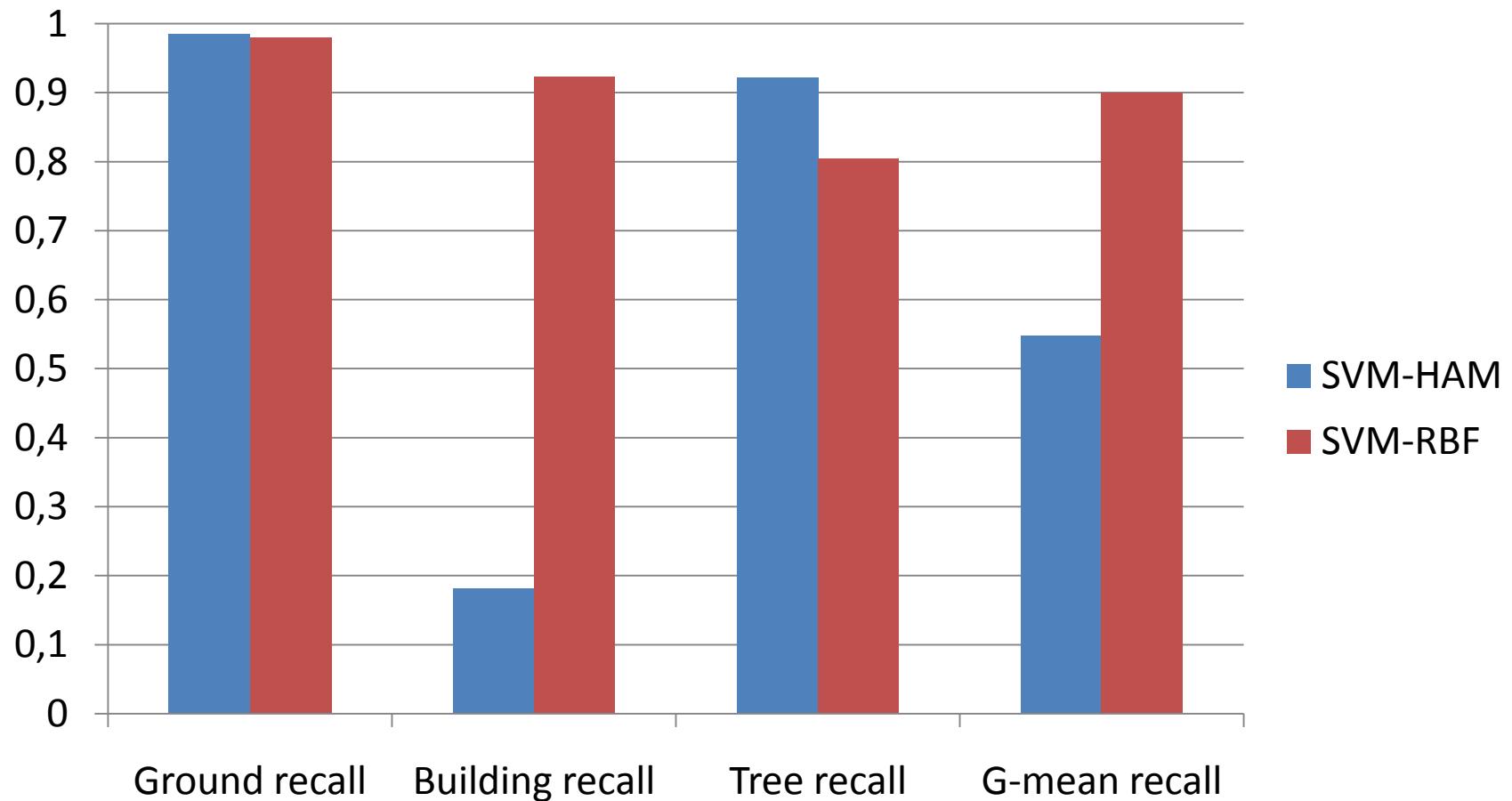


[Munoz et al., 2009]



Our method

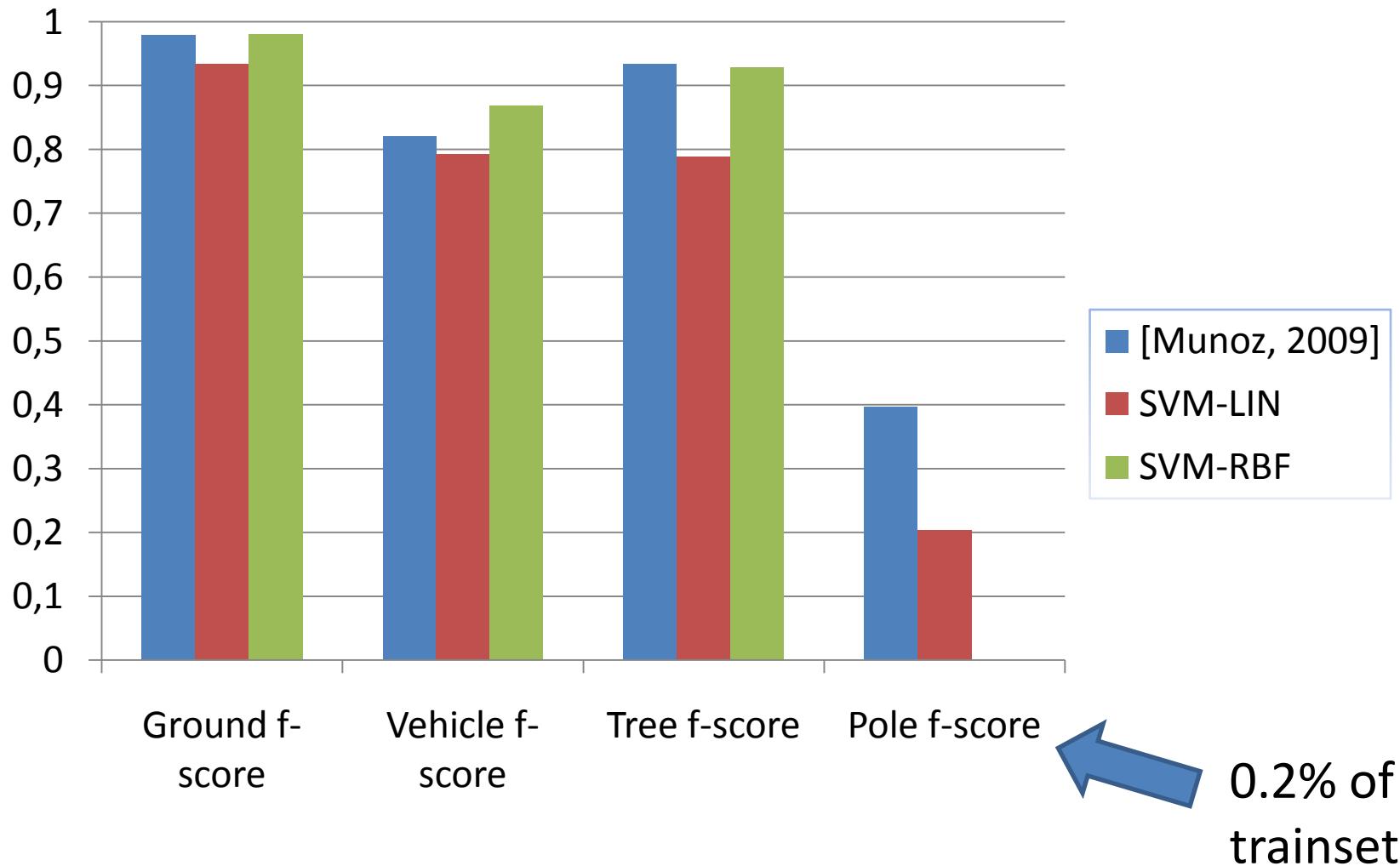
Results: balanced loss better than the Hamming one



Results: RBF better than linear



Results: fails at very small classes



Analysis

- Advantages:
 - more flexible model
 - accounts for class imbalance
 - allows kernelization
- Disadvantages:
 - really slow (esp. with kernels)
 - learns small/underrepresented classes badly