Learning to Place New Objects

Yun Jiang, Changxi Zheng, Marcus Lim, Ashutosh Saxena.

Cornell University.
Placing Objects

- Daily tasks:
  - Setting a dinner table.
  - Arranging grocery in a fridge.
  - Packing items in boxes.
  - Organizing closets.

Grasping and placing objects.

- Placing is an important skill for a personal robot.
- However, it has heretofore been little-studied.
Related Work

- **Identify flat surfaces.**
  - Surface (green) vs clutter (red).
  - Place *upright* objects on flat areas using tactile feedback.

Schuster et al., 2010
Flat Surface Heuristic?

- It is not always as easy as finding flat surface
Challenges

- Different configurations in different placing situations.
  - Plate: vertical in a dish-rack; slanted on a support.
  - Glasses: upright on table, upside-down on stemware holder,

- Stacking objects.
Problem Specification

- Input: Point cloud of an object (e.g., plate) and placing environment (e.g., a dish rack).

- Output: a **stable** and **preferred** placement specified by 3D location and 3D orientation of the object.
  - **Stability**
    - stay still after placing and stand small perturbation.
  - **Preference**
    - E.g., plates and pens are flat on a table, but vertical in a dish-rack and a pen-holder.
Brute-force Approach

- Test all orientations and locations randomly using a physics-based simulator.
- Too many configurations to try.
- Does not address the preferred placement issue.

Learning needed!
Learning Approach

Supervised Learning.
- Features $X$ for placement $\xi$
- Learning algorithm: $X \rightarrow y$
- Choose $\xi$ with highest $y$. 

Learning to Place New Objects. Jiang, Zheng, Lim, Saxena.
Features for Learning Algorithm

- **Supporting contacts**
  - Falling distance, #contacts, variance of contacts, etc.

- **Caging features**
  - Height of the neighborhood around the placement.

- **Signatures of the object and placing environment**
  - Histogram of #points in object/environment point cloud.
  - Distance ratio of obj/env along different directions.

Total 120 features.
Learning Algorithm

- Support Vector Machines.
  - Maximize the minimum margin.

\[
\min_{\omega, b} \quad \frac{1}{2} \|\omega\|^2_2 + C \sum_{j=1}^{n_i} \xi_j \\
\text{subject to} \quad Y_j (\omega^T X_j + b) \geq 1 - \xi_j, \quad \xi_j \geq 0 \\
\forall \ 1 \leq j \leq n_i
\]

- Parameters: \( \omega, b \).
- Features \( X \)
- Label \( y \)
Learning Experiments.

- **Data**
  - 7 placing areas and 8 objects.
  - For each object-environment pair, 100 distinct 3D locations with 18 different orientations. This gives 1800 different placements.
  - Total 37655 collision-free placements.

- **Metrics:**
  - $R_0$: rank of first correct placement.
  - Prec@5: precision of top 5 predictions.
Learning Results

- **Results with different features.**
  - **Same Environment Same Object (SESO).**

<table>
<thead>
<tr>
<th></th>
<th>chance</th>
<th>contact</th>
<th>caging</th>
<th>signature</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_0$</td>
<td>29.4</td>
<td>13.3</td>
<td>5.0</td>
<td>2.6</td>
<td>1.0</td>
</tr>
<tr>
<td>Pre@5</td>
<td>0.10</td>
<td>0.64</td>
<td>0.69</td>
<td>0.82</td>
<td>0.96</td>
</tr>
</tbody>
</table>

- **In New Environment New Object (NENO) setting.**
  - $R_0$ is 9.4.
  - Pre@5 is 0.54.
  - (This is unacceptable.)
Learning Algorithm: Multiple Models

- Intrinsic difference between different placing settings

- Independent models in conventional SVM
  - $r$ models: $w_i$

$$\min_{\omega_i, b_i, i=1,...,r} \sum_{i=1}^{r} \left( \frac{1}{2} \|\omega_i\|^2_2 + C \sum_{j=1}^{n_i} \xi_{i,j} \right)$$

subject to

$$Y_i^j (\omega_i^T X_i^j + b_i) \geq 1 - \xi_{i,j}, \quad \xi_{i,j} \geq 0$$

$\forall 1 \leq i \leq r, \ 1 \leq j \leq n_i$
Results: Independent Models.

- New Environment New Object (NENO).

<table>
<thead>
<tr>
<th></th>
<th>chance</th>
<th>joint</th>
<th>independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_0$</td>
<td>29.4</td>
<td>9.4</td>
<td>5.3</td>
</tr>
<tr>
<td>Pre@5</td>
<td>0.10</td>
<td>0.54</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Learning to Place New Objects. Jiang, Zheng, Lim, Saxena.

Shared Sparsity SVM

- **Introduce sparsity** (Jalali et al., NIPS 2010)
  - Each model $w_i$ is composed of $S_i$ and $B_i$
    $$w_i = S_i + B_i$$
  - $S_i$ is self-owned features reflecting individual difference
  - $B_i$ represents shared sparsity structure

- **Shared sparsity SVM**

  $$\min_{\omega_i, b_i, i=1, \ldots, r} \sum_{i=1}^{r} \left( \frac{1}{2} \|\omega_i\|_2^2 + C \sum_{j=1}^{n_i} \xi_{i,j} \right) + \lambda_S \|S\|_{1,1} + \lambda_B \|B\|_{1,\infty}$$
  subject to
  $$Y_i^j (\omega_i^T X_i^j + b_i) \geq 1 - \xi_{i,j}, \quad \xi_{i,j} \geq 0$$
  $$\forall 1 \leq i \leq r, 1 \leq j \leq n_i$$

  where
  $$\|S\|_{1,1} = \sum_{i,j} |S_i^j| \quad \text{and} \quad \|B\|_{1,\infty} = \sum_{j=1}^p \max_i |B_i^j|$$
Comparison of different algorithms.

- New Environment New Object (NENO).

<table>
<thead>
<tr>
<th></th>
<th>chance</th>
<th>joint</th>
<th>independent</th>
<th>shared</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_0$</td>
<td>29.4</td>
<td>9.4</td>
<td>5.3</td>
<td>1.9</td>
</tr>
<tr>
<td>Pre@5</td>
<td>0.10</td>
<td>0.54</td>
<td>0.61</td>
<td>0.66</td>
</tr>
</tbody>
</table>
Learning Experiments: Results.

<table>
<thead>
<tr>
<th>environment</th>
<th>objects</th>
<th>chance</th>
<th>flat surface</th>
<th>lowest pt</th>
<th>Independent SVM with voting</th>
<th>shared sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$R_0$</td>
<td>Pre@5</td>
<td>$R_0$</td>
<td>Pre@5</td>
<td>$R_0$</td>
</tr>
<tr>
<td>rack-1</td>
<td>plate,mug,martini,bowl</td>
<td>3.8</td>
<td>0.15</td>
<td>1.8</td>
<td>0.50</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rack-2</td>
<td>plate,mug,martini,bowl</td>
<td>5.0</td>
<td>0.25</td>
<td>18.3</td>
<td>0.05</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rack-3</td>
<td>plate,mug,martini,bowl</td>
<td>4.8</td>
<td>0.15</td>
<td>4.8</td>
<td>0.20</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flat</td>
<td>plate,mug,martini,bowl,</td>
<td>6.6</td>
<td>0.08</td>
<td>1.0</td>
<td>0.98</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>candy cane, disc, spoon,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tuning fork</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pen holder</td>
<td>candy cane, disc, spoon,</td>
<td>128.0</td>
<td>0.00</td>
<td>61.3</td>
<td>0.05</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>tuning fork</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hook</td>
<td>candy cane, disc</td>
<td>78.0</td>
<td>0.00</td>
<td>42.0</td>
<td>0.00</td>
<td>1.0</td>
</tr>
<tr>
<td>stemware</td>
<td>martini</td>
<td>18.0</td>
<td>0.00</td>
<td>65.0</td>
<td>0.00</td>
<td>1.0</td>
</tr>
<tr>
<td>holder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>29.4</td>
<td>0.10</td>
<td>18.6</td>
<td>0.41</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chance: 29.4
Flat surface: 18.6
Our method: 1.9
Robotic Experiments

Panda (PersonAI Non-Deterministic Arm.)

Some test objects

Some test environments
Results: Robotic Experiments.

<table>
<thead>
<tr>
<th>environment</th>
<th>object</th>
<th>SESO</th>
<th></th>
<th>NENO</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$R_0$</td>
<td>$P_s$ (%)</td>
<td>$P_p$ (%)</td>
<td>$R_0$</td>
</tr>
<tr>
<td>rack-1</td>
<td>plate</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td>martini</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>bowl</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>9.4</td>
</tr>
<tr>
<td>rack-3</td>
<td>plate</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>martini</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>bowl</td>
<td>1.0</td>
<td>80</td>
<td>80</td>
<td>1.0</td>
</tr>
<tr>
<td>flat</td>
<td>plate</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>martini</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>bowl</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>1.8</td>
</tr>
<tr>
<td>stemware</td>
<td>martini</td>
<td>1.0</td>
<td>100</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td>holder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>1.0</td>
<td>98</td>
<td>98</td>
<td>2.5</td>
</tr>
</tbody>
</table>
**Challenges**

- Different configuration in different placing situations.
  - Plate: vertical in a dish-rack; slanted on a support.
  - Glasses: upright on table, upside-down on stemware holder.
- Stacking multiple objects.
Placing Multiple Objects (Future Work)

- Multiple objects and several placing environments.
  - Objects can choose different environments.
  - Objects can stack on another.
Multiple Object: Results. (Future Work)
Learning to Place New Objects. Jiang, Zheng, Lim, Saxena.
Questions?

Download code and data at:

http://pr.cs.cornell.edu/placingobjects

(Cornell Personal Robotics.)
Thank you