Lecture 6:
Outlook / Related topics
Version July 2, 2019

Multi-Object Tracking

Lennart Svensson

Section 1:
Extended object tracking

Multi-Object Tracking

Lennart Svensson

Extended object tracking – motivation

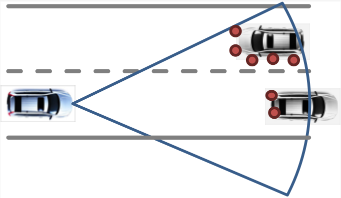
Multi-Object Tracking

Lennart Svensson

EXTENDED OBJECT TRACKING – DEFINITION

Extended object tracking

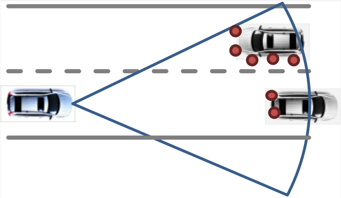
- Tracking objects that may generate multiple detections per time step.



EXTENDED OBJECT TRACKING – DEFINITION

Extended object tracking

- Tracking objects that may generate multiple detections per time step.

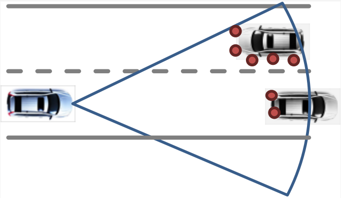


- An extended object occupies more than one sensor resolution cell.

EXTENDED OBJECT TRACKING – DEFINITION

Extended object tracking

- Tracking objects that may generate multiple detections per time step.

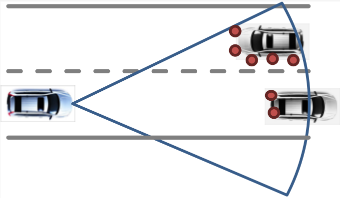


- An extended object occupies more than one sensor resolution cell.
- Are cars extended?

EXTENDED OBJECT TRACKING – DEFINITION

Extended object tracking

- Tracking objects that may generate multiple detections per time step.



- An extended object occupies more than one sensor resolution cell.
- Are cars extended?
  Depends on sensor and distance to sensor.

WHY EXTENDED OBJECT TRACKING?

- In autonomous applications, extended object tracking is useful for:

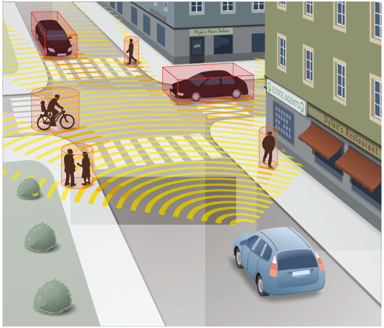


Illustration by Per Thorneus, reproduced from Granström et al (2013), "Random set methods: Estimation of multiple extended objects", *IEEE robotics & automation magazine* 21(2), 73-82.

WHY EXTENDED OBJECT TRACKING?

- In autonomous applications, extended object tracking is useful for:
  - tracking vehicles using radar or lidar sensors,

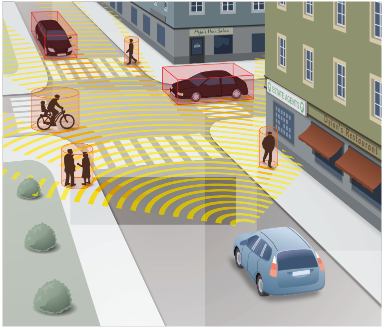


Illustration by Per Thorneus, reproduced from Granström et al (2013), "Random set methods: Estimation of multiple extended objects", *IEEE robotics & automation magazine* 21(2), 73-82.

WHY EXTENDED OBJECT TRACKING?

- In autonomous applications, extended object tracking is useful for:
  - tracking vehicles using radar or lidar sensors,
  - mapping and localization using, e.g., radar sensors.

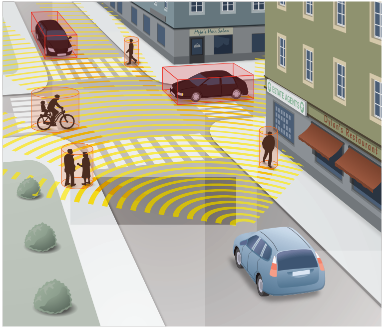


Illustration by Per Thorneus, reproduced from Granström et al (2013), “Random set methods: Estimation of multiple extended objects”, *IEEE robotics & automation magazine* 21(2), 73-82.

WHY EXTENDED OBJECT TRACKING?

- In autonomous applications, extended object tracking is useful for:
  - tracking vehicles using radar or lidar sensors,
  - mapping and localization using, e.g., radar sensors.
    Note: we are “tracking” stationary objects such as buildings, traffic signs and guard rails.

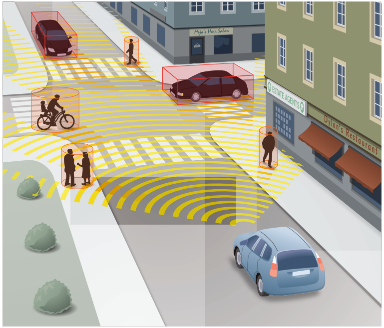
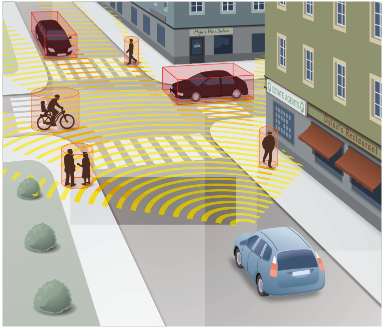


Illustration by Per Thorneus, reproduced from Granström et al (2013), “Random set methods: Estimation of multiple extended objects”, *IEEE robotics & automation magazine* 21(2), 73-82.

WHY EXTENDED OBJECT TRACKING?

- In autonomous applications, extended object tracking is useful for:
  - tracking vehicles using radar or lidar sensors,
  - mapping and localization using, e.g., radar sensors.
    Note: we are “tracking” stationary objects such as buildings, traffic signs and guard rails.



- In general, multiple detections from a single object may enable us to estimate the object’s **shape** and **orientation**.

Illustration by Per Thorneus, reproduced from Granström et al (2013), “Random set methods: Estimation of multiple extended objects”, *IEEE robotics & automation magazine* 21(2), 73-82.

WHAT'S NEW?

Bayesian filtering recursions – standard equations

Prediction: $$p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})\, \delta\mathbf{x}_{k-1}$$

Update: $$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k \mid \mathbf{x}'_k)\, p(\mathbf{x}'_k \mid \mathbf{z}_{1:k-1})\, \delta\mathbf{x}'_k}$$

WHAT'S NEW?

Bayesian filtering recursions – standard equations

Prediction: $$p(\mathbf{x}_k \middle| \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \middle| \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \middle| \mathbf{z}_{1:k-1})\, \delta\mathbf{x}_{k-1}$$

Update: $$p(\mathbf{x}_k \middle| \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \middle| \mathbf{x}_k) p(\mathbf{x}_k \middle| \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k \middle| \mathbf{x}'_k) p(\mathbf{x}'_k \middle| \mathbf{z}_{1:k-1})\, \delta\mathbf{x}'_k}$$

Differences:

- Measurement model $p(\mathbf{z}_k | \mathbf{x}_k)$ is different, since the single objects can generate multiple measurements.

WHAT'S NEW?

Bayesian filtering recursions – standard equations

Prediction: $$p(\mathbf{x}_k \big| \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \big| \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \big| \mathbf{z}_{1:k-1}) \, \delta\mathbf{x}_{k-1}$$

Update: $$p(\mathbf{x}_k \big| \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \big| \mathbf{x}_k) p(\mathbf{x}_k \big| \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k \big| \mathbf{x}'_k) p(\mathbf{x}'_k \big| \mathbf{z}_{1:k-1}) \, \delta\mathbf{x}'_k}$$

Differences:

- Measurement model $p(\mathbf{z}_k \big| \mathbf{x}_k)$ is different, since the single objects can generate multiple measurements.
- Data association hypothesis trees are different which requires new algorithms.

WHAT'S NEW?

Bayesian filtering recursions – standard equations

Prediction: $$p(\mathbf{x}_k \middle| \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \middle| \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \middle| \mathbf{z}_{1:k-1})\, \delta \mathbf{x}_{k-1}$$

Update: $$p(\mathbf{x}_k \middle| \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \middle| \mathbf{x}_k) p(\mathbf{x}_k \middle| \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k \middle| \mathbf{x}'_k) p(\mathbf{x}'_k \middle| \mathbf{z}_{1:k-1})\, \delta \mathbf{x}'_k}$$

Differences:

- Measurement model $p(\mathbf{z}_k | \mathbf{x}_k)$ is different, since the single objects can generate multiple measurements.
- Data association hypothesis trees are different which requires new algorithms.
- The single object state x_k often contains shape information.

Single object measurement models

Multi-Object Tracking

Lennart Svensson

MULTI-BERNOULLI RFSs

- **Objective:** model single object measurements, $\mathbf{o}_k \Big| \{x_k\}$.

MULTI-BERNOULLI RFSs

- **Objective:** model single object measurements, $\mathbf{o}_k \Big| \{x_k\}$.
- Number of measurements? Spatial distribution?

MULTI-BERNOULLI RFSs

- **Objective:** model single object measurements, $\mathbf{o}_k \big| \{x_k\}$.
- Number of measurements? Spatial distribution?

Multi-Bernoulli (MB) models

- Given detailed knowledge about object: model $\mathbf{o}_k \big| \{x_k\}$ as a MB RFS.

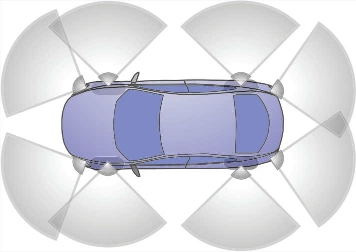


Image from Hammarstrand et al (2012), "Extended object tracking using a radar resolution model", *IEEE TAES* 48(3).

MULTI-BERNOULLI RFSs

- **Objective:** model single object measurements, $\mathbf{o}_k \big| \{x_k\}$.
- Number of measurements? Spatial distribution?

Multi-Bernoulli (MB) models

- Given detailed knowledge about object: model $\mathbf{o}_k \big| \{x_k\}$ as a MB RFS.
- One Bernoulli for each "reflector point".

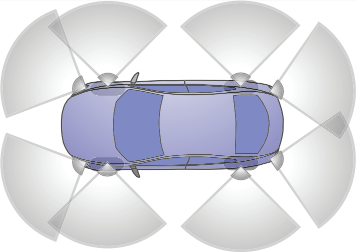


Image from Hammarstrand et al (2012), "Extended object tracking using a radar resolution model", *IEEE TAES* 48(3).

MULTI-BERNOULLI RFSs

- **Objective:** model single object measurements, $\mathbf{o}_k \big| \{x_k\}$.
- Number of measurements? Spatial distribution?

Multi-Bernoulli (MB) models

- Given detailed knowledge about object: model $\mathbf{o}_k \big| \{x_k\}$ as a MB RFS.
- One Bernoulli for each "reflector point".
- Both r^i and $p^i(o|x_k)$ depend on x_k.

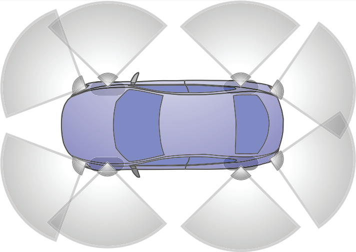


Image from Hammarstrand et al (2012), "Extended object tracking using a radar resolution model", *IEEE TAES* 48(3).

THE POISSON POINT PROCESS AND ITS INTENSITY FUNCTION

- The PPP is arguably the **standard object measurement model** for extended object tracking.

THE POISSON POINT PROCESS AND ITS INTENSITY FUNCTION

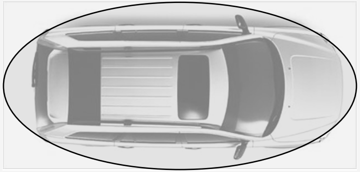
- The PPP is arguably the **standard object measurement model** for extended object tracking.
- How can we **model the intensity function** $\lambda_o(o|x_k)$?

THE POISSON POINT PROCESS AND ITS INTENSITY FUNCTION

- The PPP is arguably the **standard object measurement model** for extended object tracking.
- How can we **model the intensity function** $\lambda_o(o|x_k)$?

Random matrix model

- Approximate $\lambda_o(o|x_k)$ as a weighted Gaussian.

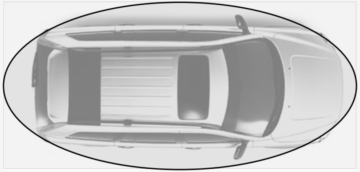


THE POISSON POINT PROCESS AND ITS INTENSITY FUNCTION

- The PPP is arguably the **standard object measurement model** for extended object tracking.
- How can we **model the intensity function** $\lambda_o(o|x_k)$?

Random matrix model

- Approximate $\lambda_o(o|x_k)$ as a weighted Gaussian.
- The covariance matrix is part of x_k and can be estimated from data.

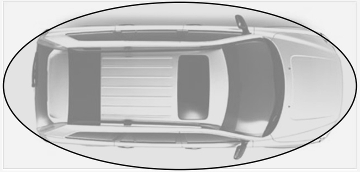


THE POISSON POINT PROCESS AND ITS INTENSITY FUNCTION

- The PPP is arguably the **standard object measurement model** for extended object tracking.
- How can we **model the intensity function** $\lambda_o(o|x_k)$?

Random matrix model

- Approximate $\lambda_o(o|x_k)$ as a weighted Gaussian.
- The covariance matrix is part of x_k and can be estimated from data.



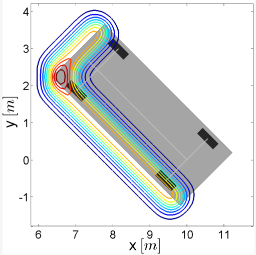
- **Note:** measurements are often described in a different coordinate system.

THE POISSON POINT PROCESS AND ITS INTENSITY FUNCTION

- The PPP is arguably the **standard object measurement model** for extended object tracking.
- How can we **model the intensity function** $\lambda_o(o|x_k)$?

Rectangular shapes

- Intensity is high along visible edges of object.
- Object dimensions may be part of x_k.
- Can be generalised to, e.g., 3D boxes.



FLEXIBLE PARAMETRISATIONS OF THE INTENSITY FUNCTION

- A flexible parametrisation of $\lambda_o(o \middle| x_k)$ may yield a more accurate model.

FLEXIBLE PARAMETRISATIONS OF THE INTENSITY FUNCTION

- A flexible parametrisation of $\lambda_o(o|x_k)$ may yield a more accurate model.

Star-convex contour

- Object contour is determined by $r(\phi)$.

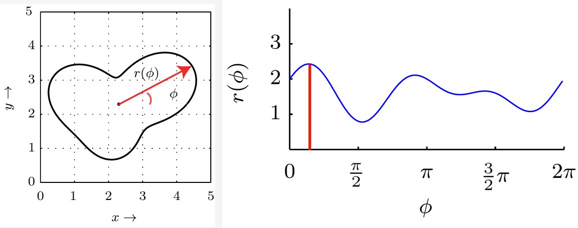


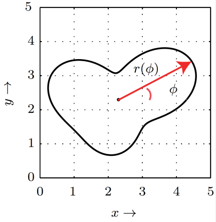
Image from Granström et al (2017), “Extended object tracking: Introduction, overview and applications”, *Journal of Advances in Information Fusion*.

FLEXIBLE PARAMETRISATIONS OF THE INTENSITY FUNCTION

- A flexible parametrisation of $\lambda_o(o|x_k)$ may yield a more accurate model.

Star-convex contour

- Object contour is determined by $r(\phi)$.
- Here, $r(\phi)$ could be, e.g., a Fourier series.



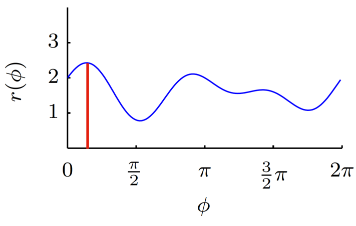


Image from Granström et al (2017), "Extended object tracking: Introduction, overview and applications", *Journal of Advances in Information Fusion*.

Extended object tracking algorithms and conjugate priors

Multi-Object Tracking

Lennart Svensson

EXTENDED OBJECT TRACKING ALGORITHMS

- Fairly rich literature on extended object tracking (EOT) algorithms, e.g.,
  - joint probabilistic data associations (JPDA),
  - particle filters,
  - probability hypothesis density (PHD) filter,
  - cardinalized probability hypothesis density (CPHD) filter,
  - probabilistic multi-hypothesis tracking (PMHT),
  - delta generalized labelled multi-Bernoulli (δ-GLMB) filters,
  - Poisson multi-Bernoulli mixture (PMBM) filters.

EXTENDED OBJECT TRACKING ALGORITHMS

- Fairly rich literature on extended object tracking (EOT) algorithms, e.g.,
  - joint probabilistic data associations (JPDA),
  - particle filters,
  - probability hypothesis density (PHD) filter,
  - cardinalized probability hypothesis density (CPHD) filter,
  - probabilistic multi-hypothesis tracking (PMHT),
  - delta generalized labelled multi-Bernoulli (δ-GLMB) filters,
  - Poisson multi-Bernoulli mixture (PMBM) filters.

- **Key difference:** the family of distributions used to approximate $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ or $p(X_k | Z_{1:k})$.

CONJUGATE PRIORS FOR EXTENDED OBJECT TRACKING (EOT)

PMBM conjugate prior

- The pdf $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$ is a conjugate prior to the standard models for EOT (Poisson birth, PPP object measurements):

Prediction: $$\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) \mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1}) \, \delta \mathbf{x}_{k-1}$$

Update: $$\mathcal{PMBM}_{k|k}(\mathbf{x}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) \mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)}{\int p(\mathbf{z}_k | \mathbf{x}'_k) \mathcal{PMBM}_{k|k-1}(\mathbf{x}'_k) \, \delta \mathbf{x}'_k}.$$

CONJUGATE PRIORS FOR EXTENDED OBJECT TRACKING (EOT)

PMBM conjugate prior

- The pdf $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$ is a conjugate prior to the standard models for EOT (Poisson birth, PPP object measurements):

Prediction: $$\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) \mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1}) \, \delta \mathbf{x}_{k-1}$$

Update: $$\mathcal{PMBM}_{k|k}(\mathbf{x}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) \mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)}{\int p(\mathbf{z}_k | \mathbf{x}'_k) \mathcal{PMBM}_{k|k-1}(\mathbf{x}'_k) \, \delta \mathbf{x}'_k}.$$

- The MBM and delta-GLMB distributions are other EOT conjugate priors.

CONJUGATE PRIORS FOR EXTENDED OBJECT TRACKING (EOT)

PMBM conjugate prior

- The pdf $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$ is a conjugate prior to the standard models for EOT (Poisson birth, PPP object measurements):

$$\text{Prediction:} \quad \mathcal{PMBM}_{k|k-1}(\mathbf{x}_k) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})\,\delta\mathbf{x}_{k-1}$$

$$\text{Update:} \quad \mathcal{PMBM}_{k|k}(\mathbf{x}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)}{\int p(\mathbf{z}_k|\mathbf{x}'_k)\mathcal{PMBM}_{k|k-1}(\mathbf{x}'_k)\,\delta\mathbf{x}'_k}.$$

- The MBM and delta-GLMB distributions are other EOT conjugate priors.
- Why study the PMBM conjugate prior?

CONJUGATE PRIORS FOR EXTENDED OBJECT TRACKING (EOT)

PMBM conjugate prior

- The pdf $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$ is a conjugate prior to the standard models for EOT (Poisson birth, PPP object measurements):

Prediction: $$\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) \mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1}) \, \delta \mathbf{x}_{k-1}$$

Update: $$\mathcal{PMBM}_{k|k}(\mathbf{x}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) \mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)}{\int p(\mathbf{z}_k | \mathbf{x}'_k) \mathcal{PMBM}_{k|k-1}(\mathbf{x}'_k) \, \delta \mathbf{x}'_k}.$$

- The MBM and delta-GLMB distributions are other EOT conjugate priors.
- Why study the PMBM conjugate prior?
  1) Describes the exact posterior: useful to understand EOT.

CONJUGATE PRIORS FOR EXTENDED OBJECT TRACKING (EOT)

PMBM conjugate prior

- The pdf $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$ is a conjugate prior to the standard models for EOT (Poisson birth, PPP object measurements):

Prediction: $$\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})\,\delta\mathbf{x}_{k-1}$$

Update: $$\mathcal{PMBM}_{k|k}(\mathbf{x}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)}{\int p(\mathbf{z}_k|\mathbf{x}'_k)\mathcal{PMBM}_{k|k-1}(\mathbf{x}'_k)\,\delta\mathbf{x}'_k}.$$

- The MBM and delta-GLMB distributions are other EOT conjugate priors.
- Why study the PMBM conjugate prior?
  1) Describes the exact posterior: useful to understand EOT.
  2) PMBM filters arguably yield state of the art performance.

PMBM CONJUGATE PRIORS

- According to $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, $\mathbf{x}_k \big| \mathbf{z}_k$ is the union of two independent sets:

PMBM CONJUGATE PRIORS

- According to $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, $\mathbf{x}_k \big| \mathbf{z}_k$ is the union of two independent sets:
  - A **PPP**: set of objects that remain undetected.

PMBM CONJUGATE PRIORS

- According to $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, $\mathbf{x}_k \big| \mathbf{z}_k$ is the union of two independent sets:
  - A **PPP**: set of objects that remain undetected.
  - A **multi-Bernoulli mixture RFS**: set of detected objects.

PMBM CONJUGATE PRIORS

- According to $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, $\mathbf{x}_k \big| \mathbf{z}_k$ is the union of two independent sets:
  - A **PPP**: set of objects that remain undetected.
  - A **multi-Bernoulli mixture RFS**: set of detected objects.
    One term in the mixture **for every global data association hypothesis**!

GLOBAL HYPOTHESES IN PMBM

- We obtain **one global hypothesis for every possible partition** of $\mathbf{z}_{1:k}$.

GLOBAL HYPOTHESES IN PMBM

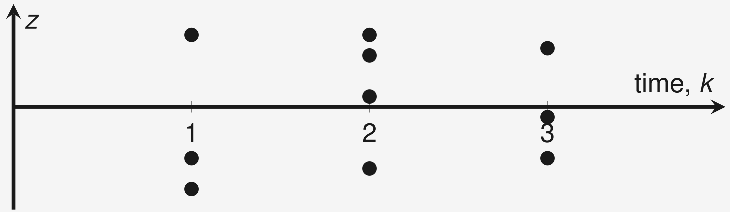
- We obtain **one global hypothesis for every possible partition** of $\mathbf{z}_{1:k}$.
- A partition is a separation of $\mathbf{z}_{1:k}$ into disjoint subsets.

GLOBAL HYPOTHESES IN PMBM

- We obtain **one global hypothesis for every possible partition** of $\mathbf{z}_{1:k}$.
- A partition is a separation of $\mathbf{z}_{1:k}$ into disjoint subsets.

Visualizing global hypotheses, given $\mathbf{z}_{1:3}$

- Here, $\mathbf{z}_{1:3}$ contains ten measurements in total.

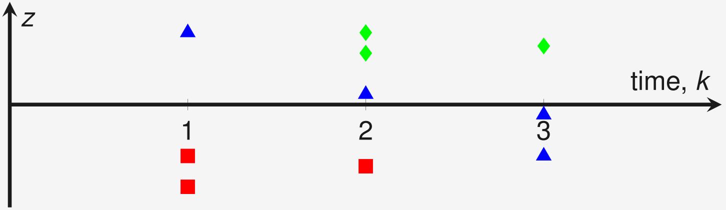


GLOBAL HYPOTHESES IN PMBM

- We obtain **one global hypothesis for every possible partition** of $\mathbf{z}_{1:k}$.
- A partition is a separation of $\mathbf{z}_{1:k}$ into disjoint subsets.

Visualizing global hypotheses, given $\mathbf{z}_{1:3}$

- Here, $\mathbf{z}_{1:3}$ contains ten measurements in total.
- In the MBM, every partition corresponds to a MB RFS and the subsets to B RFSs.

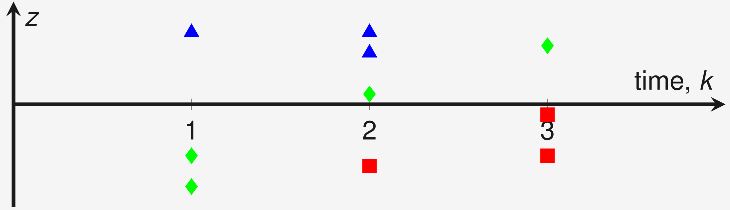


GLOBAL HYPOTHESES IN PMBM

- We obtain **one global hypothesis for every possible partition** of $\mathbf{z}_{1:k}$.
- A partition is a separation of $\mathbf{z}_{1:k}$ into disjoint subsets.

Visualizing global hypotheses, given $\mathbf{z}_{1:3}$

- Here, $\mathbf{z}_{1:3}$ contains ten measurements in total.
- In the MBM, every partition corresponds to a MB RFS and the subsets to B RFSs.

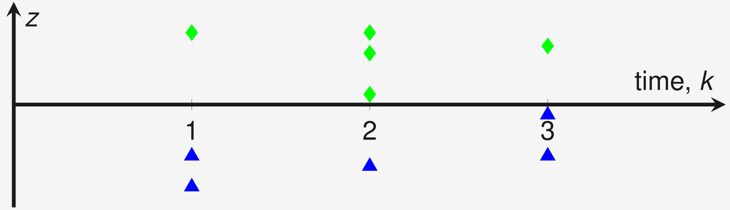


GLOBAL HYPOTHESES IN PMBM

- We obtain **one global hypothesis for every possible partition** of $\mathbf{z}_{1:k}$.
- A partition is a separation of $\mathbf{z}_{1:k}$ into disjoint subsets.

Visualizing global hypotheses, given $\mathbf{z}_{1:3}$

- Here, $\mathbf{z}_{1:3}$ contains ten measurements in total.
- In the MBM, every partition corresponds to a MB RFS and the subsets to B RFSs.



PMBM recursions for EOT

Multi-Object Tracking

Lennart Svensson

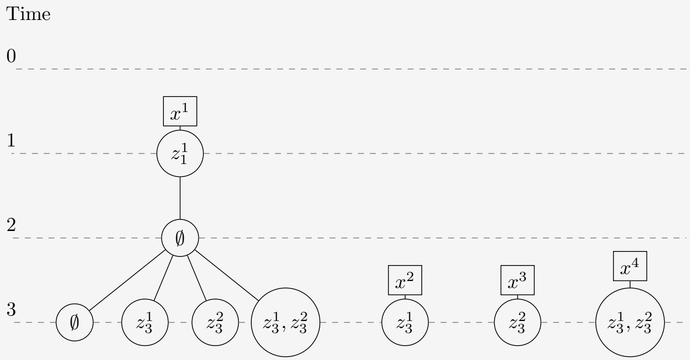
TRACK ORIENTED REPRESENTATION OF HYPOTHESES

- **Recursive algorithms:** useful to express global hypotheses as combinations of local hypotheses.

TRACK ORIENTED REPRESENTATION OF HYPOTHESES

- **Recursive algorithms:** useful to express global hypotheses as combinations of local hypotheses.

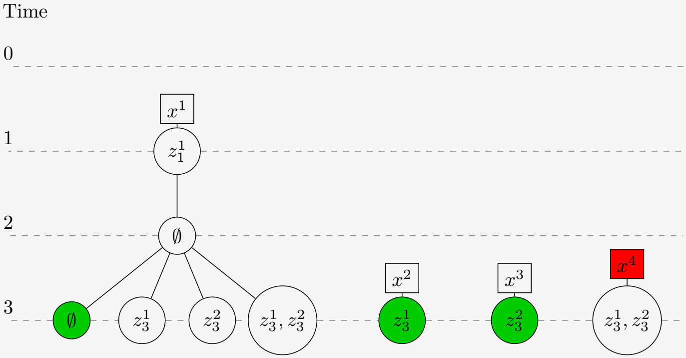
Local and global hypotheses, $\mathbf{z}_1 = \{z_1^1\}$, $\mathbf{z}_2 = \emptyset$, $\mathbf{z}_3 = \{z_3^1, z_3^2\}$



TRACK ORIENTED REPRESENTATION OF HYPOTHESES

- **Recursive algorithms:** useful to express global hypotheses as combinations of local hypotheses.

Local and global hypotheses, $\mathbf{z}_1 = \{z_1^1\}$, $\mathbf{z}_2 = \emptyset$, $\mathbf{z}_3 = \{z_3^1, z_3^2\}$



Look-up table:

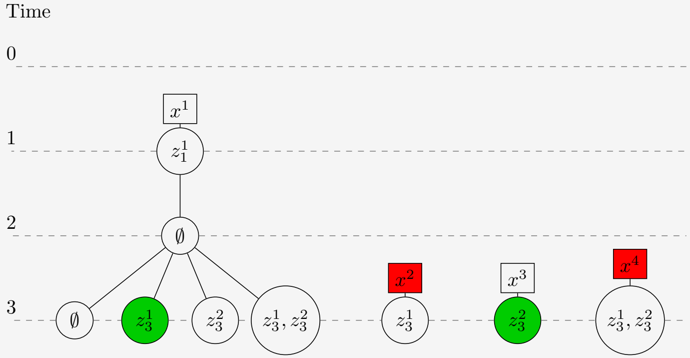
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ & & & \\ & & & \end{bmatrix}$$

Local hypotheses indexed from left to right.

TRACK ORIENTED REPRESENTATION OF HYPOTHESES

- **Recursive algorithms:** useful to express global hypotheses as combinations of local hypotheses.

Local and global hypotheses, $\mathbf{z}_1 = \{z_1^1\}$, $\mathbf{z}_2 = \emptyset$, $\mathbf{z}_3 = \{z_3^1, z_3^2\}$



Look-up table:

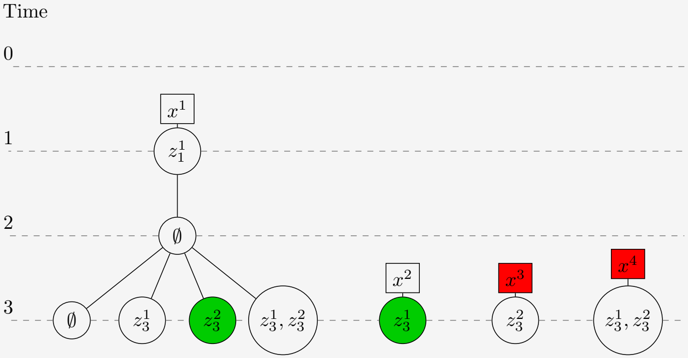
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

Local hypotheses indexed from left to right.

TRACK ORIENTED REPRESENTATION OF HYPOTHESES

- **Recursive algorithms:** useful to express global hypotheses as combinations of local hypotheses.

Local and global hypotheses, $\mathbf{z}_1 = \{z_1^1\}$, $\mathbf{z}_2 = \emptyset$, $\mathbf{z}_3 = \{z_3^1, z_3^2\}$



Look-up table:

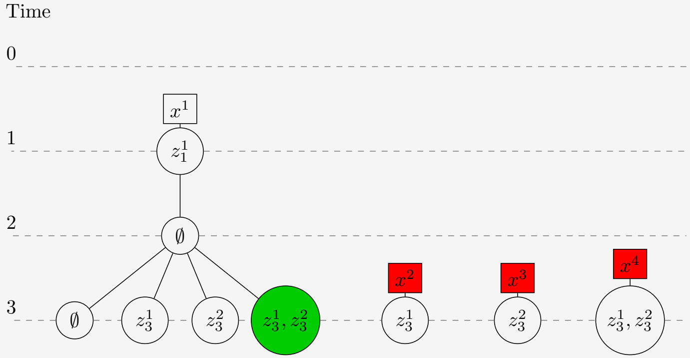
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 \\ & & & \\ & & & \end{bmatrix}$$

Local hypotheses indexed from left to right.

TRACK ORIENTED REPRESENTATION OF HYPOTHESES

- **Recursive algorithms:** useful to express global hypotheses as combinations of local hypotheses.

Local and global hypotheses, $\mathbf{z}_1 = \{z_1^1\}$, $\mathbf{z}_2 = \emptyset$, $\mathbf{z}_3 = \{z_3^1, z_3^2\}$



Look-up table:

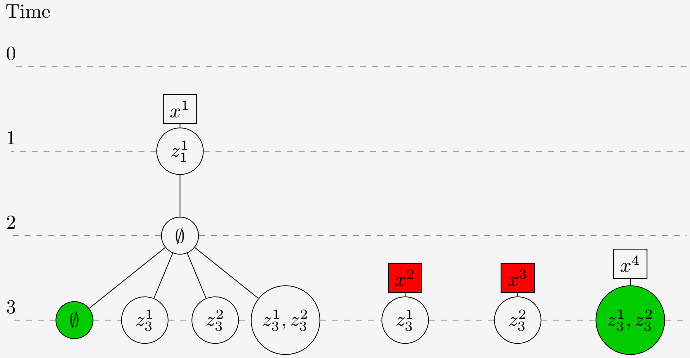
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix}$$

Local hypotheses indexed from left to right.

TRACK ORIENTED REPRESENTATION OF HYPOTHESES

- **Recursive algorithms:** useful to express global hypotheses as combinations of local hypotheses.

Local and global hypotheses, $\mathbf{z}_1 = \{z_1^1\}$, $\mathbf{z}_2 = \emptyset$, $\mathbf{z}_3 = \{z_3^1, z_3^2\}$



Look-up table:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Local hypotheses indexed from left to right.

PMBM – ALGORITHMIC OVERVIEW

- **Objective:** approximate $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, given $\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})$ and $\mathbf{z}_k$.

PMBM – ALGORITHMIC OVERVIEW

- **Objective:** approximate $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, given $\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})$ and $\mathbf{z}_k$.
- **Prediction:** same equations as for point objects.
  Every component predicted independently. Existence prob. scaled by P^{S}.
  Add birth intensity to predicted Poisson intensity.

PMBM – ALGORITHMIC OVERVIEW

- **Objective:** approximate $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, given $\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})$ and $\mathbf{z}_k$.
- **Prediction:** same equations as for point objects.
  Every component predicted independently. Existence prob. scaled by P^{S}.
  Add birth intensity to predicted Poisson intensity.
- **Update:**
  - **PPP**: scale intensity with probability that an object at x_k is undetected.

PMBM – ALGORITHMIC OVERVIEW

- **Objective:** approximate $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, given $\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})$ and $\mathbf{z}_k$.
- **Prediction:** same equations as for point objects.
  Every component predicted independently. Existence prob. scaled by P^{S}.
  Add birth intensity to predicted Poisson intensity.
- **Update:**
  - **PPP**: scale intensity with probability that an object at x_k is undetected.
  - **MBM**:

PMBM – ALGORITHMIC OVERVIEW

- **Objective:** approximate $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, given $\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})$ and $\mathbf{z}_k$.
- **Prediction:** same equations as for point objects.
  Every component predicted independently. Existence prob. scaled by P^{S}.
  Add birth intensity to predicted Poisson intensity.
- **Update:**
  - **PPP**: scale intensity with probability that an object at x_k is undetected.
  - **MBM**:
    1. Identify global hypotheses with significant weights.

PMBM – ALGORITHMIC OVERVIEW

- **Objective:** approximate $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, given $\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})$ and $\mathbf{z}_k$.
- **Prediction:** same equations as for point objects.
  Every component predicted independently. Existence prob. scaled by P^{S}.
  Add birth intensity to predicted Poisson intensity.
- **Update:**
  - **PPP**: scale intensity with probability that an object at x_k is undetected.
  - **MBM**:
    1. Identify global hypotheses with significant weights.
    2. Prune all other global hypotheses. Also prune local hypotheses that do not appear in global hyp. that we keep.

PMBM – ALGORITHMIC OVERVIEW

- **Objective:** approximate $\mathcal{PMBM}_{k|k}(\mathbf{x}_k)$, given $\mathcal{PMBM}_{k-1|k-1}(\mathbf{x}_{k-1})$ and $\mathbf{z}_k$.
- **Prediction:** same equations as for point objects.
  Every component predicted independently. Existence prob. scaled by P^{S}.
  Add birth intensity to predicted Poisson intensity.
- **Update:**
  - **PPP**: scale intensity with probability that an object at x_k is undetected.
  - **MBM**:
    1. Identify global hypotheses with significant weights.
    2. Prune all other global hypotheses. Also prune local hypotheses that do not appear in global hyp. that we keep.
  - **Note:** assignments methods (auction, Murty's, ...) assume *at most one measurement per object* and cannot be used (directly) to find probably global hypotheses.

Pruning and clustering

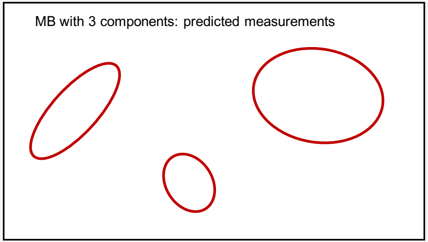
Multi-Object Tracking

Lennart Svensson

PROBLEM FORMULATION

Problem formulation

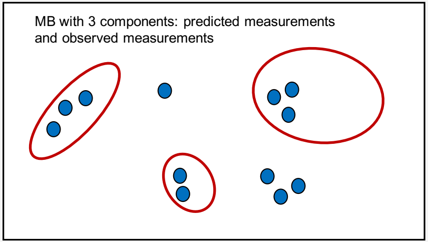
- Suppose $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ is PMB.



PROBLEM FORMULATION

Problem formulation

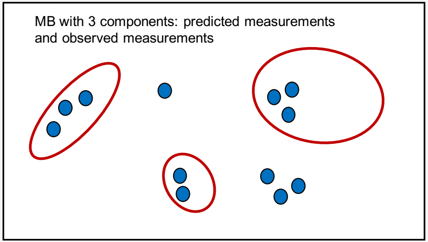
- Suppose $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ is PMB.
- Given $\mathbf{z}_k$, find most probable association hypotheses.



PROBLEM FORMULATION

Problem formulation

- Suppose $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ is PMB.
- Given $\mathbf{z}_k$, find most probable association hypotheses.

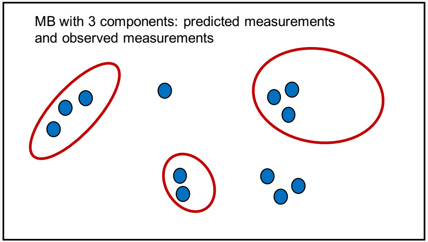


- If $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)$:

PROBLEM FORMULATION

Problem formulation

- Suppose $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ is PMB.
- Given $\mathbf{z}_k$, find most probable association hypotheses.

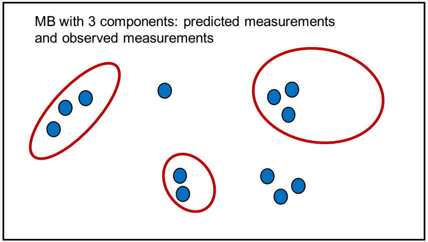


- If $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)$:
  1. Find the most probable hypotheses for every PMB in $\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)$.

PROBLEM FORMULATION

Problem formulation

- Suppose $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is PMB.
- Given $\mathbf{z}_k$, find most probable association hypotheses.



- If $p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)$:
  1. Find the most probable hypotheses for every PMB in $\mathcal{PMBM}_{k|k-1}(\mathbf{x}_k)$.
  2. Store the most probable hypotheses found in step 1.

CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?

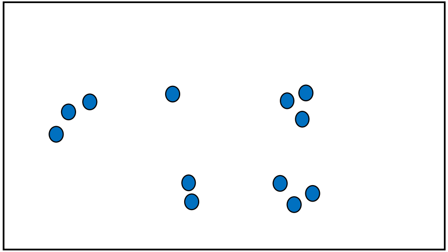
CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?
  Yes! Using the following two-step procedure:

CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?
  Yes! Using the following two-step procedure:

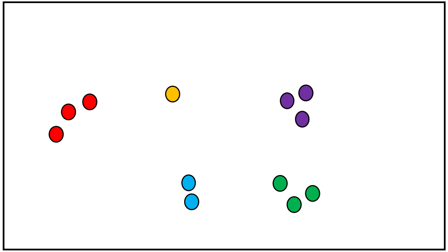
- Use **clustering:** to partition $\mathbf{z}_k$.



CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?
  Yes! Using the following two-step procedure:

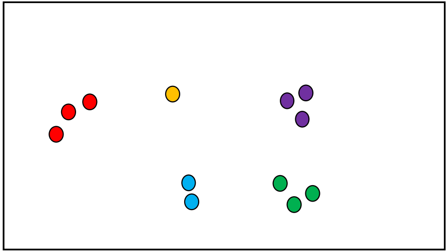
- Use **clustering:** to partition $\mathbf{z}_k$.



CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?
  Yes! Using the following two-step procedure:

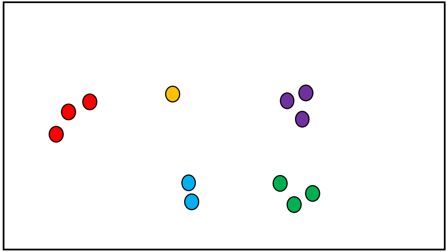
- Use **clustering:** to partition $\mathbf{z}_k$.
  Note: 1) we then associate clusters to potential objects,



CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?
  Yes! Using the following two-step procedure:

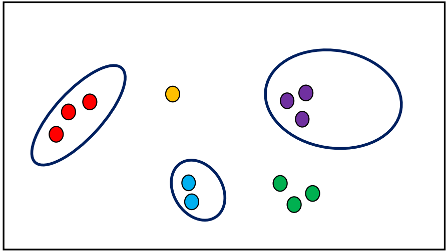
- Use **clustering:** to partition $\mathbf{z}_k$.
  Note: 1) we then associate clusters to potential objects, 2) two clusters cannot be associated to the same potential object.



CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?
  Yes! Using the following two-step procedure:

- Use **clustering:** to partition $\mathbf{z}_k$.
  Note: 1) we then associate clusters to potential objects, 2) two clusters cannot be associated to the same potential object.



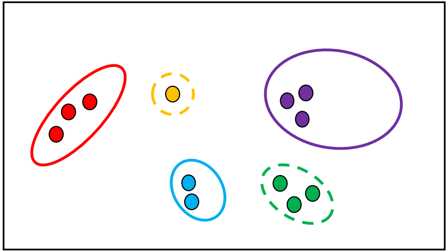
- **Assignment:** we can now use Murty's algorithm to find the K best assignments.

CLUSTERING AND ASSIGNMENT ALGORITHMS?

- Is it possible to make use of assignment algorithms?
  Yes! Using the following two-step procedure:

- Use **clustering:** to partition $\mathbf{z}_k$.
  Note: 1) we then associate clusters to potential objects, 2) two clusters cannot be associated to the same potential object.

- **Assignment:** we can now use Murty's algorithm to find the K best assignments.



CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.

CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.

CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.
- **A simple fix:**
  1. Try several different clustering algorithms.

CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.
- **A simple fix:**
  1. Try several different clustering algorithms.
  2. Apply Murty's algorithms to all.

CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.
- **A simple fix:**
  1. Try several different clustering algorithms.
  2. Apply Murty's algorithms to all.

  Increases the chances of finding good clusters!

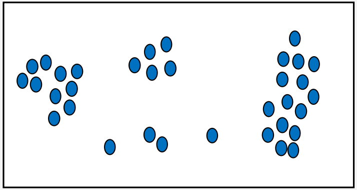
CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.
- **A simple fix:**
  1. Try several different clustering algorithms.
  2. Apply Murty's algorithms to all.

  Increases the chances of finding good clusters!

Challenging scenarios

- Clustering only considers measurements.



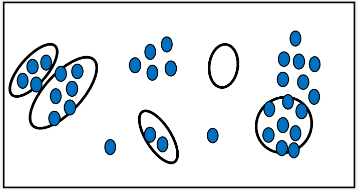
CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.
- **A simple fix:**
    1. Try several different clustering algorithms.
    2. Apply Murty's algorithms to all.

  Increases the chances of finding good clusters!

Challenging scenarios

- Clustering only considers measurements.



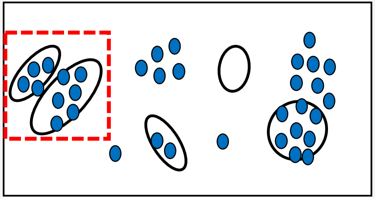
CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.
- **A simple fix:**
  1. Try several different clustering algorithms.
  2. Apply Murty's algorithms to all.

  Increases the chances of finding good clusters!

Challenging scenarios

- Clustering only considers measurements.
- Two predicted objects next to each other.



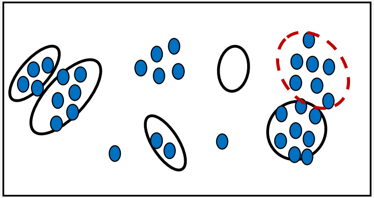
CHALLENGES WITH CLUSTERING

- **Clustering + Murty's**: fast and accurate if clustering works well.
- Sadly, clustering works poorly in many cases.
- **A simple fix:**
  1. Try several different clustering algorithms.
  2. Apply Murty's algorithms to all.

  Increases the chances of finding good clusters!

Challenging scenarios

- Clustering only considers measurements.
- Two predicted objects next to each other.
- Cluster from new object next to predicted object.



SAMPLING METHODS

Sampling methods to find likely associations

- Sample data associations according to their probabilities (weights).

SAMPLING METHODS

Sampling methods to find likely associations

- Sample data associations according to their probabilities (weights).
- Standard approach: run a Markov chain that modifies the associations in each step.
  $\rightsquigarrow$ Finds associations with large weights!

SAMPLING METHODS

Sampling methods to find likely associations

- Sample data associations according to their probabilities (weights).
- Standard approach: run a Markov chain that modifies the associations in each step.
  $\rightsquigarrow$ Finds associations with large weights!
- Yields very good performance.

SAMPLING METHODS

Sampling methods to find likely associations

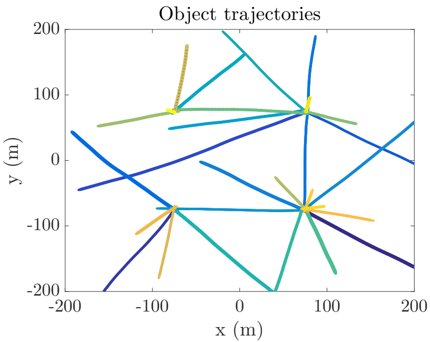
- Sample data associations according to their probabilities (weights).
- Standard approach: run a Markov chain that modifies the associations in each step.
  $\rightsquigarrow$ Finds associations with large weights!
- Yields very good performance.
- Difficult to analyse convergence.

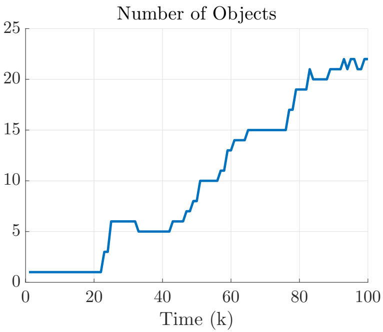
Simulation example

Multi-Object Tracking

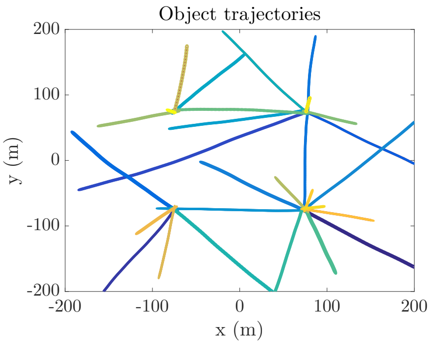
Lennart Svensson

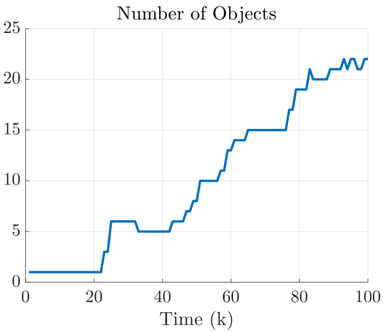
SIMULATION SCENARIO





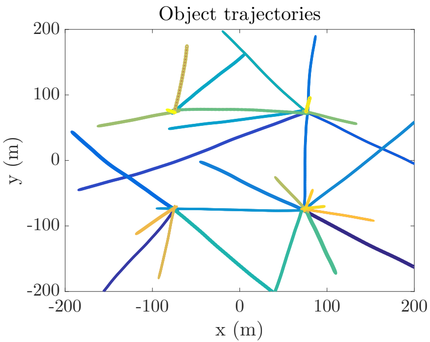
SIMULATION SCENARIO

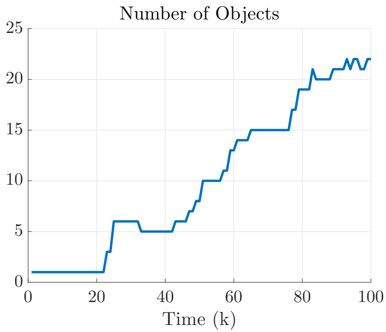




- **Measurement model:**
  - Single object: PPP, random matrix, Poisson rate ≈ 8, $P^{\mathrm{D}} = 0.9$.
  - Clutter: Poisson rate 60.

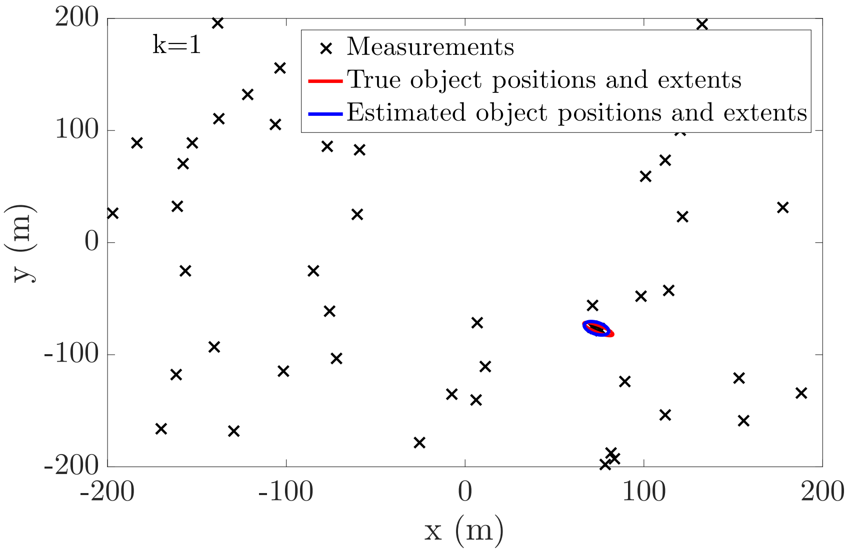
SIMULATION SCENARIO





- **Measurement model:**
  - Single object: PPP, random matrix, Poisson rate ≈ 8, $P^{\mathrm{D}} = 0.9$.
  - Clutter: Poisson rate 60.
- **Motion model:**
  - GGIW, constant velocity, $P^{\mathrm{S}} = 0.99$.
  - Birth intensity: peaks around $(\pm 75, \pm 75)$.

SIMULATION RESULTS



Section 2:
Sets of trajectories

Multi-Object Tracking

Lennart Svensson

Sets of trajectories – motivation

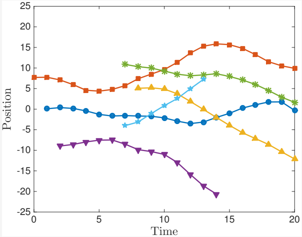
Multi-Object Tracking

Lennart Svensson

POSSIBLE PROBLEM FORMULATIONS (1)

The set of all trajectories

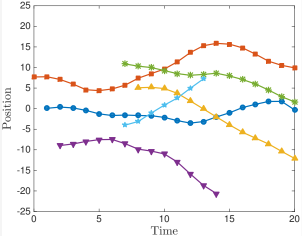
- Estimate trajectories of all objects present at some point in time ($\leq k$).



POSSIBLE PROBLEM FORMULATIONS (1)

The set of all trajectories

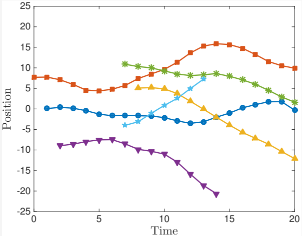
- Estimate trajectories of all objects present at some point in time ($\leq k$).
- **Example:** to collect ground truth data for self-driving vehicles.



POSSIBLE PROBLEM FORMULATIONS (1)

The set of all trajectories

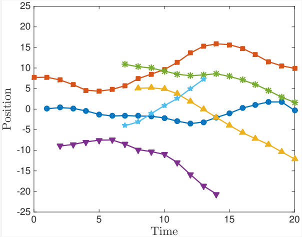
- Estimate trajectories of all objects present at some point in time ($\leq k$).
- **Example:** to collect ground truth data for self-driving vehicles. For instance, to train machine learning algorithms.
- **More examples:** to track human cells, sport athletes, etc.



POSSIBLE PROBLEM FORMULATIONS (1)

The set of all trajectories

- Estimate trajectories of all objects present at some point in time ($\leq k$).
- **Example:** to collect ground truth data for self-driving vehicles. For instance, to train machine learning algorithms.
- **More examples:** to track human cells, sport athletes, etc.

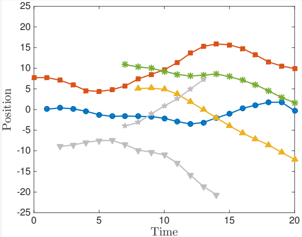


- We focus on recursive estimation algorithms, but batch solutions are also important.

POSSIBLE PROBLEM FORMULATIONS (2)

The set of current trajectories

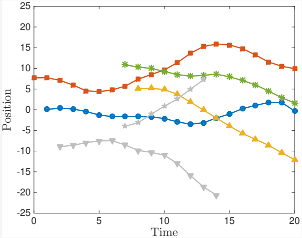
- Estimate trajectories of all currently present objects.



POSSIBLE PROBLEM FORMULATIONS (2)

The set of current trajectories

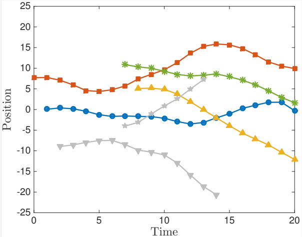
- Estimate trajectories of all currently present objects.
- **Example:** for surveillance systems.



POSSIBLE PROBLEM FORMULATIONS (2)

The set of current trajectories

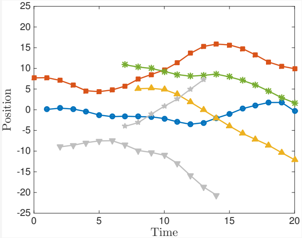
- Estimate trajectories of all currently present objects.
- **Example:** for surveillance systems. For instance, of ships at sea, people at airports, etc.



POSSIBLE PROBLEM FORMULATIONS (2)

The set of current trajectories

- Estimate trajectories of all currently present objects.
- **Example:** for surveillance systems. For instance, of ships at sea, people at airports, etc.

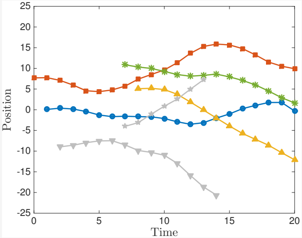


- **Note:** from now on, we refer to recursively computing $p(\mathbf{x}_k | \mathbf{z}_{1:k})$, as multi-object filtering.

POSSIBLE PROBLEM FORMULATIONS (2)

The set of current trajectories

- Estimate trajectories of all currently present objects.
- **Example:** for surveillance systems. For instance, of ships at sea, people at airports, etc.



- **Note:** from now on, we refer to recursively computing $p(\mathbf{x}_k | \mathbf{z}_{1:k})$, as multi-object filtering.
- In upcoming videos, **multi-object tracking** refers to recursively estimating the set of trajectories.

SETS OF TRAJECTORIES – BASIC IDEA

Basic idea

- We want to find the set of trajectories.

SETS OF TRAJECTORIES – BASIC IDEA

Basic idea

- We want to find the set of trajectories.
- Bayesian approach: compute posterior over quantities of interest.

SETS OF TRAJECTORIES – BASIC IDEA

Basic idea

- We want to find the set of trajectories.
- Bayesian approach: compute posterior over quantities of interest.
- **Basic idea:** use the **set of trajectories as state variable**.

SINGLE TRAJECTORY: PARAMETRISATION

Single trajectory parametrisation

- We write single trajectories as

$$X_k = (\beta, \epsilon, x_{\beta:\epsilon})$$

where β is the time of birth, ϵ is the time of trajectory's most recent state and $x_{\beta:\epsilon}$ is the sequence of states.

SINGLE TRAJECTORY: PARAMETRISATION

Single trajectory parametrisation

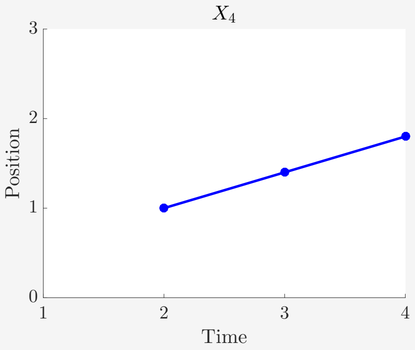
- We write single trajectories as

$$X_k = (\beta, \epsilon, x_{\beta:\epsilon})$$

where β is the time of birth, ϵ is the time of trajectory's most recent state and $x_{\beta:\epsilon}$ is the sequence of states.

Example

- The trajectory $X_4 = (2, 4, (1, 1.4, 1.8))$:



SINGLE TRAJECTORY: PARAMETRISATION

Single trajectory parametrisation

- We write single trajectories as

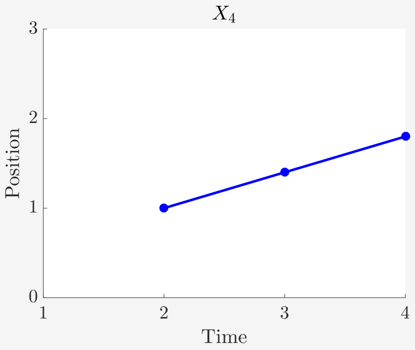
$$X_k = (\beta, \epsilon, x_{\beta:\epsilon})$$

  where β is the time of birth, ϵ is the time of trajectory's most recent state and $x_{\beta:\epsilon}$ is the sequence of states.

- **Note:** $\epsilon = k$ means that trajectory is ongoing, whereas $\epsilon < k$ means that it ended at time ϵ.

Example

- The trajectory $X_4 = (2, 4, (1, 1.4, 1.8))$:



SETS OF TRAJECTORIES: PARAMETRISATION

Sets of trajectories

- We use

$$\mathbf{X}_k = \left\{X_k^1, \ldots, X_k^{N_k}\right\}$$

to denote the set of trajectories at time k.

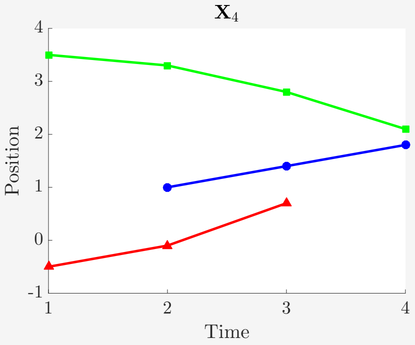
SETS OF TRAJECTORIES: PARAMETRISATION

Sets of trajectories

- We use
$$\mathbf{X}_k = \left\{X_k^1, \ldots, X_k^{N_k}\right\}$$
to denote the set of trajectories at time k.

Example

- A set $\mathbf{X}_4 = \{X_4^1, X_4^2, X_4^3\}$:



SETS OF TRAJECTORIES: PARAMETRISATION

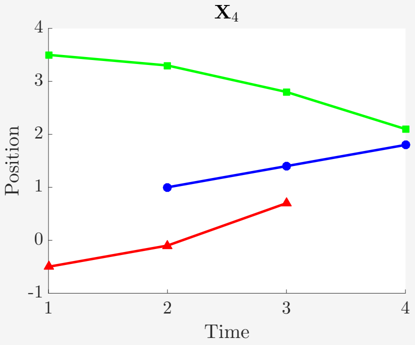
Sets of trajectories

- We use
$$\mathbf{X}_k = \left\{X_k^1, \ldots, X_k^{N_k}\right\}$$
to denote the set of trajectories at time k.

- The number of trajectories N_k may be larger than the number of objects at time k, n_k.

Example

- A set $\mathbf{X}_4 = \{X_4^1, X_4^2, X_4^3\}$:



SETS OF TRAJECTORIES: PARAMETRISATION

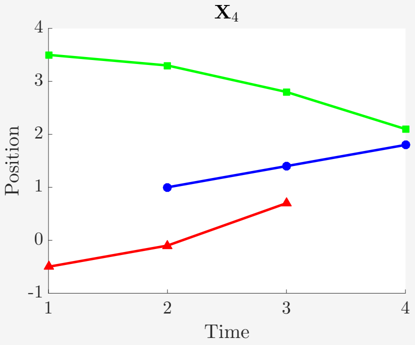
Sets of trajectories

- We use
$$\mathbf{X}_k = \left\{ X_k^1, \ldots, X_k^{N_k} \right\}$$
to denote the set of trajectories at time k.

- The number of trajectories N_k may be larger than the number of objects at time k, n_k.
- The set $\mathbf{X}_k$ is a random finite set.

Example

- A set $\mathbf{X}_4 = \{X_4^1, X_4^2, X_4^3\}$:



WHY USE X_K AS STATE VARIABLE?

WHY USE X_K AS STATE VARIABLE?

- The set of trajectories is the quantity of interest.

WHY USE $\mathbf{X}_K$ AS STATE VARIABLE?

- The set of trajectories is the quantity of interest.
- We now obtain the posterior of $\mathbf{X}_k$ using filtering.

WHY USE $\mathbf{X}_K$ AS STATE VARIABLE?

- The set of trajectories is the quantity of interest.
- We now obtain the posterior of $\mathbf{X}_k$ using filtering.
- There is a one-to-one mapping between $\mathbf{X}_k$ and the physical reality:
  $\mathbf{X}_k$ is a minimal representation of the quantity of interest!

WHY USE $\mathbf{X}_K$ AS STATE VARIABLE?

- The set of trajectories is the quantity of interest.
- We now obtain the posterior of $\mathbf{X}_k$ using filtering.
- There is a one-to-one mapping between $\mathbf{X}_k$ and the physical reality: $\mathbf{X}_k$ is a minimal representation of the quantity of interest!
- Sets of trajectories can be used to develop performance metrics.

WHY USE $\mathbf{X}_K$ AS STATE VARIABLE?

- The set of trajectories is the quantity of interest.
- We now obtain the posterior of $\mathbf{X}_k$ using filtering.
- There is a one-to-one mapping between $\mathbf{X}_k$ and the physical reality:
  $\mathbf{X}_k$ is a minimal representation of the quantity of interest!
- Sets of trajectories can be used to develop performance metrics.
- Have been used to develop efficient and accurate tracking algorithms for both point objects and extended objects.

Sets of trajectories – basic concepts

Multi-Object Tracking

Lennart Svensson

Single trajectory integral

- Suppose we consider trajectories up to time step t. The single trajectory integral is then defined as
$$\int f(X)\,\mathrm{d}X = \sum_{(\beta,\epsilon)\in\mathbf{i}_t} \int f\left((\beta, \epsilon, x_{\beta:\epsilon})\right)\,\mathrm{d}x_{\beta:\epsilon}$$
where $\mathbf{i}_t = \{(\beta, \epsilon) : 1 \leq \beta \leq \epsilon \leq t\}$.

INTEGRATION

Single trajectory integral

- Suppose we consider trajectories up to time step t. The single trajectory integral is then defined as
$$\int f(X)\,\mathrm{d}X = \sum_{(\beta,\epsilon)\in\mathbf{i}_t} \int f\left((\beta,\epsilon,x_{\beta:\epsilon})\right)\,\mathrm{d}x_{\beta:\epsilon}$$
where $\mathbf{i}_t = \{(\beta,\epsilon) : 1 \leq \beta \leq \epsilon \leq t\}$.

Set integrals

- The set integral is defined as
$$\int f(\mathbf{X})\,\delta\mathbf{X} = \sum_{i=0}^{\infty} \frac{1}{i!} \int f(\{X^1,\dots,X^i\})\,\mathrm{d}X^{1:i}.$$

BAYESIAN FILTERING RECURSION FOR MOT

- Our variables
  - the set of trajectories, $\mathbf{X}_k$,
  - the set of measurement vectors, $\mathbf{z}_k$,

  are random finite sets (RFSs).

BAYESIAN FILTERING RECURSION FOR MOT

- Our variables
  - the set of trajectories, $\mathbf{X}_k$,
  - the set of measurement vectors, $\mathbf{z}_k$,

  are random finite sets (RFSs).

Bayesian filtering recursions

$$\text{Prediction:} \quad p(\mathbf{X}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{X}_k | \mathbf{X}_{k-1}) p(\mathbf{X}_{k-1} | \mathbf{z}_{1:k-1}) \, \delta \mathbf{X}_{k-1}$$

$$\text{Update:} \quad p(\mathbf{X}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{X}_k) p(\mathbf{X}_k | \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k | \mathbf{X}'_k) p(\mathbf{X}'_k | \mathbf{z}_{1:k-1}) \, \delta \mathbf{X}'_k}.$$

- Standard equations: Chapman-Kolmogorov for prediction and Bayes' rule for update.

MODELS

- Our **models** of the world **have not changed**.

MODELS

- Our **models** of the world **have not changed**.

Measurement model

- Measurements $\mathbf{z}_k$ only depend on $\mathbf{x}_k$

$$p(\mathbf{z}_k | \mathbf{X}_k) = p(\mathbf{z}_k | \mathbf{x}_k = \tau_k(\mathbf{X}_k)),$$

where τ_k extracts $\mathbf{x}_k$ from $\mathbf{X}_k$.

MODELS

- Our **models** of the world **have not changed**.

Measurement model

- Measurements $\mathbf{z}_k$ only depend on $\mathbf{x}_k$

$$p(\mathbf{z}_k | \mathbf{X}_k) = p(\mathbf{z}_k | \mathbf{x}_k = \tau_k(\mathbf{X}_k)),$$

where τ_k extracts $\mathbf{x}_k$ from $\mathbf{X}_k$.

Motion model

- Objects appear, move and disappear.

MODELS

- Our **models** of the world **have not changed**.

Measurement model

- Measurements $\mathbf{z}_k$ only depend on $\mathbf{x}_k$

$$p(\mathbf{z}_k | \mathbf{X}_k) = p(\mathbf{z}_k | \mathbf{x}_k = \tau_k(\mathbf{X}_k)),$$

where τ_k extracts $\mathbf{x}_k$ from $\mathbf{X}_k$.

Motion model

- Objects appear, move and disappear. Trajectories at time k:
  - Appearing objects: $X_k = (k, k, x_k)$.

MODELS

- Our **models** of the world **have not changed**.

Measurement model

- Measurements $\mathbf{z}_k$ only depend on $\mathbf{x}_k$

$$p(\mathbf{z}_k | \mathbf{X}_k) = p(\mathbf{z}_k | \mathbf{x}_k = \tau_k(\mathbf{X}_k)),$$

where τ_k extracts $\mathbf{x}_k$ from $\mathbf{X}_k$.

Motion model

- Objects appear, move and disappear. Trajectories at time k:
  - Appearing objects: $X_k = (k, k, x_k)$.
  - Surviving objects: $X_{k-1} = (\beta, k-1, x_{\beta:k-1}) \to X_k = (\beta, k, x_{\beta:k})$.

MODELS

- Our **models** of the world **have not changed**.

Measurement model

- Measurements $\mathbf{z}_k$ only depend on $\mathbf{x}_k$

$$p(\mathbf{z}_k | \mathbf{X}_k) = p(\mathbf{z}_k | \mathbf{x}_k = \tau_k(\mathbf{X}_k)),$$

where τ_k extracts $\mathbf{x}_k$ from $\mathbf{X}_k$.

Motion model

- Objects appear, move and disappear. Trajectories at time k:
  - Appearing objects: $X_k = (k, k, x_k)$.
  - Surviving objects: $X_{k-1} = (\beta, k-1, x_{\beta:k-1}) \to X_k = (\beta, k, x_{\beta:k})$.
  - Disappearing objects, two cases:
    1. Tracking all objects that have been present: keep X_{k-1} in $\mathbf{X}_k, \mathbf{X}_{k+1}, \ldots$

MODELS

- Our **models** of the world **have not changed**.

Measurement model

- Measurements $\mathbf{z}_k$ only depend on $\mathbf{x}_k$

$$p(\mathbf{z}_k | \mathbf{X}_k) = p(\mathbf{z}_k | \mathbf{x}_k = \tau_k(\mathbf{X}_k)),$$

where τ_k extracts $\mathbf{x}_k$ from $\mathbf{X}_k$.

Motion model

- Objects appear, move and disappear. Trajectories at time k:
  - Appearing objects: $X_k = (k, k, x_k)$.
  - Surviving objects: $X_{k-1} = (\beta, k-1, x_{\beta:k-1}) \to X_k = (\beta, k, x_{\beta:k})$.
  - Disappearing objects, two cases:
    1. Tracking all objects that have been present: keep X_{k-1} in $\mathbf{X}_k, \mathbf{X}_{k+1}, \ldots$
    2. Tracking objects present at time k: remove X_{k-1} from set of trajectories.

COMMON RFSs: BERNOULLI

Bernoulli RFSs

- A Bernoulli RFS $\mathbf{X}$ has the multitrajectory pdf

$$p(\mathbf{X}) = \begin{cases} 1 - r & \text{if } \mathbf{X} = \emptyset \\ r\, p_X(X) & \text{if } \mathbf{X} = \{X\} \\ 0 & \text{if } |\mathbf{X}| > 1, \end{cases}$$

where $0 \leq r \leq 1$ and $p_X(X)$ is a single trajectory density.

COMMON RFSs: BERNOULLI

Bernoulli RFSs

- A Bernoulli RFS $\mathbf{X}$ has the multitrajectory pdf

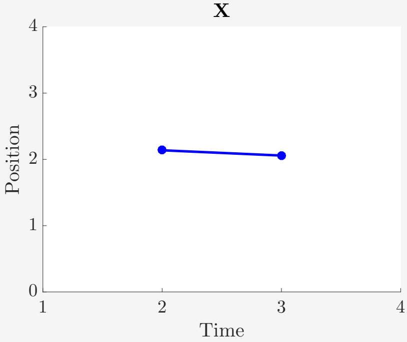
$$p(\mathbf{X}) = \begin{cases} 1 - r & \text{if } \mathbf{X} = \emptyset \\ r\, p_X(X) & \text{if } \mathbf{X} = \{X\} \\ 0 & \text{if } |\mathbf{X}| > 1, \end{cases}$$

where $0 \leq r \leq 1$ and $p_X(X)$ is a single trajectory density.

Sampling a Bernoulli RFS

- Suppose $r = 0.8$ and

$$p_X((\beta, \epsilon, x_{\beta:\epsilon})) = \delta_{\beta-2}\delta_{\epsilon-3} \times \times \mathcal{N}(x_2; 2, 0.1)\,\mathcal{N}(x_3; 2.5, 0.3).$$



COMMON RFSs: BERNOULLI

Bernoulli RFSs

- A Bernoulli RFS $\mathbf{X}$ has the multitrajectory pdf

$$p(\mathbf{X}) = \begin{cases} 1 - r & \text{if } \mathbf{X} = \emptyset \\ r\, p_X(X) & \text{if } \mathbf{X} = \{X\} \\ 0 & \text{if } |\mathbf{X}| > 1, \end{cases}$$

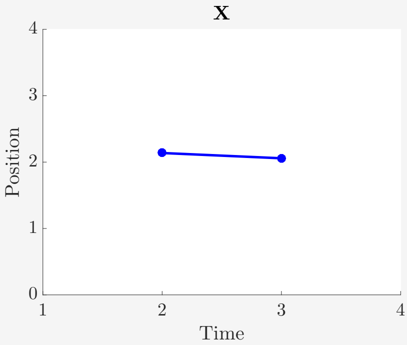
where $0 \leq r \leq 1$ and $p_X(X)$ is a single trajectory density.

- **Note:** r and $p_X(X)$ jointly determine existence probability at a specific time step t.

Sampling a Bernoulli RFS

- Suppose $r = 0.8$ and

$$p_X((\beta, \epsilon, x_{\beta:\epsilon})) = \delta_{\beta-2}\delta_{\epsilon-3} \times$$
$$\times \mathcal{N}(x_2; 2, 0.1)\, \mathcal{N}(x_3; 2.5, 0.3).$$



COMMON RFSs: PPPs

Poisson RFSs

- A Poisson RFS $\mathbf{X}$, with intensity function $\lambda(X)$, has the multitrajectory pdf

$$p(\mathbf{X}) = \exp(-\bar{\lambda}) \prod_{X \in \mathbf{X}} \lambda(X),$$

where $\bar{\lambda} = \int \lambda(X) \, \mathrm{d}X$ is the Poisson rate.

COMMON RFSs: PPPs

Poisson RFSs

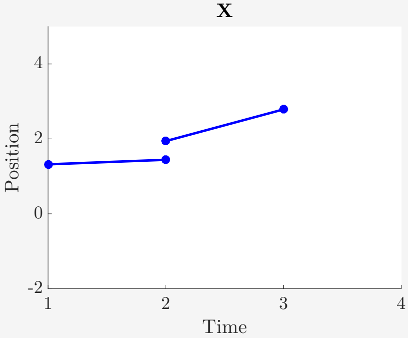
- A Poisson RFS $\mathbf{X}$, with intensity function $\lambda(X)$, has the multitrajectory pdf

$$p(\mathbf{X}) = \exp(-\bar{\lambda}) \prod_{X \in \mathbf{X}} \lambda(X),$$

where $\bar{\lambda} = \int \lambda(X)\,\mathrm{d}X$ is the Poisson rate.

Sampling a Poisson RFS

- Suppose $\lambda((\beta, \epsilon, x_{\beta:\epsilon})) = \delta_{\beta-1}\delta_{\epsilon-2}\mathcal{N}(x_1; 1, 1)\mathcal{N}(x_2; 1, 1) + 0.7\delta_{\beta-2}\delta_{\epsilon-3}\mathcal{N}(x_2; 2, 1)\mathcal{N}(x_3; 2, 1)$



COMMON RFSs: PPPs

Poisson RFSs

- A Poisson RFS $\mathbf{X}$, with intensity function $\lambda(X)$, has the multitrajectory pdf

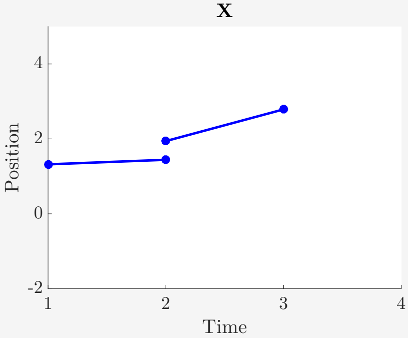
$$p(\mathbf{X}) = \exp(-\bar{\lambda}) \prod_{X \in \mathbf{X}} \lambda(X),$$

where $\bar{\lambda} = \int \lambda(X) \, \mathrm{d}X$ is the Poisson rate.

- **Note:** $|\mathbf{X}| \sim \mathrm{Po}(\bar{\lambda})$ and $\lambda(X)/\bar{\lambda}$ is the "spatial" distribution (including β and ϵ).

Sampling a Poisson RFS

- Suppose $\lambda((\beta, \epsilon, x_{\beta:\epsilon})) = \delta_{\beta-1}\delta_{\epsilon-2}\mathcal{N}(x_1; 1, 1)\mathcal{N}(x_2; 1, 1) + 0.7\delta_{\beta-2}\delta_{\epsilon-3}\mathcal{N}(x_2; 2, 1)\mathcal{N}(x_3; 2, 1)$



PMBM trackers – part 1

Multi-Object Tracking

Lennart Svensson

SET OF TRAJECTORY CONJUGACY

PMBM conjugate prior

- The Poisson Multi-Bernoulli Mixture (PMBM) multitrajectory pdf $\mathcal{PMBM}_{k|k}(\mathbf{X}_k)$ is a conjugate prior to the standard models (point objects, Poisson birth):

Prediction: $$\mathcal{PMBM}_{k|k-1}(\mathbf{X}_k) = \int p(\mathbf{X}_k|\mathbf{X}_{k-1})\mathcal{PMBM}_{k-1|k-1}(\mathbf{X}_{k-1})\,\delta\mathbf{X}_{k-1}$$

Update: $$\mathcal{PMBM}_{k|k}(\mathbf{X}_k) = \frac{p(\mathbf{z}_k|\mathbf{X}_k)\mathcal{PMBM}_{k|k-1}(\mathbf{X}_k)}{\int p(\mathbf{z}_k|\mathbf{X}'_k)\mathcal{PMBM}_{k|k-1}(\mathbf{X}'_k)\,\delta\mathbf{X}'_k}.$$

SET OF TRAJECTORY CONJUGACY

PMBM conjugate prior

- The Poisson Multi-Bernoulli Mixture (PMBM) multitrajectory pdf $\mathcal{PMBM}_{k|k}(\mathbf{X}_k)$ is a conjugate prior to the standard models (point objects, Poisson birth):

Prediction: $$\mathcal{PMBM}_{k|k-1}(\mathbf{X}_k) = \int p(\mathbf{X}_k|\mathbf{X}_{k-1})\mathcal{PMBM}_{k-1|k-1}(\mathbf{X}_{k-1})\,\delta\mathbf{X}_{k-1}$$

Update: $$\mathcal{PMBM}_{k|k}(\mathbf{X}_k) = \frac{p(\mathbf{z}_k|\mathbf{X}_k)\mathcal{PMBM}_{k|k-1}(\mathbf{X}_k)}{\int p(\mathbf{z}_k|\mathbf{X}'_k)\mathcal{PMBM}_{k|k-1}(\mathbf{X}'_k)\,\delta\mathbf{X}'_k}.$$

- Hypotheses and weights are as in the PMBM filters (see Karl's videos).

SET OF TRAJECTORY CONJUGACY

PMBM conjugate prior

- The Poisson Multi-Bernoulli Mixture (PMBM) multitrajectory pdf $\mathcal{PMBM}_{k|k}(\mathbf{X}_k)$ is a conjugate prior to the standard models (point objects, Poisson birth):

Prediction: $$\mathcal{PMBM}_{k|k-1}(\mathbf{X}_k) = \int p(\mathbf{X}_k|\mathbf{X}_{k-1})\mathcal{PMBM}_{k-1|k-1}(\mathbf{X}_{k-1})\,\delta\mathbf{X}_{k-1}$$

Update: $$\mathcal{PMBM}_{k|k}(\mathbf{X}_k) = \frac{p(\mathbf{z}_k|\mathbf{X}_k)\mathcal{PMBM}_{k|k-1}(\mathbf{X}_k)}{\int p(\mathbf{z}_k|\mathbf{X}'_k)\mathcal{PMBM}_{k|k-1}(\mathbf{X}'_k)\,\delta\mathbf{X}'_k}.$$

- Hypotheses and weights are as in the PMBM filters (see Karl's videos).
- Conjugacy also holds for MBM birth and for extended objects.

PPP INTERPRETATION

The PPP component

- Models trajectories of the set of undetected objects.

PPP INTERPRETATION

The PPP component

- Models trajectories of the set of undetected objects.
- We parametrise the PPP using an intensity function, $\lambda^{u}_{k|k'}(X_k)$.

PPP INTERPRETATION

The PPP component

- Models trajectories of the set of undetected objects.
- We parametrise the PPP using an intensity function, $\lambda^u_{k|k'}(X_k)$.

- The intensity $\lambda^u_{k|k'}(X_k)/\bar{\lambda}^u_{k|k'}$ describes a single trajectory distribution over $(\beta, \epsilon, x_{\beta:\epsilon})$.

PPP INTERPRETATION

The PPP component

- Models trajectories of the set of undetected objects.
- We parametrise the PPP using an intensity function, $\lambda^u_{k|k'}(X_k)$.

- The intensity $\lambda^u_{k|k'}(X_k)/\bar{\lambda}^u_{k|k'}$ describes a single trajectory distribution over $(\beta, \epsilon, x_{\beta:\epsilon})$.
- The distribution $\lambda^u_{k|k'}(X_k)/\bar{\lambda}^u_{k|k'}$ conveys information about where we may have undetected objects right now, but also when those objects might have entered the surveillance area.

MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

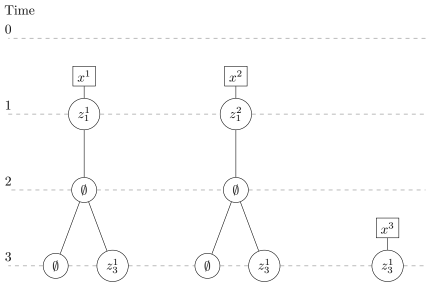
- Why a multi-Bernoull mixture?

MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

- Why a multi-Bernoull mixture?
  - **Mixture**: multiple global hypotheses.

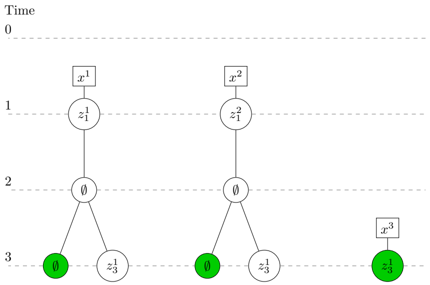


MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

- Why a multi-Bernoull mixture?
  - **Mixture**: multiple global hypotheses.

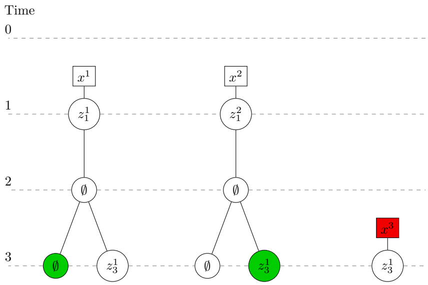


MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

- Why a multi-Bernoull mixture?
  - **Mixture**: multiple global hypotheses.

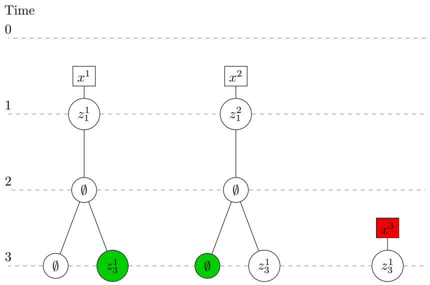


MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

- Why a multi-Bernoull mixture?
  - **Mixture**: multiple global hypotheses.

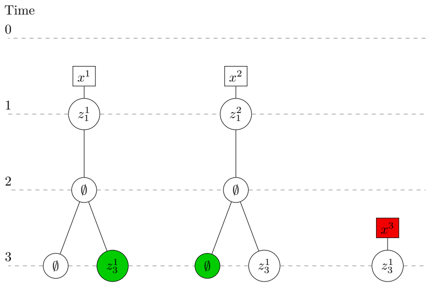


MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

- Why a multi-Bernoull mixture?
  - Mixture: multiple global hypotheses.
  - **MB RFS**: set of detected trajectories for a global hypothesis.

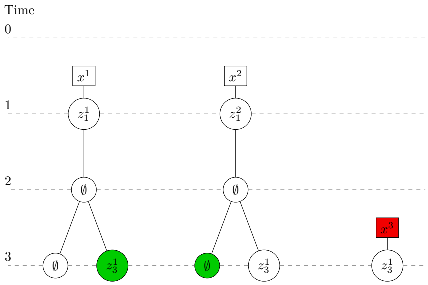


MBM RFS INTERPRETATION

The MBM

- Models trajectories of objects that have been detected (in one of observed measurement sets).

- Why a multi-Bernoull mixture?
  - Mixture: multiple global hypotheses.
  - MB RFS: set of detected trajectories for a global hypothesis.



- **Note:** we describe the distribution of $\mathbf{X}_k$, not just $\mathbf{x}_k$.

PMBM trackers – part 2

Multi-Object Tracking

Lennart Svensson

BERNOULLI RFS: INTERPRETATION

The Bernoulli RFS

- The Bernoulli RFSs model a single potential trajectory given a sequence of associations.

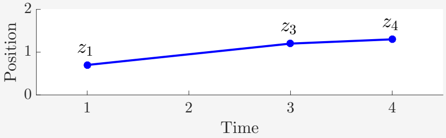
BERNOULLI RFS: INTERPRETATION

The Bernoulli RFS

- The Bernoulli RFSs model a single potential trajectory given a sequence of associations.

Example

- Suppose z_1, z_3 and z_4 are assoc. to a potential object.



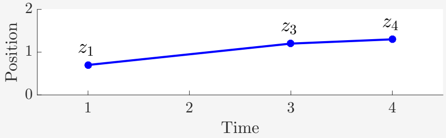
BERNOULLI RFS: INTERPRETATION

The Bernoulli RFS

- The Bernoulli RFSs model a single potential trajectory given a sequence of associations.

Example

- Suppose z_1, z_3 and z_4 are assoc. to a potential object.



- This would yield a Bernoulli RFS

$$p(\mathbf{X}_4^B) = \begin{cases} p_4^B(X) & \text{if } \mathbf{X}_4^B = \{X\} \\ 0 & \text{if } |\mathbf{X}_4^B| \neq 1, \end{cases}$$

$$p_4^B(X) = \delta_{\beta-1}\delta_{\epsilon-4}p_{4|4}(x_{\beta:\epsilon}|\beta, \epsilon)$$

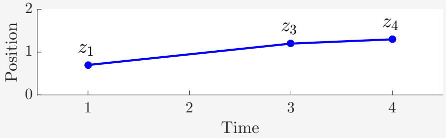
BERNOULLI RFS: INTERPRETATION

The Bernoulli RFS

- The Bernoulli RFSs model a single potential trajectory given a sequence of associations.

Example

- Suppose z_1, z_3 and z_4 are assoc. to a potential object.



- This would yield a Bernoulli RFS

$$p(\mathbf{X}_4^B) = \begin{cases} p_4^B(X) & \text{if } \mathbf{X}_4^B = \{X\} \\ 0 & \text{if } |\mathbf{X}_4^B| \neq 1, \end{cases}$$

$$p_4^B(X) = \delta_{\beta-1}\delta_{\epsilon-4}p_{4|4}(x_{\beta:\epsilon}|\beta, \epsilon)$$

- **Bottom line:** hypotheses describe the association history
  $\Rightarrow$ enables us to compute trajectory distribution!

BERNOULLI RFS APPROXIMATIONS

- In theory, complexity grows with length of trajectory.

BERNOULLI RFS APPROXIMATIONS

- In theory, complexity grows with length of trajectory.
- New measurement $\Rightarrow$ update all of $p(x_{\beta:\epsilon} \big| \beta, \epsilon = k)$.

BERNOULLI RFS APPROXIMATIONS

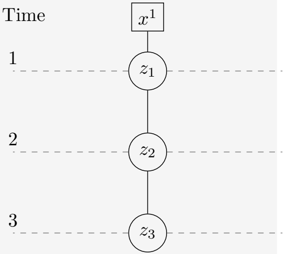
- In theory, complexity grows with length of trajectory.
- New measurement $\Rightarrow$ update all of $p(x_{\beta:\epsilon} \big| \beta, \epsilon = k)$.
- Kalman update is expensive when $\epsilon - \beta$ is large $\Rightarrow$ we need approximations!

BERNOULLI RFS APPROXIMATIONS

- In theory, complexity grows with length of trajectory.
- New measurement $\Rightarrow$ update all of $p(x_{\beta:\epsilon} \mid \beta, \epsilon = k)$.
- Kalman update is expensive when $\epsilon - \beta$ is large $\Rightarrow$ we need approximations!

Approximation example

Exact: $p(x_{1:3} \mid z_{1:3}) \propto p(x_{1:3} \mid z_{1:2}) p(z_3 \mid x_3)$



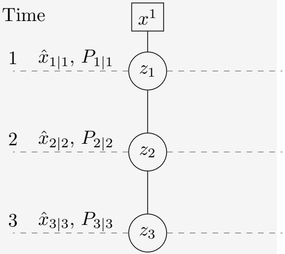
BERNOULLI RFS APPROXIMATIONS

- In theory, complexity grows with length of trajectory.
- New measurement $\Rightarrow$ update all of $p(x_{\beta:\epsilon} \big| \beta, \epsilon = k)$.
- Kalman update is expensive when $\epsilon - \beta$ is large $\Rightarrow$ we need approximations!

Approximation example

Exact: $p(x_{1:3} \big| z_{1:3}) \propto p(x_{1:3} \big| z_{1:2}) p(z_3 \big| x_3)$

Approx.: $p(x_{1:3} \big| z_{1:3}) \approx p(x_3 \big| z_{1:3}) p(x_2 \big| z_{1:2}) p(x_1 \big| z_1)$



BERNOULLI RFS APPROXIMATIONS

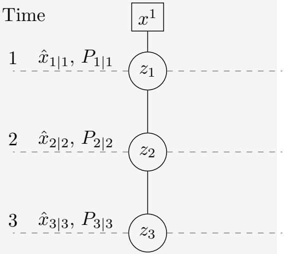
- In theory, complexity grows with length of trajectory.
- New measurement $\Rightarrow$ update all of $p(x_{\beta:\epsilon} \mid \beta, \epsilon = k)$.
- Kalman update is expensive when $\epsilon - \beta$ is large $\Rightarrow$ we need approximations!

Approximation example

Exact: $p(x_{1:3} \mid z_{1:3}) \propto p(x_{1:3} \mid z_{1:2}) p(z_3 \mid x_3)$

Approx.: $p(x_{1:3} \mid z_{1:3}) \approx p(x_3 \mid z_{1:3}) p(x_2 \mid z_{1:2}) p(x_1 \mid z_1)$

- The sequence $\hat{x}_{1|1}, \hat{x}_{2|2}, \ldots$ depends on the local hypothesis.



BERNOULLI RFS APPROXIMATIONS

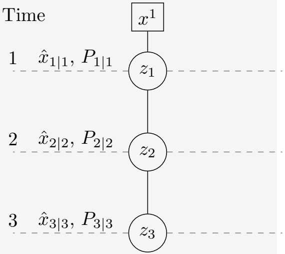
- In theory, complexity grows with length of trajectory.
- New measurement $\Rightarrow$ update all of $p(x_{\beta:\epsilon} | \beta, \epsilon = k)$.
- Kalman update is expensive when $\epsilon - \beta$ is large $\Rightarrow$ we need approximations!

Approximation example

$$\text{Exact:} \quad p(x_{1:3} | z_{1:3}) \propto p(x_{1:3} | z_{1:2}) p(z_3 | x_3)$$

$$\text{Approx.:} \quad p(x_{1:3} | z_{1:3}) \approx p(x_3 | z_{1:3}) p(x_2 | z_{1:2}) p(x_1 | z_1)$$

- The sequence $\hat{x}_{1|1}, \hat{x}_{2|2}, \ldots$ depends on the local hypothesis.
- $\Rightarrow$ same computational complexity as PMBM filtering.



MARGINALIZATION

- Consider a trajectory Bernoulli RFS, with parameters

$$\begin{cases} r_{k|k'} \\ p_{k|k'}(X) = \Pr_{k|k'}(\beta, \epsilon) p^x_{k|k'}(x_{\beta:\epsilon} \big| \beta, \epsilon). \end{cases}$$

MARGINALIZATION

- Consider a trajectory Bernoulli RFS, with parameters

$$\begin{cases} r_{k|k'} \\ p_{k|k'}(X) = \Pr_{k|k'}(\beta, \epsilon) p^{x}_{k|k'}(x_{\beta:\epsilon} \big| \beta, \epsilon). \end{cases}$$

- Using marginalization, we can compute a Bernoulli RFS representing the set of objects at time k.

MARGINALIZATION

- Consider a trajectory Bernoulli RFS, with parameters

$$\begin{cases} r_{k|k'} \\ p_{k|k'}(X) = \mathrm{Pr}_{k|k'}(\beta, \epsilon) p^{x}_{k|k'}(x_{\beta:\epsilon} | \beta, \epsilon). \end{cases}$$

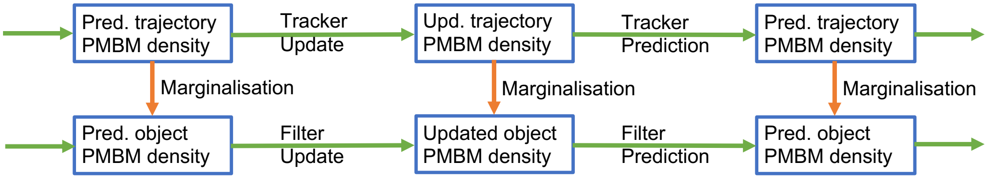
- Using marginalization, we can compute a Bernoulli RFS representing the set of objects at time k.
- This Bernoulli RFS has parameters

$$r^{m}_{k|k'} = r_{k|k'} \sum_{\beta=1}^{k} \mathrm{Pr}_{k|k'}(\beta, k)$$

$$p^{m}_{k|k'}(x_k) = \frac{r_{k|k'}}{r^{m}_{k|k'}} \sum_{\beta=1}^{k} \mathrm{Pr}_{k|k'}(\beta, k) \int p^{x}_{k|k'}(x_{\beta:\epsilon} | \beta, \epsilon) \, \mathrm{d}x_{\beta:k-1}.$$

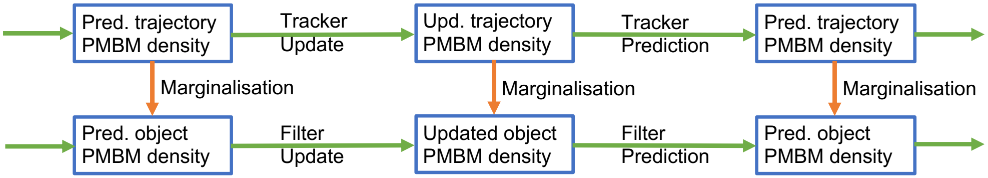
PMBM TRACKERS VS PMBM FILTERS

- More interestingly, marginalizing $\mathcal{PMBM}_{k|k'}(\mathbf{X}_k)$ gives us $\mathcal{PMBM}_{k|k'}(\mathbf{x}_k)$.



PMBM TRACKERS VS PMBM FILTERS

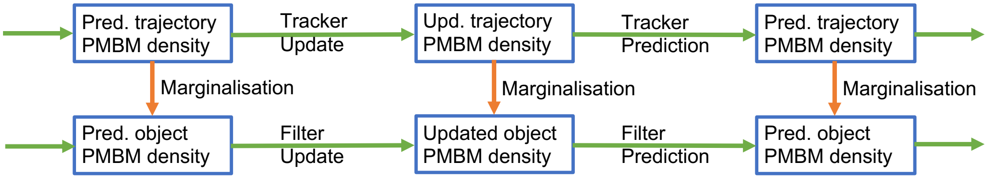
- More interestingly, marginalizing $\mathcal{PMBM}_{k|k'}(\mathbf{X}_k)$ gives us $\mathcal{PMBM}_{k|k'}(\mathbf{x}_k)$.



- **Key insight:**
  - In PMBM filtering we marginalize out the history in every prediction step.

PMBM TRACKERS VS PMBM FILTERS

- More interestingly, marginalizing $\mathcal{PMBM}_{k|k'}(\mathbf{X}_k)$ gives us $\mathcal{PMBM}_{k|k'}(\mathbf{x}_k)$.



- **Key insight:**
  - In PMBM filtering we marginalize out the history in every prediction step.
  - In PMBM trackers, we maintain knowledge about the past which enables us to estimate trajectories.

Simulation example 1 – missed detections

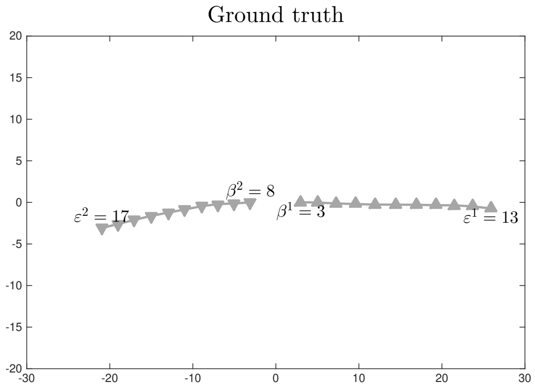
Multi-Object Tracking

Lennart Svensson

SCENARIO WITH MISSED DETECTIONS: SETUP

Two objects

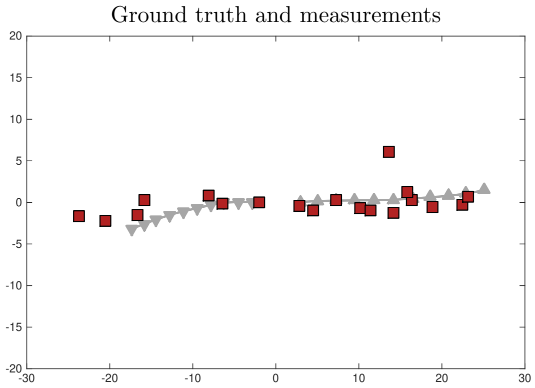
- Twenty time steps
- Two objects appear at time step β^i, disappear at time step ε^i



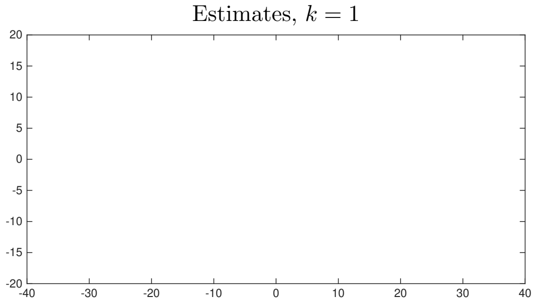
SCENARIO WITH MISSED DETECTIONS: SETUP

Two objects

- Twenty time steps
- Two objects appear at time step β^i, disappear at time step ε^i
- Left object misdetected at times $k = 11, 12, 13$

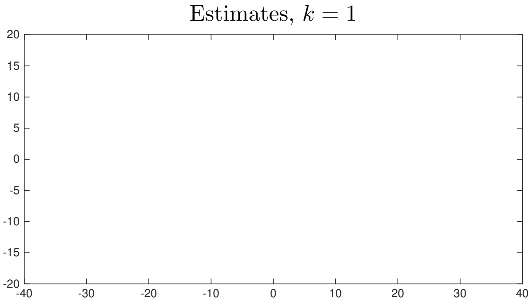


SCENARIO WITH MISSED DETECTIONS: NO SMOOTHING



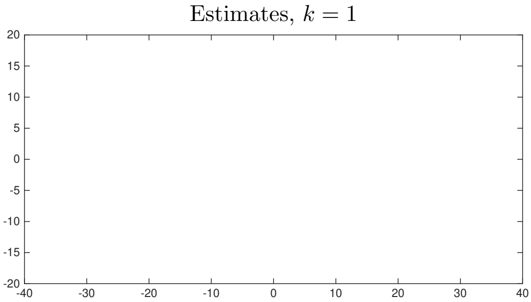
- PMBM tracker
  - Gaussian state densities and a constant velocity model.
  - Trajectory estimates computed at time k. No smoothing.

SCENARIO WITH MISSED DETECTIONS: NO SMOOTHING



- PMBM tracker
  - Gaussian state densities and a constant velocity model.
  - Trajectory estimates computed at time k. No smoothing.
  - Once object is detected again, we produce complete trajectory estimates.

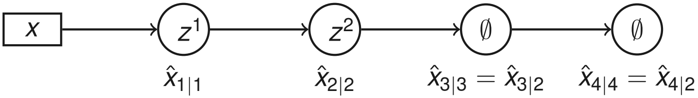
SCENARIO WITH MISSED DETECTIONS: NO SMOOTHING



- PMBM tracker
  - Gaussian state densities and a constant velocity model.
  - Trajectory estimates computed at time k. No smoothing.
  - Once object is detected again, we produce complete trajectory estimates.
  - Trajectory estimates are less accurate when object is not detected.

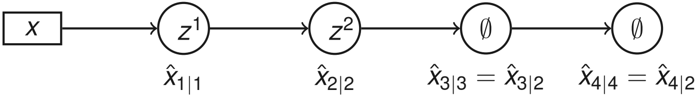
EXPLAINING THE TRACKING RESULTS

- Suppose there is only
  1. one potential object,
  2. one reasonable hypothesis at each time.



EXPLAINING THE TRACKING RESULTS

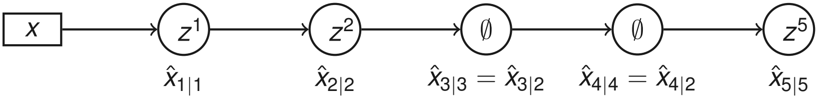
- Suppose there is only
  1. one potential object,
  2. one reasonable hypothesis at each time.



- At time 4, our best estimate may be $\hat{\mathbf{X}}_4 = \{(1, 2, (\hat{x}_{1|1}, \hat{x}_{2|2}))\}$.

EXPLAINING THE TRACKING RESULTS

- Suppose there is only
  1. one potential object,
  2. one reasonable hypothesis at each time.

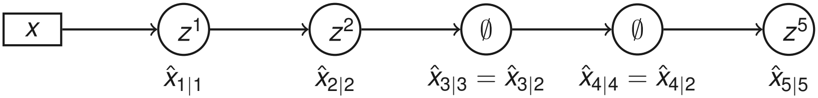


- At time 4, our best estimate may be $\hat{\mathbf{X}}_4 = \{(1, 2, (\hat{x}_{1|1}, \hat{x}_{2|2}))\}$.
- At time 5, our best estimate is that the trajectory continues until time 5:

$$\hat{\mathbf{X}}_5 = \{(1, 5, (\hat{x}_{1|1}, \hat{x}_{2|2}, \hat{x}_{3|3}, \hat{x}_{4|4}, \hat{x}_{5|5}))\}.$$

EXPLAINING THE TRACKING RESULTS

- Suppose there is only
  1. one potential object,
  2. one reasonable hypothesis at each time.



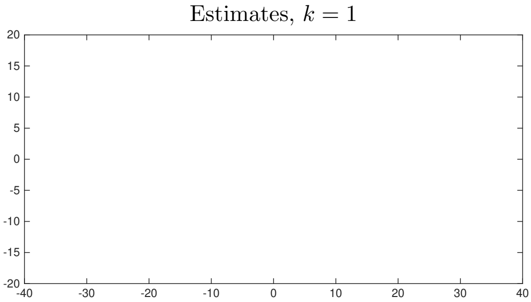
- At time 4, our best estimate may be $\hat{\mathbf{X}}_4 = \{(1, 2, (\hat{x}_{1|1}, \hat{x}_{2|2}))\}$.
- At time 5, our best estimate is that the trajectory continues until time 5:

$$\hat{\mathbf{X}}_5 = \{(1, 5, (\hat{x}_{1|1}, \hat{x}_{2|2}, \hat{x}_{3|3}, \hat{x}_{4|4}, \hat{x}_{5|5}))\}.$$

- Of course, an even better estimate would be:

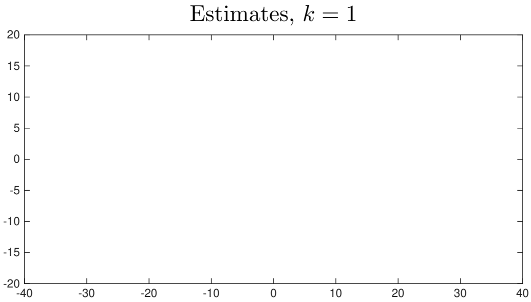
$$\hat{\mathbf{X}}_5 = \{(1, 5, (\hat{x}_{1|5}, \hat{x}_{2|5}, \hat{x}_{3|5}, \hat{x}_{4|5}, \hat{x}_{5|5}))\}.$$

SCENARIO WITH MISSED DETECTIONS: SMOOTHING



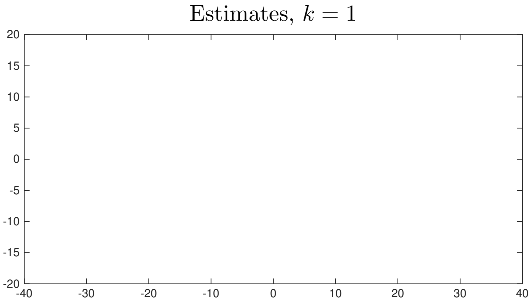
- PMBM tracker
  - Smoothing: use measurements to improve earlier state estimates.

SCENARIO WITH MISSED DETECTIONS: SMOOTHING



- PMBM tracker
  - Smoothing: use measurements to improve earlier state estimates.
  - Final trajectory estimates are accurate also at times when we have missed detections.

SCENARIO WITH MISSED DETECTIONS: SMOOTHING



- PMBM tracker
  - Smoothing: use measurements to improve earlier state estimates.
  - Final trajectory estimates are accurate also at times when we have missed detections.
  - Both versions of the PMBM tracker provide trajectory estimates without gaps.

Simulation example 2 – closely spaced objects

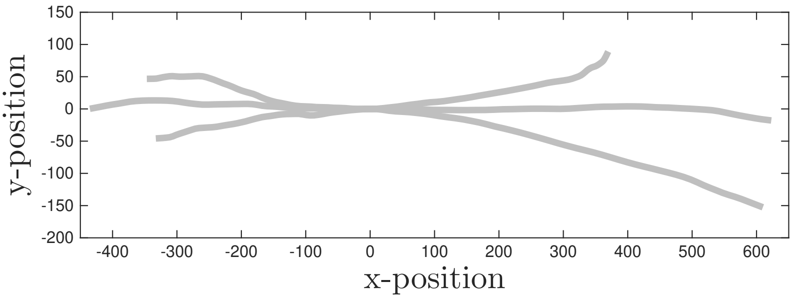
Multi-Object Tracking

Lennart Svensson

CLOSELY SPACED OBJECTS: SETUP

Three objects that become very close, and then separate

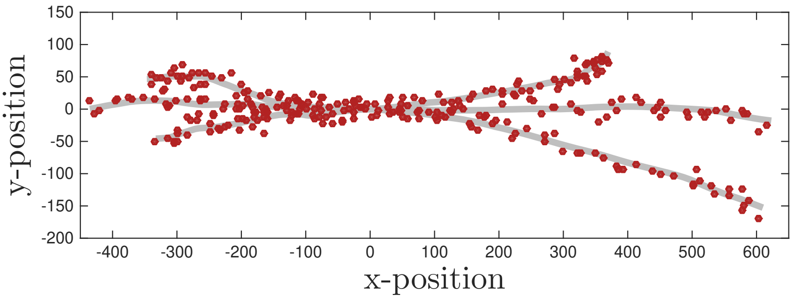
- Objects move left to right, are initially well-separated.



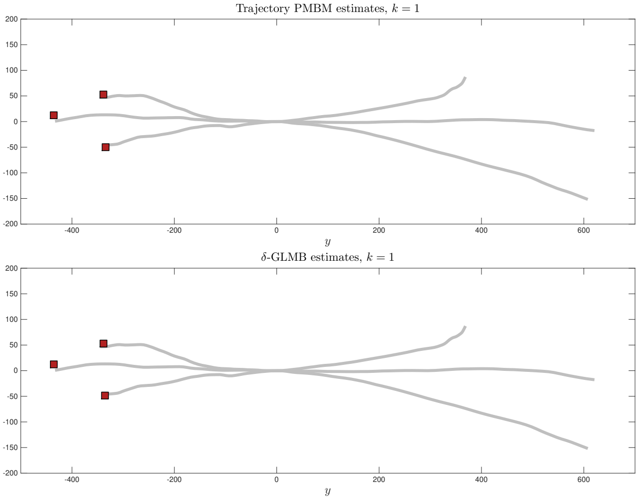
CLOSELY SPACED OBJECTS: SETUP

Three objects that become very close, and then separate

- Objects move left to right, are initially well-separated.
- Sequences last 100 time steps. After time 50, ambiguous which object goes where.



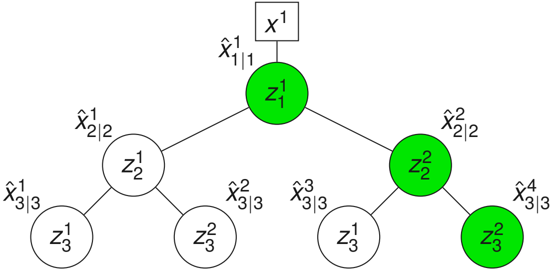
CLOSELY SPACED OBJECTS: TRACKING RESULTS



- PMBM tracker: 1) uses smoothing to improve trajectory estimates, 2) experiences track switching (at $k = 89$), 3) does not report trajectories with unrealistic switches.

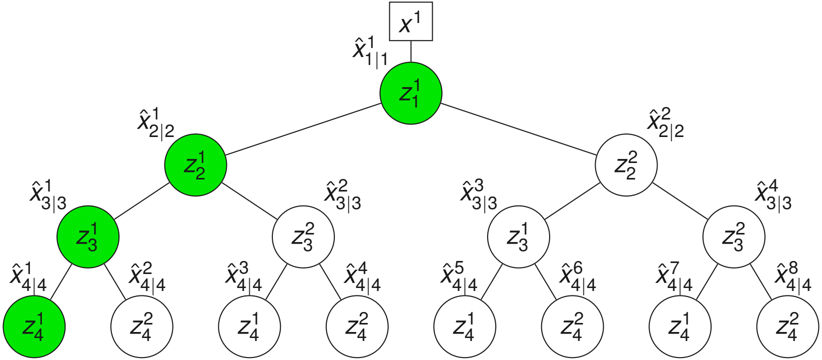
WHY NO TRACK SWITCHES?

- We select estimates from the most likely hypothesis, e.g., $\hat{\mathbf{X}}_3 = \{(1, 3, \hat{x}^1_{1|1}, \hat{x}^2_{2|2}, \hat{x}^4_{3|3})\}$.



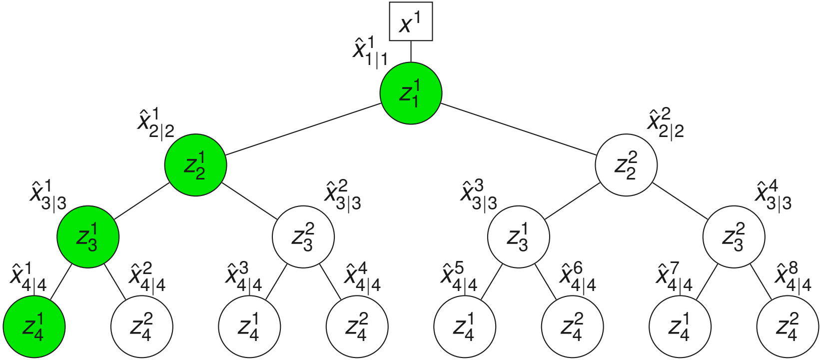
WHY NO TRACK SWITCHES?

- We select estimates from the most likely hypothesis, e.g., $\hat{\mathbf{X}}_3 = \{(1, 3, \hat{x}^1_{1|1}, \hat{x}^2_{2|2}, \hat{x}^4_{3|3})\}$.
- Later estimates may be from a different branch, e.g., $\hat{\mathbf{X}}_4 = \{(1, 4, \hat{x}^1_{1|1}, \hat{x}^1_{2|2}, \hat{x}^1_{3|3}, \hat{x}^1_{4|4})\}$.



WHY NO TRACK SWITCHES?

- We select estimates from the most likely hypothesis, e.g., $\hat{\mathbf{X}}_3 = \{(1, 3, \hat{x}^1_{1|1}, \hat{x}^2_{2|2}, \hat{x}^4_{3|3})\}$.
- Later estimates may be from a different branch, e.g., $\hat{\mathbf{X}}_4 = \{(1, 4, \hat{x}^1_{1|1}, \hat{x}^1_{2|2}, \hat{x}^1_{3|3}, \hat{x}^1_{4|4})\}$.
- **Note:** all estimates are from **the same branch**.



SETS OF TRAJECTORIES: CONCLUSIONS

- Sets of trajectories can leverage on the many advantages with PMBM filters, e.g., few hypotheses and initiate tracks based on measurements.

SETS OF TRAJECTORIES: CONCLUSIONS

- Sets of trajectories can leverage on the many advantages with PMBM filters, e.g., few hypotheses and initiate tracks based on measurements.
- We obtain trajectory estimates, essentially without increasing the computational complexity.

SETS OF TRAJECTORIES: CONCLUSIONS

- Sets of trajectories can leverage on the many advantages with PMBM filters, e.g., few hypotheses and initiate tracks based on measurements.
- We obtain trajectory estimates, essentially without increasing the computational complexity.
- The challenges that we observed with labels are resolved:
  every trajectory estimate in $\hat{\mathbf{X}}_k$ is from a single local hypothesis.

SETS OF TRAJECTORIES: CONCLUSIONS

- Sets of trajectories can leverage on the many advantages with PMBM filters, e.g., few hypotheses and initiate tracks based on measurements.
- We obtain trajectory estimates, essentially without increasing the computational complexity.
- The challenges that we observed with labels are resolved: every trajectory estimate in $\hat{\mathbf{X}}_k$ is from a single local hypothesis.
- Using smoothing, we can improve performance further.

SETS OF TRAJECTORIES: CONCLUSIONS

- Sets of trajectories can leverage on the many advantages with PMBM filters, e.g., few hypotheses and initiate tracks based on measurements.
- We obtain trajectory estimates, essentially without increasing the computational complexity.
- The challenges that we observed with labels are resolved: every trajectory estimate in $\hat{\mathbf{X}}_k$ is from a single local hypothesis.
- Using smoothing, we can improve performance further.
- Sets of trajectories can be used for point objects, extended objects, track-before-detect, etc.

Section 3:
Deep learning

Multi-Object Tracking

Lennart Svensson

An introduction to deep learning

Multi-Object Tracking

Lennart Svensson

DEEP LEARNING VS ARTIFICIAL INTELLIGENCE

- The objective in **artificial intelligence** (A.I.) is to build “smart” machines.

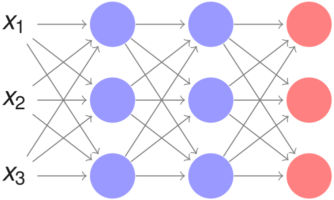
DEEP LEARNING VS ARTIFICIAL INTELLIGENCE

- The objective in **artificial intelligence** (A.I.) is to build “smart” machines.

- In **machine learning**, the idea is to give a machine access to data and ask it to learn for itself.

DEEP LEARNING VS ARTIFICIAL INTELLIGENCE

- The objective in **artificial intelligence** (A.I.) is to build “smart” machines.
- In **machine learning**, the idea is to give a machine access to data and ask it to learn for itself.
- **Deep learning** is a subfield of machine learning where we use **deep neural networks** to make decisions.



SPEECH RECOGNITION AND NATURAL LANGUAGE PROCESSING

- Deep learning has lead to much better **speech analysis**.

Speech Recognition

Reduced word errors by more than 30%

Google Research Blog – August 2012, August 2015

SPEECH RECOGNITION AND NATURAL LANGUAGE PROCESSING

- Deep learning has lead to much better **speech analysis**.

Speech Recognition

Reduced word errors by more than 30%

Google Research Blog – August 2012, August 2015

- It is also used by Google for **translations**.

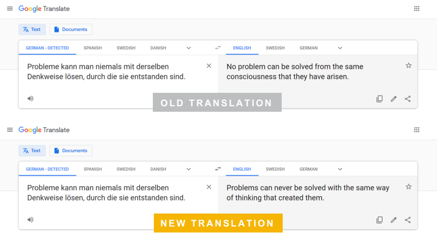


IMAGE CLASSIFICATION

- Top-5 ImageNet classification challenge:
  - 1000 categories,
  - 5 chances to guess,
  - $>$ 1 million images.

Top-5 guesses:

74.21% : mountain bike
5.89% : bicycle-built-for-two
4.37% : crash helmet
2.02% : lakeside
0.76% : alp

IMAGE CLASSIFICATION

- Top-5 ImageNet classification challenge:
    - 1000 categories,
    - 5 chances to guess,
    - $>$ 1 million images.

- First **deep learning** algorithm participated in 2012.

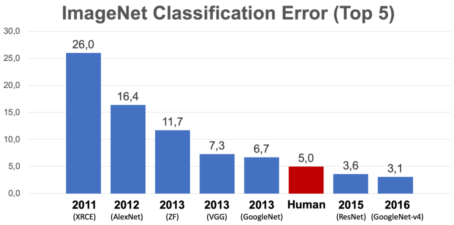
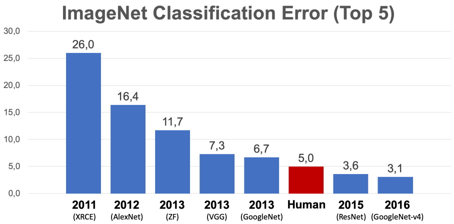


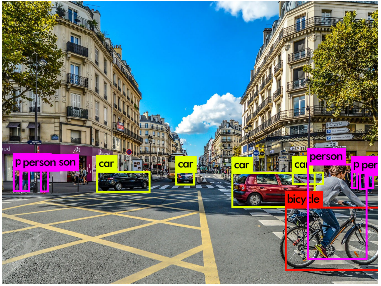
IMAGE CLASSIFICATION

- Top-5 ImageNet classification challenge:
  - 1000 categories,
  - 5 chances to guess,
  - $>$ 1 million images.
- First **deep learning** algorithm participated in 2012.
- Surpassed "human performance" in 2015.



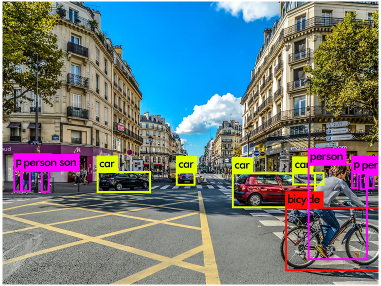
OBJECT DETECTION AND REINFORCEMENT LEARNING

- Deep learning also yields very good **object detection** performance.



OBJECT DETECTION AND REINFORCEMENT LEARNING

- Deep learning also yields very good **object detection** performance.



- In 2016, deep reinforcement learning won against the **world champion in Go**.

Deep neural networks

Multi-Object Tracking

Lennart Svensson

FEEDFORWARD NEURAL NETWORKS

- A neural network is a function.

FEEDFORWARD NEURAL NETWORKS

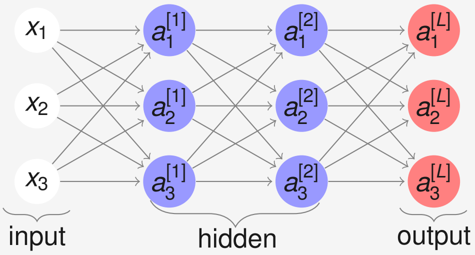
- A neural network is a function.
- Given enough neurons it can approximate (almost) any function.

FEEDFORWARD NEURAL NETWORKS

- A neural network is a function.
- Given enough neurons it can approximate (almost) any function.

Feedforward neural networks

- $L = \sharp$ layers, $L - 1$ hidden layers

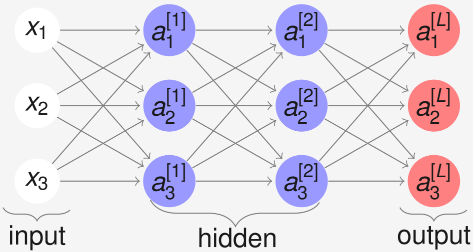


FEEDFORWARD NEURAL NETWORKS

- A neural network is a function.
- Given enough neurons it can approximate (almost) any function.

Feedforward neural networks

- $L = \sharp$ layers, $L - 1$ hidden layers
- $a_i^{[j]}$: value of neuron i in layer j

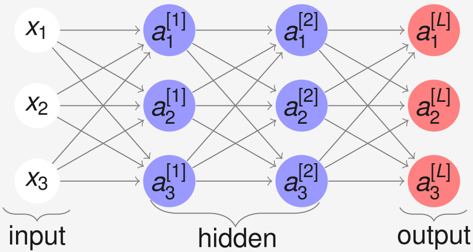


FEEDFORWARD NEURAL NETWORKS

- A neural network is a function.
- Given enough neurons it can approximate (almost) any function.

Feedforward neural networks

- $L = \sharp$ layers, $L - 1$ hidden layers
- $a_i^{[j]}$: value of neuron i in layer j
- We assume $x_i = a_i^{[0]}$.

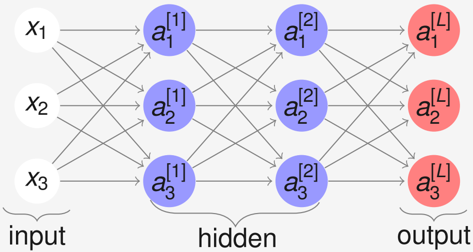


FEEDFORWARD NEURAL NETWORKS

- A neural network is a function.
- Given enough neurons it can approximate (almost) any function.

Feedforward neural networks

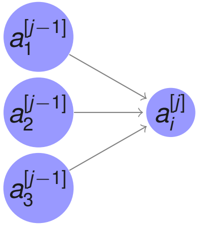
- $L = \sharp$ layers, $L - 1$ hidden layers
- $a_i^{[j]}$: value of neuron i in layer j
- We assume $x_i = a_i^{[0]}$.
- Number of neurons usually varies with depth.



A SINGLE NEURON

- To evaluate a single neuron:

$$z_i^{[j]} = b_i^{[j]} + \sum_i w_i^{[j]} a_i^{[j-1]}$$
$$a_i^{[j]} = g^{[j]}(z_i^{[j]}).$$



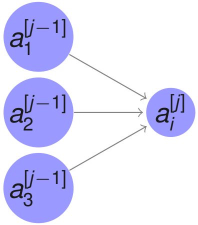
A SINGLE NEURON

- To evaluate a single neuron:

$$z_i^{[j]} = b_i^{[j]} + \sum_i w_i^{[j]} a_i^{[j-1]}$$

$$a_i^{[j]} = g^{[j]}(z_i^{[j]}).$$

- Here $g^{[j]}(\cdot)$ is an **activation function**.



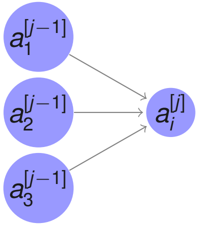
A SINGLE NEURON

- To evaluate a single neuron:

$$z_i^{[j]} = b_i^{[j]} + \sum_i w_i^{[j]} a_i^{[j-1]}$$

$$a_i^{[j]} = g^{[j]}(z_i^{[j]}).$$

- Here $g^{[j]}(\cdot)$ is an **activation function**.
- $\left\{b_i^{[j]}, w_i^{[j]}\right\}_{i,j}$: learned from data.



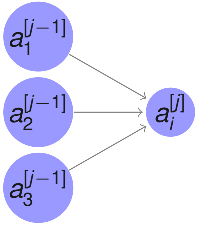
A SINGLE NEURON

- To evaluate a single neuron:

$$z_i^{[j]} = b_i^{[j]} + \sum_i w_i^{[j]} a_i^{[j-1]}$$

$$a_i^{[j]} = g^{[j]}(z_i^{[j]}).$$

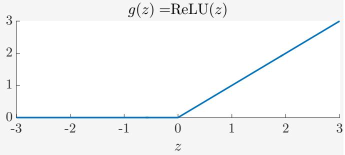
- Here $g^{[j]}(\cdot)$ is an **activation function**.
- $\left\{b_i^{[j]}, w_i^{[j]}\right\}_{i,j}$: learned from data.



Rectified linear units (ReLU)

- The rectified linear unit is a commonly used activation function:

$$g(z) = \mathrm{ReLU}(z) = \max(0, z).$$



ACTIVATION FUNCTIONS – OUTPUT LAYER

- We use different output activation functions for different tasks.

ACTIVATION FUNCTIONS – OUTPUT LAYER

- We use different output activation functions for different tasks.
- **Regression:** we often use $g(z) = z$ or $g(z) = \text{ReLU}(z)$.

ACTIVATION FUNCTIONS – OUTPUT LAYER

- We use different output activation functions for different tasks.
- **Regression:** we often use $g(z) = z$ or $g(z) = \mathrm{ReLU}(z)$.

Softmax

- For classification tasks, we often use a softmax activation

$$\tilde{a}_i^{[L]} = \exp(z_i^{[L]})$$

$$a_i^{[L]} = \frac{\tilde{a}_i^{[L]}}{\sum_s \tilde{a}_s^{[L]}}.$$

ACTIVATION FUNCTIONS – OUTPUT LAYER

- We use different output activation functions for different tasks.
- **Regression:** we often use $g(z) = z$ or $g(z) = \mathrm{ReLU}(z)$.

Softmax

- For classification tasks, we often use a softmax activation

$$\tilde{a}_i^{[L]} = \exp(z_i^{[L]})$$

$$a_i^{[L]} = \frac{\tilde{a}_i^{[L]}}{\sum_s \tilde{a}_s^{[L]}}.$$

- **Note 1:** $a_i^{[L]}$ sum to one, and we use it to approximate $\Pr[Y = i | x]$.

ACTIVATION FUNCTIONS – OUTPUT LAYER

- We use different output activation functions for different tasks.
- **Regression:** we often use $g(z) = z$ or $g(z) = \mathrm{ReLU}(z)$.

Softmax

- For classification tasks, we often use a softmax activation

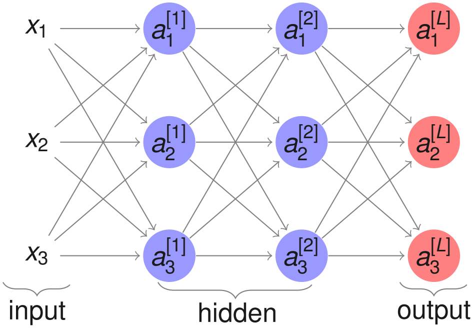
$$\tilde{a}_i^{[L]} = \exp(z_i^{[L]})$$

$$a_i^{[L]} = \frac{\tilde{a}_i^{[L]}}{\sum_s \tilde{a}_s^{[L]}}.$$

- **Note 1:** $a_i^{[L]}$ sum to one, and we use it to approximate $\Pr[Y = i | x]$.
- **Note 2:** $a_i^{[L]}$ depends on $z_1^{[L]}, z_2^{[L]}, \ldots$

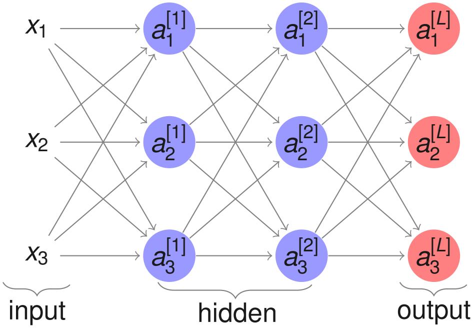
GENERAL NEURAL NETWORKS

- Many neural networks can be viewed as feedforward networks.



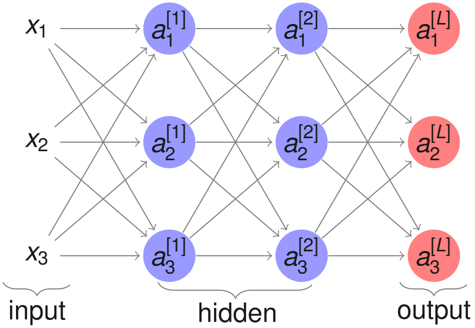
GENERAL NEURAL NETWORKS

- Many neural networks can be viewed as feedforward networks.
- **Common adjustments:** remove edges and weight sharing.



GENERAL NEURAL NETWORKS

- Many neural networks can be viewed as feedforward networks.
- **Common adjustments:** remove edges and weight sharing.
- **Other techniques:** max-pooling, batch normalization, attention, etc.



Supervised learning

Multi-Object Tracking

Lennart Svensson

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

Image classification

In:

Out: cat.

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

Image classification

In:

Out: dog.

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

Common approach:

1. Collect a dataset of inputs x (e.g., images) and labels y.

Image classification

In:

Out: dog.

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

Common approach:

1. Collect a dataset of inputs x (e.g., images) and labels y.
2. Train algorithm to predict y from x. That is, find θ such that

$$f(x;\theta) \approx y$$

(in some sense).

Image classification

In:

Out: dog.

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

Common approach:

1. Collect a dataset of inputs x (e.g., images) and labels y.
2. Train algorithm to predict y from x.
   That is, find θ such that
   $$f(x; \theta) \approx y$$
   (in some sense).
3. Evaluate algorithm on new data.

Image classification

In:

Out: dog.

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

Common approach:

1. Collect a dataset of inputs x (e.g., images) and labels y.
2. Train algorithm to predict y from x. That is, find θ such that
$$f(x;\theta) \approx y$$
(in some sense).
3. Evaluate algorithm on new data.

Image classification

In: 

Out: dog.

- **More examples:**
  - Translation: x sentence in German, y sentence in English.

SUPERVISED LEARNING

- **Supervised learning:** learn to predict an output given an input.

Common approach:

1. Collect a dataset of inputs x (e.g., images) and labels y.
2. Train algorithm to predict y from x. That is, find θ such that

$$f(x;\theta) \approx y$$

   (in some sense).
3. Evaluate algorithm on new data.

Image classification

In:

Out: dog.

- **More examples:**
  - Translation: x sentence in German, y sentence in English.
  - Object detections: x image, y object detections.

EMPIRICAL RISK

- Suppose we are given

$$\text{training data:} \quad \left\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\right\}$$

$$\text{a loss function:} \quad L(f, y).$$

EMPIRICAL RISK

- Suppose we are given

training data: $\left\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\right\}$

a loss function: $L(f, y)$.

Empirical risk

- Normally, we then seek to minimize the **empirical risk**

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} L\left(f(x^{(i)}; \theta), y^{(i)}\right).$$

EMPIRICAL RISK

- Suppose we are given

$$\begin{aligned} \text{training data:} \quad & \left\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\right\} \\ \text{a loss function:} \quad & L(f, y). \end{aligned}$$

Empirical risk

- Normally, we then seek to minimize the **empirical risk**

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L\left(f(x^{(i)}; \theta), y^{(i)}\right).$$

- We now learn how to solve the task by minimizing $J(\theta)$.
  $\Rightarrow$ Optimisation based learning!

EMPIRICAL RISK

- Suppose we are given

$$\begin{aligned} \text{training data:} \quad & \left\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\right\} \\ \text{a loss function:} \quad & L(f, y). \end{aligned}$$

Empirical risk

- Normally, we then seek to minimize the **empirical risk**

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L\left(f(x^{(i)}; \theta), y^{(i)}\right).$$

- We now learn how to solve the task by minimizing $J(\theta)$.
  $\Rightarrow$ Optimisation based learning!
- We use modified versions of gradient descent to minimize $J(\theta)$.

LOSS FUNCTIONS

Loss function for regression

- If $f(x;\theta) = \hat{y}(x;\theta)$, we can use, e.g.,

$$L(f, y) = \|f - y\|_2^2 = (f_1 - y_1)^2 + (f_2 - y_2)^2 + \cdots + (f_n - y_n)^2.$$

LOSS FUNCTIONS

Loss function for regression

- If $f(x;\theta) = \hat{y}(x;\theta)$, we can use, e.g.,

$$L(f,y) = \|f - y\|_2^2 = (f_1 - y_1)^2 + (f_2 - y_2)^2 + \cdots + (f_n - y_n)^2.$$

Note: the empirical risk then approximates the **mean squared error**.

LOSS FUNCTIONS

Loss function for regression

- If $f(x;\theta) = \hat{y}(x;\theta)$, we can use, e.g.,

$$L(f, y) = \|f - y\|_2^2 = (f_1 - y_1)^2 + (f_2 - y_2)^2 + \cdots + (f_n - y_n)^2.$$

Note: the empirical risk then approximates the **mean squared error**.

Cross-entropy and classification, $y \in \{1, 2, \ldots, C\}$

- The **cross-entropy** is the standard loss:

$$L(a^{[L]}, y) = -\log a_y^{[L]}.$$

LOSS FUNCTIONS

Loss function for regression

- If $f(x;\theta) = \hat{y}(x;\theta)$, we can use, e.g.,

$$L(f, y) = \|f - y\|_2^2 = (f_1 - y_1)^2 + (f_2 - y_2)^2 + \cdots + (f_n - y_n)^2.$$

Note: the empirical risk then approximates the **mean squared error**.

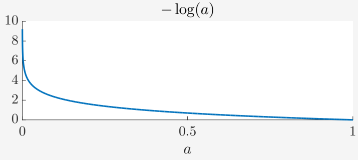
Cross-entropy and classification, $y \in \{1, 2, \ldots, C\}$

- The **cross-entropy** is the standard loss:

$$L(a^{[L]}, y) = -\log a_y^{[L]}.$$

- We want to maximise

$$a_y^{[L]} = \Pr[Y = y \,|\, X = x, \theta].$$



An introduction to object detection

Multi-Object Tracking

Lennart Svensson

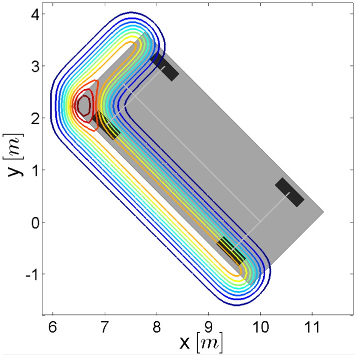
DEEP LEARNING AND MOT

- How can we use deep learning for multi-object tracking (MOT)?

DEEP LEARNING AND MOT

– How can we use deep learning for multi-object tracking (MOT)?

- Perform tracking directly on, e.g., lidar point clouds.
  No need for **deep learning**.



DEEP LEARNING AND MOT

– How can we use deep learning for multi-object tracking (MOT)?

- Perform tracking directly on, e.g., lidar point clouds.
  No need for **deep learning**.
- **Detect objects** using deep learning.
  Detections fed into a MOT alg. ⇒ estimate set of objects/trajectories.

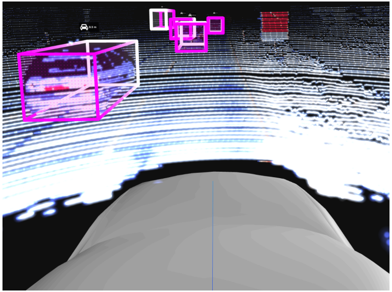


Image created using data from Geiger et al (2012), "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite", *CVPR*.

DEEP LEARNING AND MOT

– How can we use deep learning for multi-object tracking (MOT)?

- Perform tracking directly on, e.g., lidar point clouds.
  No need for **deep learning**.
- **Detect objects** using deep learning. Detections fed into a MOT alg. $\Rightarrow$ estimate set of objects/trajectories.
- Compute **data association probabilities** using deep learning. Use object features to recognize an object at different times.

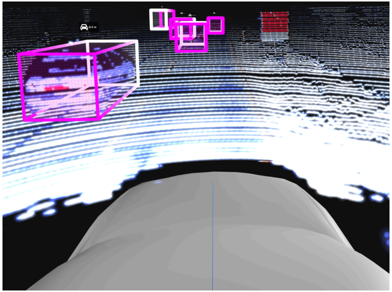


Image created using data from Geiger et al (2012), "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite", *CVPR*.

DEEP LEARNING AND MOT

– How can we use deep learning for multi-object tracking (MOT)?

- Perform tracking directly on, e.g., lidar point clouds.
  No need for **deep learning**.
- **Detect objects** using deep learning. Detections fed into a MOT alg. ⇒ estimate set of objects/trajectories.
- Compute **data association probabilities** using deep learning. Use object features to recognize an object at different times.
- **Perform MOT** purely based on deep learning.
  Directly estimate object trajectories.

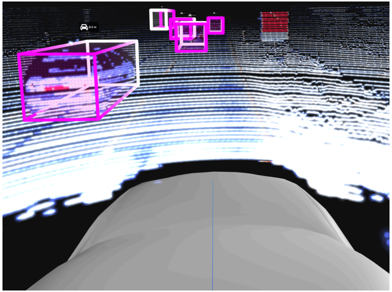


Image created using data from Geiger et al (2012), "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite", *CVPR*.

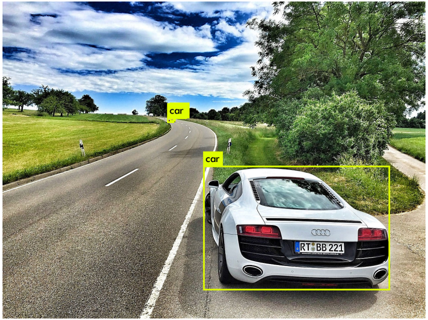
WHY OBJECT DETECTION?

- Why focus on using deep learning for **object detection**?

WHY OBJECT DETECTION?

– Why focus on using deep learning for **object detection**?

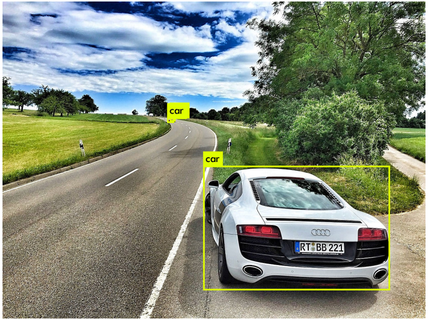
- Techniques are fairly “mature”: already yield good performance.



WHY OBJECT DETECTION?

– Why focus on using deep learning for **object detection**?

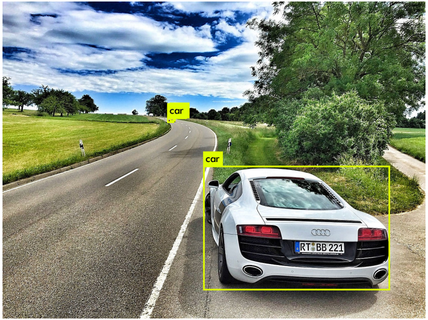
- Techniques are fairly “mature”:
  already yield good performance.
- Challenging to directly work with, e.g., raw images.



WHY OBJECT DETECTION?

– Why focus on using deep learning for **object detection**?

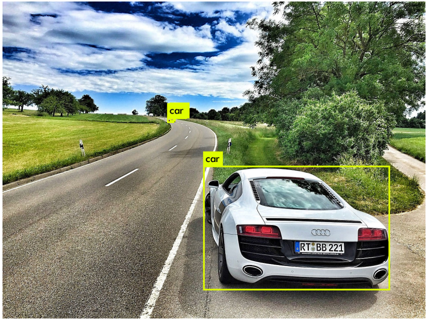
- Techniques are fairly “mature”: already yield good performance.
- Challenging to directly work with, e.g., raw images.
- Enables us to still leverage on motion models and MOT algorithms.



WHY OBJECT DETECTION?

– Why focus on using deep learning for **object detection**?

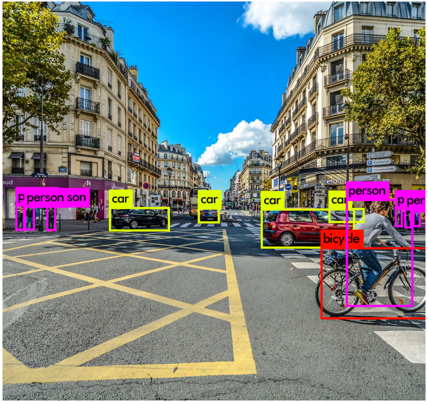
- Techniques are fairly “mature”: already yield good performance.
- Challenging to directly work with, e.g., raw images.
- Enables us to still leverage on motion models and MOT algorithms.
- Often manageable to obtain the data needed to train an object detection algorithm.



WHAT IS OBJECT DETECTION?

Object detection

- Determine the number of objects (of relevant classes).

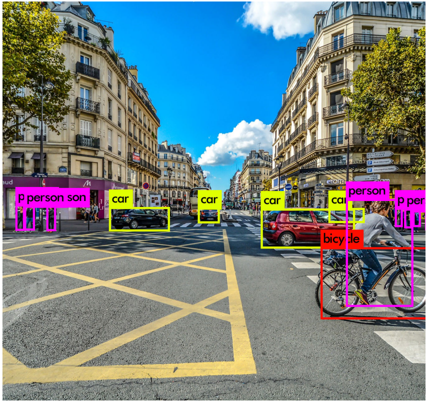


WHAT IS OBJECT DETECTION?

Object detection

- Determine the number of objects (of relevant classes).
- For each object, determine:

$$\begin{cases}\text{object class} \\ \text{object shape.}\end{cases}$$



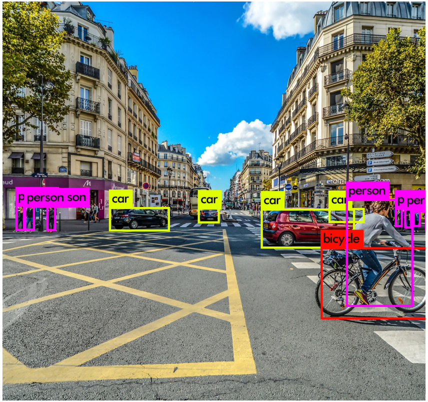
WHAT IS OBJECT DETECTION?

Object detection

- Determine the number of objects (of relevant classes).
- For each object, determine:

$$\begin{cases} \text{object class} \\ \text{object shape.} \end{cases}$$

- Object shape is often a bounding box in 2D or 3D.



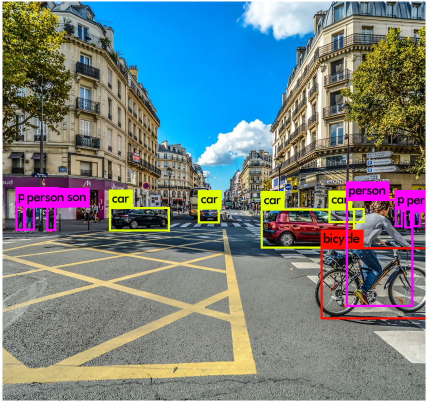
WHAT IS OBJECT DETECTION?

Object detection

- Determine the number of objects (of relevant classes).
- For each object, determine:

  object class
  object shape.

- Object shape is often a bounding box in 2D or 3D.
- **Potential challenges:** many objects, partially occluded objects, objects of different sizes and distances, etc.



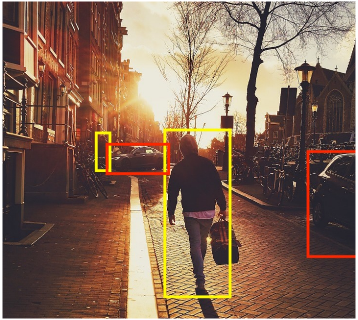
Single shot detectors – training

Multi-Object Tracking

Lennart Svensson

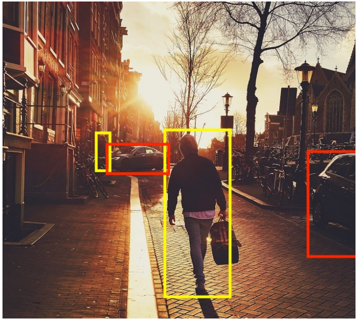
BASIC APPROACH

- Suppose we only have two classes: pedestrians and cars.



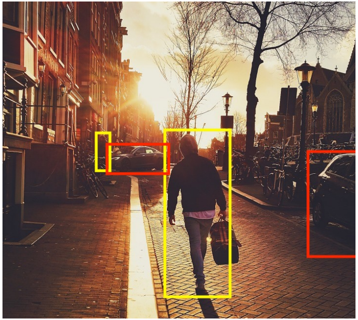
BASIC APPROACH

- Suppose we only have two classes: pedestrians and cars.
- **Key challenge:** number of objects is unknown
  $\Rightarrow$ variable number of outputs!



BASIC APPROACH

- Suppose we only have two classes: pedestrians and cars.
- **Key challenge:** number of objects is unknown $\Rightarrow$ variable number of outputs!



Suggested solution

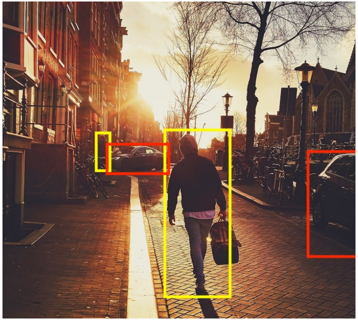
- Provide a fixed number of bounding boxes+classifications.

BASIC APPROACH

- Suppose we only have two classes: pedestrians and cars.
- **Key challenge:** number of objects is unknown $\Rightarrow$ variable number of outputs!

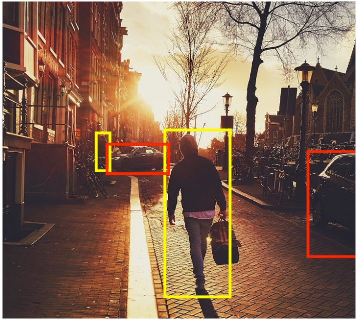
Suggested solution

- Provide a fixed number of bounding boxes+classifications.
- Classify bounded boxes as “object” or “not an object”.



BASIC APPROACH

- Suppose we only have two classes: pedestrians and cars.
- **Key challenge:** number of objects is unknown $\Rightarrow$ variable number of outputs!



Suggested solution

- Provide a fixed number of bounding boxes+classifications.
- Classify bounded boxes as “object” or “not an object”.

- By only considering “objects” we produce a variable number of boxes+classifications.

TRAINING

- Separate input into 3×3 cells.

TRAINING

- Separate input into 3×3 cells.
- For each cell, we want to produce

$$y = \begin{bmatrix} p \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \text{0 if “no object”, 1 if “object”} \\ \\ \\ \\ \end{bmatrix}.$$

TRAINING

- Separate input into 3×3 cells.
- For each cell, we want to produce

$$y = \begin{bmatrix} p \\ c_1 \\ c_2 \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \text{0 if “no object”, 1 if “object”} \\ \text{1 if “pedestrian”, 0 “otherwise”} \\ \text{1 if “car”, 0 “otherwise”} \\ \\ \\ \\ \end{bmatrix}.$$

TRAINING

- Separate input into 3×3 cells.
- For each cell, we want to produce

$$y = \begin{bmatrix} p \\ c_1 \\ c_2 \\ x \\ y \\ w \\ h \end{bmatrix} = \begin{bmatrix} \text{0 if “no object”, 1 if “object”} \\ \text{1 if “pedestrian”, 0 “otherwise”} \\ \text{1 if “car”, 0 “otherwise”} \\ \text{x-position} \\ \text{y-position} \\ \text{width} \\ \text{height} \end{bmatrix}.$$

TRAINING

- Separate input into 3×3 cells.
- For each cell, we want to produce

$$y = \begin{bmatrix} p \\ c_1 \\ c_2 \\ x \\ y \\ w \\ h \end{bmatrix} = \begin{bmatrix} \text{0 if “no object”, 1 if “object”} \\ \text{1 if “pedestrian”, 0 “otherwise”} \\ \text{1 if “car”, 0 “otherwise”} \\ \text{x-position} \\ \text{y-position} \\ \text{width} \\ \text{height} \end{bmatrix}.$$

- For **green cell**:

$$y = \begin{bmatrix} 1 & 0 & 1 & x & y & w & h \end{bmatrix}^T$$

TRAINING

- Separate input into 3×3 cells.
- For each cell, we want to produce

$$y = \begin{bmatrix} p \\ c_1 \\ c_2 \\ x \\ y \\ w \\ h \end{bmatrix} = \begin{bmatrix} \text{0 if "no object", 1 if "object"} \\ \text{1 if "pedestrian", 0 "otherwise"} \\ \text{1 if "car", 0 "otherwise"} \\ \text{x-position} \\ \text{y-position} \\ \text{width} \\ \text{height} \end{bmatrix}.$$

- For green cell:

$$y = \begin{bmatrix} 0 & ? & ? & ? & ? & ? & ? \end{bmatrix}^T$$

TRAINING

- Separate input into 3×3 cells.
- For each cell, we want to produce

$$y = \begin{bmatrix} p \\ c_1 \\ c_2 \\ x \\ y \\ w \\ h \end{bmatrix} = \begin{bmatrix} \text{0 if "no object", 1 if "object"} \\ \text{1 if "pedestrian", 0 "otherwise"} \\ \text{1 if "car", 0 "otherwise"} \\ \text{x-position} \\ \text{y-position} \\ \text{width} \\ \text{height} \end{bmatrix}.$$

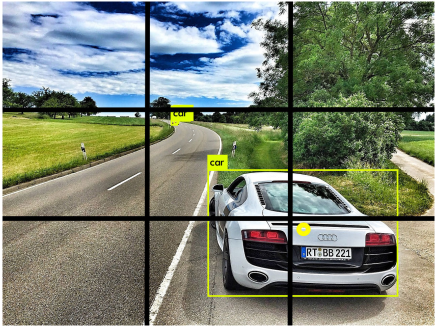
- During training, we try to make network output "similar" to these vectors.

- For **green cell**:

$$y = \begin{bmatrix} 1 & 0 & 1 & x & y & w & h \end{bmatrix}^T$$

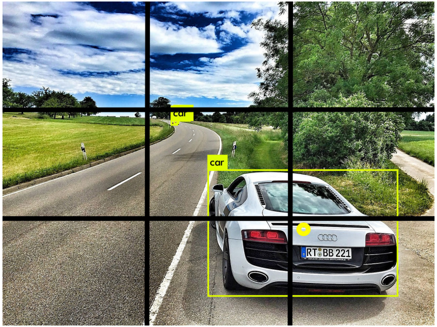
ANCHOR BOXES

- **Q:** Which cell is an object associated to?



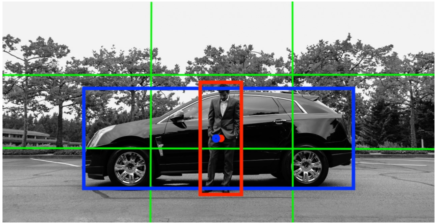
ANCHOR BOXES

- **Q:** Which cell is an object associated to?
  A: The cell that contains its center point.



ANCHOR BOXES

- **Q:** Which cell is an object associated to?
  A: The cell that contains its center point.

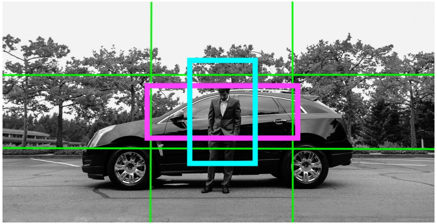


- **Q:** Can we handle multiple center points in one cell?

ANCHOR BOXES

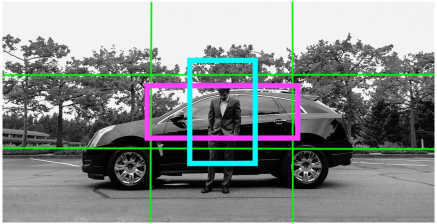
- **Q:** Which cell is an object associated to?
  A: The cell that contains its center point.

- **Q:** Can we handle multiple center points in one cell?
  A: We have multiple **anchor boxes** in each cell.



ANCHOR BOXES

- **Q:** Which cell is an object associated to?
  A: The cell that contains its center point.

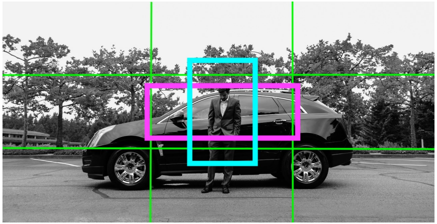


- **Q:** Can we handle multiple center points in one cell?
  A: We have multiple **anchor boxes** in each cell.

- **Q:** What is an anchor box?

ANCHOR BOXES

- **Q:** Which cell is an object associated to?
  A: The cell that contains its center point.



- **Q:** Can we handle multiple center points in one cell?
  A: We have multiple **anchor boxes** in each cell.

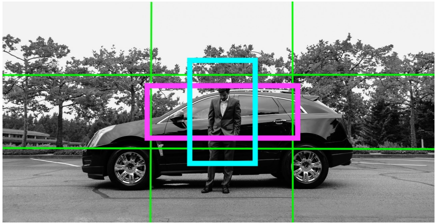
- **Q:** What is an anchor box?
  A: An initial guess for a bounding box (fixed size). We output one y for every anchor box.

ANCHOR BOXES

- **Q:** Which cell is an object associated to?
  A: The cell that contains its center point.
  Note: we associate the object to an anchor box, in that cell, with similar shape (highest IoU).

- **Q:** Can we handle multiple center points in one cell?
  A: We have multiple **anchor boxes** in each cell.

- **Q:** What is an anchor box?
  A: An initial guess for a bounding box (fixed size). We output one y for every anchor box.



Single shot detectors – testing

Multi-Object Tracking

Lennart Svensson

TOO MANY BOUNDING BOXES

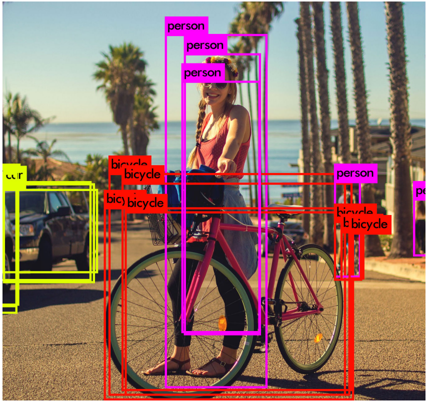
- We compute a vector y for every anchor box.

TOO MANY BOUNDING BOXES

- We compute a vector y for every anchor box.

- We often then remove all bounding boxes:

$$p < 0.5.$$

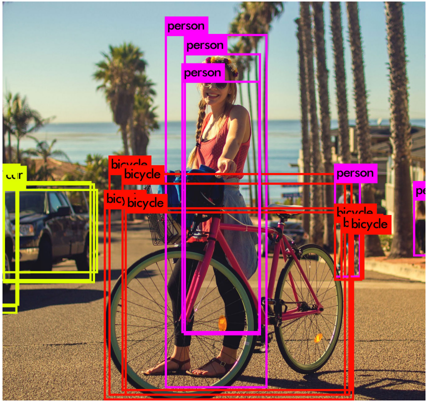


TOO MANY BOUNDING BOXES

- We compute a vector y for every anchor box.
- We often then remove all bounding boxes:

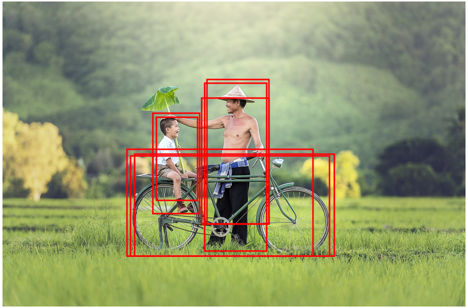
$$p < 0.5.$$

- We are normally left with a few bounding boxes for each object.



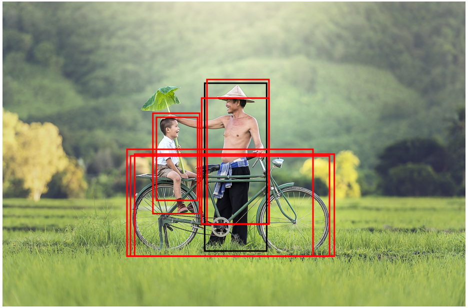
NON-MAXIMUM SUPPRESSION

- 1. Remove all anchor boxes: $p < 0.5$.



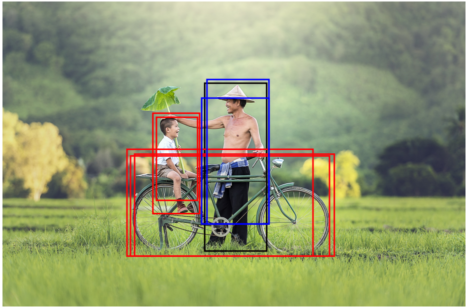
NON-MAXIMUM SUPPRESSION

- 1. Remove all anchor boxes: $p < 0.5$.
- 2. Find anchor box, i, with largest p_i,
  and store i to set of approved anchor boxes.



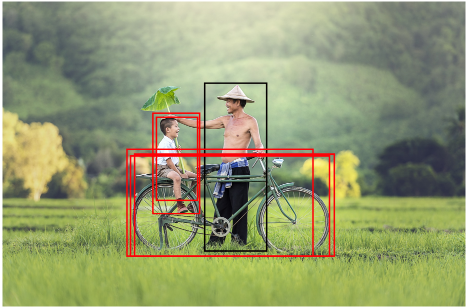
NON-MAXIMUM SUPPRESSION

- 1. Remove all anchor boxes: $p < 0.5$.
- 2. Find anchor box, i, with largest p_i,
  and store i to set of approved anchor boxes.
- 3. Remove all anchor boxes:
  - the same most probable class,
  - the bounding box overlaps substantially with bounding box i.



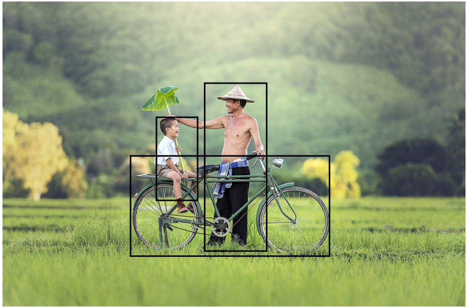
NON-MAXIMUM SUPPRESSION

- 1. Remove all anchor boxes: $p < 0.5$.
- 2. Find anchor box, i, with largest p_i, and store i to set of approved anchor boxes.
- 3. Remove all anchor boxes:
  - the same most probable class,
  - the bounding box overlaps substantially with bounding box i.



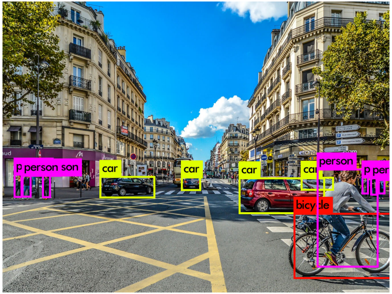
NON-MAXIMUM SUPPRESSION

- 1. Remove all anchor boxes: $p < 0.5$.
- 2. Find anchor box, i, with largest p_i, and store i to set of approved anchor boxes.
- 3. Remove all anchor boxes:
  - the same most probable class,
  - the bounding box overlaps substantially with bounding box i.
- 4. Repeat.



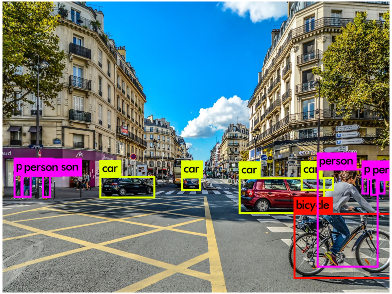
CONCLUSIONS

- We have studied **object detection** using deep neural networks.



CONCLUSIONS

- We have studied **object detection** using deep neural networks.
- For every object in the image, we want to produce a classification and a bounding box.

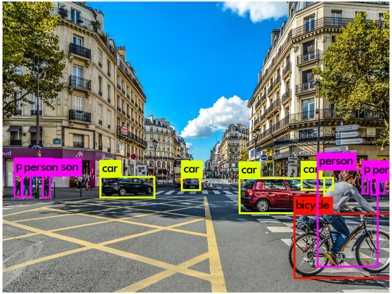


CONCLUSIONS

- We have studied **object detection** using deep neural networks.

- For every object in the image, we want to produce a classification and a bounding box.

- Object detection can also be used for lidar data, stereo images, etc,



CONCLUSIONS

- We have studied **object detection** using deep neural networks.
- For every object in the image, we want to produce a classification and a bounding box.
- Object detection can also be used for lidar data, stereo images, etc, and combined MOT algorithms.