Monte Carlo Filtering Using Kernel Embedding of Distributions

Motonobu Kanagawa
The Institute of Statistical Mathematics, Tokyo

Gatsby Unit External Seminar: 19 Nov. 2014
(Joint work with Yu Nishiyama, Arthur Gretton, and Kenji Fukumizu)

Introduction and Problem Setup

Filtering with a state-space model

We deal with time-series data consisting of

- state:

$$x_1, x_2, \dots, x_t, \dots \in \mathcal{X}$$

- observation:

$$y_1, y_2, \dots, y_t, \dots \in \mathcal{Y}$$

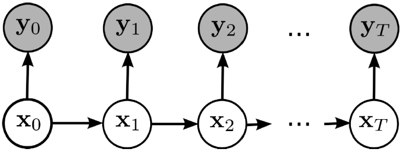
State-space model: these data are generated by

- transition model $p(x_t|x_{t-1})$:

$$x_t \sim p(x_t|x_{t-1})$$

- observation model $p(y_t|x_t)$:

$$y_t \sim p(y_t|x_t)$$



Filtering with a state-space model

Filtering: For each time t, we observe y_t.
Then estimate the posterior distribution on state x_t, given a history $y_{1:t} := y_1, \ldots, y_t$:

$$p(x_t|y_{1:t}).$$

Make use of Bayes' rule:

$$\begin{aligned} p(x_t|y_{1:t}) &\propto \underbrace{p(y_t|x_t)}_{likelihood}\,\underbrace{p(x_t|y_{1:t-1})}_{prior} \\ &= p(y_t|x_t)\int p(x_t|x_{t-1})\,\underbrace{p(x_{t-1}|y_{1:t-1})}_{posterior\ \ at\ \ t-1}\,dx_{t-1}. \end{aligned}$$

Posterior estimation is to be done recursively.

Basics of filtering algorithms

Suppose $p(x_{t-1}|y_{1:t-1})$ was already estimated.

At time t, decompose posterior estimation into two steps:

1. Prediction: Estimate the prior

$$p(x_t|y_{1:t-1}) = \int \underbrace{p(x_t|x_{t-1})}_{transition\ \ model} p(x_{t-1}|y_{1:t-1})dx_{t-1}.$$

2. Correction: Given y_t, estimate the posterior with Bayes' rule

$$p(x_t|y_{1:t}) \propto \underbrace{p(y_t|x_t)}_{observation\ \ model} \underbrace{p(x_t|y_{1:t-1})}_{prior}$$

$p(x_t|x_{t-1})$ and $p(y_t|x_t)$ are assumed to be known.

Existing filtering methods

Kalman filters:

- $p(x_t|x_{t-1})$ and $p(y_t|x_t)$ are assumed to be linear-Gaussian.
- Nonlinear approximation: extended/unscented kalman filters.

Particle filters (Doucet et al., 2001):

- $p(x_t|x_{t-1})$ and $p(y_t|x_t)$ can be nonlinear-nonGaussian.
- Posterior is estimated as a weighted empirical distribution:

$$\hat{p}(x_t|y_{1:t}) = \sum_{i=1}^{n} w_{t,i}\delta_{X_{t,i}},$$

- $w_{t,1}, \dots w_{t,n} \geq 0$ are importance weights.
- $X_{t,1}, \dots, X_{t,n} \in \mathcal{X}$ are called particles.

Particle filters (PF) (Doucet et al., 2001)

Suppose $p(x_{t-1}|y_{1:t-1})$ was already estimated as

$$\hat{p}(x_{t-1}|y_{1:t-1}) = \sum_{i=1}^{n} w_{t-1,i}\delta_{X_{t-1,i}}$$

At time t, PF estimates $p(x_t|y_{1:t})$ by the following steps:

1. Prediction step : estimate $p(x_t|y_{1:t-1})$ by sampling:

$$X_{t,i} \sim \underbrace{p(x_t|x_{t-1} = X_{t-1,i})}_{transition\ \ model}, \quad (i = 1, ..., n),$$

$$\hat{p}(x_t|y_{1:t-1}) := \sum_{i=1}^{n} w_{t-1,i}\delta_{X_{t,i}}.$$

2. Correction step : given y_t, estimate $p(x_t|y_{1:t})$ by importance weighting:

$$w_{t,i} \propto \underbrace{p(y_t|x_t = X_{t,i})}_{observation\ \ model} w_{t-1,i}, \quad (i = 1, ..., n)$$

$$\hat{p}(x_t|y_{1:t}) = \sum_{i=1}^{n} w_{t,i}\delta_{X_{t,i}}.$$

3. Resampling step: resample particles from $\hat{p}(x_t|y_{1:t})$ to collapse small weight particles.

Limitation of particle filters

PF requires evaluation of the observation model $p(y_t|x_t)$:

$$w_{t,i} \propto \underbrace{p(y_t|x_t = X_{t,i})}_{observation\ \ model} w_{t-1,i}.$$

But this may not be possible if

- $p(y_t|x_t)$ is unknown: e.g. robot localization (Vlassis et al., 2002), brain computer interface (Wang et al., 2011).
- $p(y_t|x_t)$ is intractable: e.g. econometrics (Jasra et al., 2012).

Example 1: vision-based mobile robot localization

Problem: estimate a mobile robot's position x_t for each time t, only given its vision images $y_1, \ldots, y_t$.

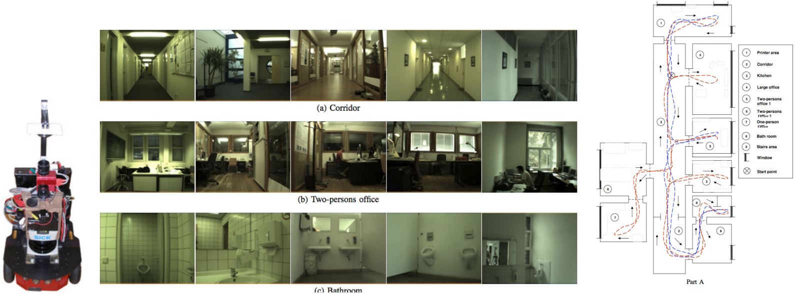


Figure: COLD database (Pronobis and Caputo, 2009)

Example 1: vision-based mobile robot localization

State-space formulation:

- state x_t: position (2/3-dim location + angle).
- observation y_t: vision image (high-dimensional).
- Problem reduces to filtering: estimation of $p(x_t|y_{1:t})$.

Observation model $p(y_t|x_t)$: conditional distribution on vision-images given a position:

- unknown: parametric models cannot be easily defined.
- But, training samples $\{(X_i, Y_i)\}_{i=1}^n$ are available (collected in training phase).

Transition model $p(x_t|x_{t-1})$: robot motion model; many available models (Thrun et al., 2005).

Example 2: brain-computer interface

Problem: decoding subject's finger flexion x_t from ECoG signals $y_1, \ldots, y_t$.

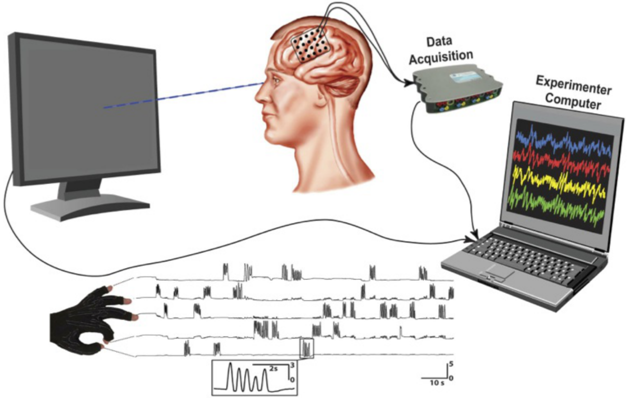


Figure: Experimental setup (Wang et al., 2011)

Example 2: brain-computer interface

State-space modeling is possible for BCI tasks (Pistohl et al., 2008; Wang et al., 2011):

- state x_t: finger positons (5 dim)
- observation y_t: ECoG signals (64 dim)
- Decoding can be formulated as filtering: $p(x_t|y_{1:t})$

Observation model $p(y_t|x_t)$: conditional distribution on ECoG signals given finger positions:

- unknown: parametric models cannot be easily defined.
- But, training samples $\{(X_i, Y_i)\}_{i=1}^n$ are available (collected in training phase).

Transition model $p(x_t|x_{t-1})$: can be modeled using prior knowledge (Wang et al., 2011)

Problem setting

Filtering under the following assumptions:

- transition model $p(x_t|x_{t-1})$: known (same as PF).
- observation model $p(y_t|x_t)$: unknown. But training samples are given:

$$(X_1, Y_1), \ldots, (X_n, Y_n) \subset \mathcal{X} \times \mathcal{Y}.$$

We develop a filter based on kernel embedding of distributions.

Preliminaries: Kernel Embedding of Distributions

Positive definite kernel

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel.

e.g. for $\mathcal{X} = \mathbb{R}^d$

- Gaussian kernel: $k(x, x') = \exp(-\|x - x'\|^2/\gamma^2)$.
- Polynomial kernel: $k(x, x') = (\langle x, x' \rangle + c)^m$.

$\mathcal{X}$ can be a set of structured data, such as image, text and graph, once an appropriate kernel is defined (Schölkopf and Smola, 2002).

Reproducing Kernel Hilbert space (RKHS)

Any positive definite kernel k defines an RKHS $\mathcal{H}$, such that

- For all $x \in \mathcal{X}$,
$$k(\cdot, x) \in \mathcal{H}.$$
- For all $f \in \mathcal{H}$ and $x \in \mathcal{X}$ (reproducing property),
$$f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}.$$

Function $k(\cdot, x) \in \mathcal{H}$ is called the feature of x.

- Kernel is the inner-product between the features:
$$k(x, x') = \left\langle k(\cdot, x), k(\cdot, x') \right\rangle_{\mathcal{H}}, \quad x, x' \in \mathcal{X}.$$

Kernel embedding of distributions (Smola et al., 2007)

Represent any probability distribution P on $\mathcal{X}$ as the mean of features:

$$m_P := \int k(\cdot, x) dP(x) \in \mathcal{H},$$

which will be referred to as the kernel mean of P.

- If k is characteristic, e.g. the Gaussian kernel on $\mathbb{R}^d$, m_P uniquely identifies P (Fukumizu et al., 2008):

$$m_P = m_Q \Rightarrow P = Q$$

- Estimation of P can be cast as estimation of m_P.

Empirical estimate

Given i.i.d. sample $X_1, \ldots, X_n$ from P, one can estimate m_P by

$$\hat{m}_P := \frac{1}{n} \sum_{i=1}^{n} k(\cdot, X_i).$$

Convergence rate: $\|\hat{m}_P - m_P\|_{\mathcal{H}} = O_P(n^{-1/2})$
(Smola et al., 2007).
Other popular estimators:

- Conditional mean embedding (Song et al., 2009).
- Kernel Bayes' rule (Fukumizu et al., 2011).

These provide estimates in the form of a weighted sum

$$\hat{m}_P = \sum_{i=1}^{n} w_i k(\cdot, X_i)$$

with some $w_1, \ldots, w_n \in \mathbb{R}$ and $X_1, \ldots, X_n \in \mathcal{X}$.

Proposed Method

Filtering by kernel embedding

Filtering can be cast as estimation of posterior kernel mean:

$$m_{x_t|y_{1:t}} := \int k_{\mathcal{X}}(\cdot, x_t) p(x_t|y_{1:t}) dx_t.$$

This is to be done under the following assumptions:

- transition model $p(x_t|x_{t-1})$: known (same as PF).
- observation model $p(y_t|x_t)$: unknown. But training samples are given:

$$(X_1, Y_1), \ldots, (X_n, Y_n) \subset \mathcal{X} \times \mathcal{Y}.$$

Define kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ on $\mathcal{X}$ and $\mathcal{Y}$, respectively.

($\mathcal{X}$ and $\mathcal{Y}$ can be any spaces, once kernels are defined.)

Proposed method

Suppose the posterior kernel mean at time $t-1$

$$m_{x_{t-1}|y_{1:t-1}} := \int k_{\mathcal{X}}(\cdot, x_{t-1}) p(x_{t-1}|y_{1:t-1}) dx_{t-1}.$$

was already estimate.

At time t, do the following steps:

1. Prediction: estimate the prior kernel mean:

$$m_{x_t|y_{1:t-1}} := \int k_{\mathcal{X}}(\cdot, x_t) p(x_t|y_{1:t-1}) dx_t$$

2. Correction: given y_t, estimate the posterior kernel mean:

$$m_{x_t|y_{1:t}} := \int k_{\mathcal{X}}(\cdot, x_t) p(x_t|y_{1:t}) dx_t.$$

1. Prediction step

Suppose $m_{x_{t-1}|y_{1:t-1}}$ was estimated as

$$\hat{m}_{x_{t-1}|y_{1:t-1}} := \sum_{i=1}^{n} w_{t-1,i} k_{\mathcal{X}}(\cdot, X_{t-1,i}).$$

Estimate the prior kernel mean $m_{x_t|y_{1:t-1}}$ by sampling:

$$X_{t,i} \sim \underbrace{p(x_t|x_{t-1} = X_{t-1,i})}_{transition\ \ model}, \quad (i = 1, ..., n),$$

$$\hat{m}_{x_t|y_{1:t-1}} := \sum_{i=1}^{n} w_{t-1,i} k_{\mathcal{X}}(\cdot, X_{t,i}).$$

1. Prediction step: theoretical analysis

Under some assumptions, the error of the prediction step is

$$\begin{aligned} & \|m_{x_t|y_{t-1}} - \hat{m}_{x_t|y_{1:t-1}}\|^2_{\mathcal{H}_\mathcal{X}} \\ = \quad & O(\underbrace{\|m_{x_{t-1}|y_{1:t-1}} - \hat{m}_{x_{t-1}|y_{1:t-1}}\|^2_{\mathcal{H}_\mathcal{X}}}_{(a)} + \underbrace{\sum_{i=1}^{n} w^2_{t-1,i}}_{(b)}) \end{aligned}$$

(a): error at $t-1$.

(b): inverse of effective sample size: close to 0 when the variance of the weights are small.

e.g. if $w_{t-1,i} = 1/n$, (b) $= 1/n$.

2. Correction step

Given y_t, estimate the posterior kernel mean $m_{x_t|y_{1:t}}$ by the kernel Bayes' rule (Fukumizu et al., 2011):

$$\hat{m}_{x_t|y_{1:t}} := \sum_{i=1}^{n} w_{t,i} k_{\mathcal{X}}(\cdot, X_i),$$

where $w_{t,i}$ are calculated from y_t, $\underbrace{\hat{m}_{x_t|y_{1:t-1}}}_{\textit{prior}}$ and $\underbrace{\{(X_i, Y_i)\}_{i=1}^{n}}_{\textit{trainings samples}}$:

$$\mathbf{m} := (\hat{m}_{x_t|y_{1:t-1}}(X_j)) \in \mathbb{R}^n, \quad \mathbf{k}_Y := (k_{\mathcal{Y}}(y_t, Y_j)) \in \mathbb{R}^n$$
$$G_X = (k_{\mathcal{X}}(X_i, X_j)) \in \mathbb{R}^{n\times n}, \quad G_Y = (k_{\mathcal{Y}}(Y_i, Y_j)) \in \mathbb{R}^{n\times n}$$
$$\Lambda := \mathrm{diag}((G_X + n\varepsilon I_n)^{-1}\mathbf{m}) \in \mathbb{R}^{n\times n}$$
$$w_t := \Lambda G_Y((\Lambda G_Y)^2 + \delta I_n)^{-1}\Lambda \mathbf{k}_Y \in \mathbb{R}^n,$$

This is a consistent estimator of $m_{x_t|y_{1:t}}$ (Fukumizu et al., 2013).

State estimation

The estimate $\hat{m}_{x_t|y_{1:t}} = \sum_{i=1}^{n} w_{t,i} k_{\mathcal{X}}(\cdot, X_i)$ can be used for estimating the expectation of a smooth function (Kanagawa and Fukumizu, 2014):

$$\int f(x_t) p(x_t|y_{1:t}) dx_t.$$

This is done by

$$\sum_{i=1}^{n} w_{t,i} f(X_i).$$

e.g. posterior mean $\int x_t p(x_t|y_{1:t}) dx_t$ can be estimated by

$$\sum_{i=1}^{n} w_{t,i} X_i.$$

Experiments

Setup

We compare with the following methods:

kNN-PF (Vlassis et al., 2002): the observation model is learned with k-NN approach, and then combined with a particle filter.

GP-PF (Ferris et al., 2006): the observation model is learned with with Gaussian process regression, and then combined with a particle filter.

KBR filter (Fukumizu et al., 2011): kernel embedding-based filter that also learns the transition model from data as well as the observation model. Used as baseline.

State estimation is done by estimation of posterior mean.

Synthetic experiment 1

transition model:

$$x_1 = v_1, \quad v_1 \sim \mathbb{N}(0, 1/(1-0.9^2)).$$
$$x_t = 0.9x_{t-1} + 0.5u_t + 0.5v_t, \quad v_t \sim \mathbb{N}(0, 1).$$

observation model:

$$y_t = x_t + w_t, \quad w_t \sim \mathbb{N}(0, 1).$$

- Linear Gaussian model.
- Control u_t are generated randomly generated from Gaussian $\mathbb{N}(0, 1)$.
- Training samples $\{(X_i, Y_i)\}$ are generated by running the model.

Synthetic experiment 1

- GP-PF performed the best, since the noise is Gaussian.

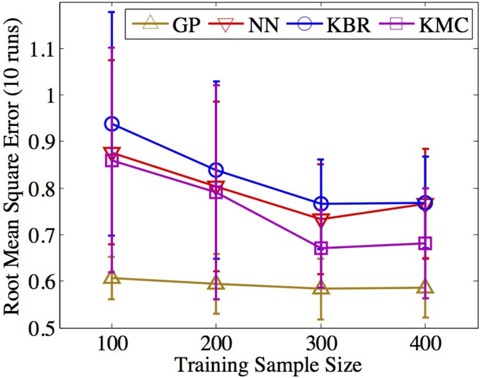


Figure: RMSE of state estimation, varying the training data size. (KMC) proposed method. (KBR) KBR filter. (NN) kNN-PF. (GP) GP-PF.

Synthetic experiment 2

transition model:

$$x_1 = v_1, \quad v_1 \sim \mathbb{N}(0, 1/(1 - 0.9^2)).$$
$$x_t = 0.9x_{t-1} + 0.5u_t + 0.5v_t, \quad v_t \sim \mathbb{N}(0, 1).$$

observation model:

$$y_t = 0.5 \exp(x_t/2)w_t, \quad w_t \sim \mathbb{N}(0, 1).$$

- Transition model is the same as Experiment 1.
- Observation model is nonlinear-transformation + multiplicative noise.
- Control u_t are generated randomly generated from Gaussian $\mathbb{N}(0, 1)$.

Synthetic experiment 2

- Proposed method performed the best, due to strong nonlinearity of the observation model.

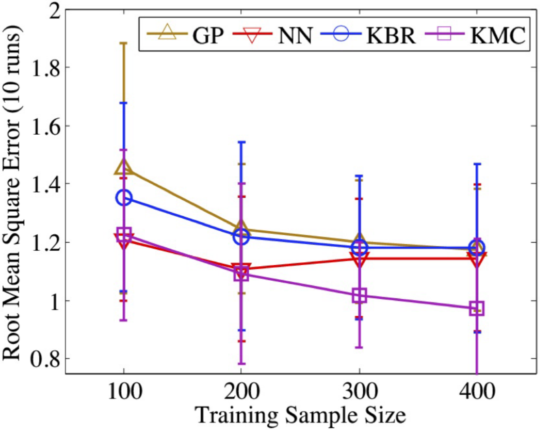
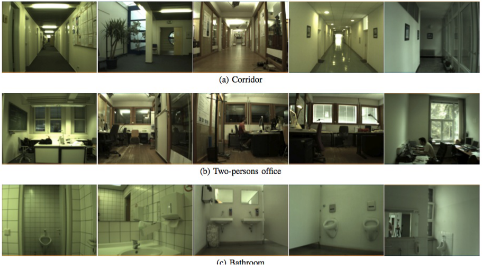


Figure: RMSE of state estimation, varying the training data size. (KMC) proposed method. (KBR) KBR filter. (NN) kNN-PF. (GP) GP-PF.

Vision-based mobile robot localization

Problem: estimate a mobile robot's position x_t for each time t, only given its vision images $y_1, \ldots, y_t$.



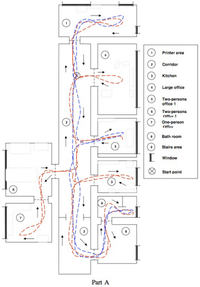


Figure: COLD database (Pronobis and Caputo, 2009)

Vision-based mobile robot localization

Experiment using the COLD database
(Pronobis and Caputo, 2009).

- state x_t (position): Gaussian kernel $k_{\mathcal{X}}$.
- observation y_t (image): Spatial pyramid kernel $k_{\mathcal{Y}}$ (Lazebnik et al., 2006).
- Training samples $\{(X_i, Y_i)\}_{i=1}^n$: position-image pairs.
- Transition model $p(x_t|x_{t-1}, u_t)$: odometry motion model (Thrun et al., 2005).

Position is estimated as a sample point with maximum weight:

$$\arg\max_{X_i} w_{t,i}$$

Result

- Proposed method performed best: even superior to kNN-PF, which was originally proposed to this task.

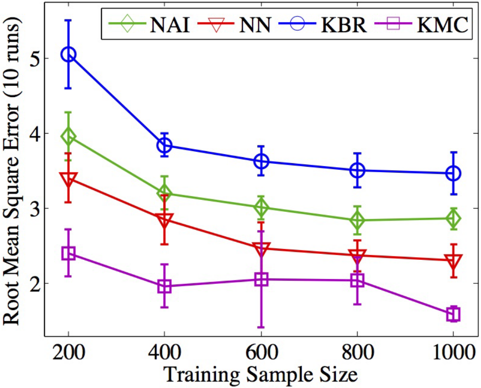


Figure: RMSE of state estimation, varying the training data size. (KMC) Proposed method, (KBR) KBR filter, (NN) kNN PF, (NAI) Baseline

Conclusion

We developed a filter for the setting where

- transition model $p(x_t|x_{t-1})$ is known.
- observation model $p(y_t|x_t)$ is unknown, but training samples

$$(X_1, Y_1), \ldots, (X_n, Y_n)$$

  are given.

Our method can be applied if

- kernels are defined on $\mathcal{X}$ and $\mathcal{Y}$.

Doucet, A., Freitas, N. D., and Gordon, N. J., editors (2001).
Sequential Monte Carlo Methods in Practice.
Springer.

Ferris, B., Hähnel, D., and Fox, D. (2006).
Gaussian processes for signal strength-based location estimation.
In *Proc. Robotics: Science and Systems*.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2008).
Kernel measures of conditional dependence.
In *Advances in NIPS 20*, pages 489–496.

Fukumizu, K., Song, L., and Gretton, A. (2011).
Kernel Bayes' rule.
In *Advances in NIPS 24*, pages 1737–1745.

Fukumizu, K., Song, L., and Gretton, A. (2013).
Kernel Bayes' rule: Bayesian inference with positive definite kernels.
Journal of Machine Learning Research, 14:3753–3783.

Jasra, A., Singh, S. S., Martin, J. S., and McCoy, E. (2012).
Filtering via approximate Bayesian computation.
Statistics and Computing, 22:1223–1237.

Kanagawa, M. and Fukumizu, K. (2014).
Recovering distributions from gaussian rkhs embeddings.
In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS 2014)*, pages 457–465.

Kanagawa, M., Nishiyama, Y., Gretton, A., and Fukumizu, K. (2014).
Monte carlo filtering using kernel embedding of distributions.
In *Proc. 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, pages xx–xx.

Lazebnik, S., Schmid, C., and Ponce, J. (2006).
Beyond bags of features: spatial pyramid matching for recognizing natural scene categories.
In *Proc. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178.

Pistohl, T., Ball, T., Schulze-Bonhage, A., Aertsen, A., and Mehring, C. (2008).
Prediction of arm movement trajectories from ecog-recordings in humans.
Journal of Neuroscience Methods, 167:105–114.

Pronobis, A. and Caputo, B. (2009).
COLD: COsy Localization Database.
Intern. J. Robotics Research, 28(5):588–594.

Schölkopf, B. and Smola, A. J. (2002).
Learning with Kernels.
MIT Press.

Smola, A., Gretton, A., Song, L., and Schölkopf, B. (2007).
A Hilbert space embedding for distributions.
In *Proc. 18th Intern. Conf. on Algorithmic Learning Theory*, pages 13–31.

Song, L., Huang, J., Smola, A., and Fukumizu, K. (2009).

Hilbert space embeddings of conditional distributions with applications to dynamical systems.
In *Proc. 26th ICML*, pages 961–968.

Thrun, S., Burgard, W., and Fox, D. (2005).
Probabilistic Robotics.
MIT Press.

Vlassis, N., Terwijn, B., and Kröse, B. (2002).
Auxiliary particle filter robot localization from high-dimensional sensor observations.
In *Proc. Intern. Conf. on Robotics and Automation (ICRA)*, pages 7–12.

Wang, Z., Ji, Q., Miller, K. J., and Schalk, G. (2011).
Prior knowledge improves decoding of finger flexion from electrocorticographic signals.
Frontiers in Neuroscience, 5:127.