Triadic Split-Merge Sampler

Anne C. van Rossum[1], Hai Xiang Lin[2,3], Johan Dubbeldam[2], H. Jaap van den Herik[3]

[1]Crownstone B.V., Rotterdam, the Netherlands;
[2]Delft University of Technology, Delft, the Netherlands;
[3]Leiden University, Leiden, the Netherlands

ABSTRACT

In machine vision typical heuristic methods to extract parameterized objects out of raw data points are the Hough transform and RANSAC. Bayesian models carry the promise to optimally extract such parameterized objects given a correct definition of the model and the type of noise at hand. A category of solvers for Bayesian models are Markov chain Monte Carlo methods. Naive implementations of MCMC methods suffer from slow convergence in machine vision due to the complexity of the parameter space. Towards this blocked Gibbs and split-merge samplers have been developed that assign multiple data points to clusters at once. In this paper we introduce a new split-merge sampler, the triadic split-merge sampler, that perform steps between two and three randomly chosen clusters. This has two advantages. First, it reduces the asymmetry between the split and merge steps. Second, it is able to propose a new cluster that is composed out of data points from two different clusters. Both advantages speed up convergence which we demonstrate on a line extraction problem. We show that the triadic split-merge sampler outperforms the conventional split-merge sampler. Although this new MCMC sampler is demonstrated in this machine vision context, its application extend to the very general domain of statistical inference.

Keywords: Bayesian statistics, MCMC, Split-Merge Sampler

1. INTRODUCTION

In machine vision many heuristic methods have been developed to extract parameterized objects out of raw data points or colloquially called point clouds. Such points are typically the output of depth sensors. To be able to make sense of this type of data is relevant to autonomous robots, self-driving cars, quadcopters, and all kind of intelligent machines that manoeuvre in a world where colliding with volumetric, solid objects means damage or worse. For the particular problem of determining line parameters from individual points the Hough transform and the RANSAC method have been extensively used in machine vision. However, through recent advances in statistical inference it is no longer necessary to rely on heuristic methods like these. A full Bayesian approach using a Dirichlet Process as a prior is nowadays computationally feasible.

We will consider a particular mixture model in which a Dirichlet Process (DP) prior is placed on the distribution over parameters G. The form of this model is:

$$\begin{aligned} y_i|\theta_i &\sim F(\theta_i) \\ \theta_i|G &\sim G \\ G &\sim DP(H,\alpha)\,. \end{aligned} \tag{1}$$

Observations y_i are related to parameters θ_i according to the likelihood function $F(\theta_i)$. The parameters θ_i are sampled from a (continuous) base distribution H (with additional hyperparameters, which are not shown). The function of the Dirichlet Process is to sample exactly the same value from this continuous distribution H multiple times: the distribution becomes almost surely discrete. The concentration parameter α, a scalar, controls the discretization effect. The lower α the more the same values will be sampled, the higher α the more the distribution will resemble H itself. In computer vision, to cluster points across lines, F defines the likelihood of data point y_i belonging to a particular line with parameters θ_i. The same θ_i is sampled almost surely multiple times for other data points from a base distribution H, a Gaussian over slope and intercept values for example.

We introduce in this paper a new sampler, the triadic sampler, that speeds up sampling from a mixture model, such as the line inference problem at hand.

In our methods (Sect. 2) we describe a conventional split-merge sampler (Sect. 2.1), that we will refer to as the dyadic sampler. We will briefly review the Metropolis-Hastings acceptance ratios from the literature for the split (Sect. 2.1.1) and merge step (Sect. 2.1.2). We treat one variant on the dyadic sampler that shows improved convergence, called the Sequentially-Allocated Merge-Split (SAMS) sampler (Sect. 2.2). We introduce our triadic split-merge sampler (Sect. 2.3) and calculate the Metropolis-Hastings acceptance ratios for the split (Sect. 2.3.1) and merge steps (Sect. 2.3.2) for this sampler. We implement and compare the dyadic and triadic samplers and an additional auxiliary variable sampler from the literature (Sect. 3). In the discussion we hint on ways to improve the sampling methods even more (Sect. 4).

2. METHODS

2.1 Conventional split-merge sampler

The conventional split-merge sampler [1] splits a single cluster into two clusters, and merges two clusters into a single cluster. Hence, this split-merge sampler operates on two clusters at each time step, for which reason we will call it a dyadic split-merge sampler in contrast with our sampler, which we will refer to as a triadic sampler. The algorithm proceeds as follows: (1) Two data points are sampled random uniformly and distinct from each other. (2) These points are sampled from the same cluster or different clusters. (3) If the data points are from the same cluster, a split probability is calculated. If the data points are from different clusters a merge probability is calculated. (4) A simple random split or other split and corresponding merge method suggest a potential reassignment of data points in the clusters under consideration. (5) The actual probabilities are calculated using the equations to be described in Sect. 2.1.1 and 2.1.2. (6) The probabilities are used in a Metropolis-Hastings step to accept or reject the split or merge proposal at hand.

2.1.1 Acceptance for the split step

The acceptance ratio contains the Metropolis ratio to step from c to c':

$$\frac{P(c')L(c'|y)}{P(c)L(c|y)} \,. \tag{2}$$

Additionally, the Hastings correction $q(c|c')/q(c'|c)$ is applied because of the asymmetry of the proposal distribution. This yields an acceptance ratio of:

$$a_{split}(c^{(2)}, c^{(1)}) = \min\left[1, \frac{q(c^{(1)}|c^{(2)})}{q(c^{(2)}|c^{(1)})}\frac{P(c^{(2)})}{P(c^{(1)})}\frac{L(c^{(2)}|y)}{L(c^{(1)}|y)}\right] \,. \tag{3}$$

The notation $c^{(2)}$ is used to indicate that the cluster index vector is referencing 2 unique clusters (in this case after the split step).

The prior distribution is represented through de Finetti's theorem [2] by a Chinese Restaurant Process [3] with concentration parameter α and no discount factor. Data not yet assigned is assigned with probability $\alpha/(n+\alpha)$ to a new cluster and with probability $n_c/(n+\alpha)$ to an existing cluster c. Here n are the total number of assigned data points, n_c are the number of data points assigned to cluster c. There are D clusters. Hence, the prior over clusters:

$$P(c) = \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)}\alpha^D\prod_{c_l}\Gamma(n_{c_l}) = \alpha^D\frac{\prod_{c_l}(n_{c_l}-1)!}{\prod_{k=1}^{n}(\alpha+k-1)} \,. \tag{4}$$

In the prior distribution ratio before and after the split step many of the factors drop out. There is one factor α remaining and the number of data points in the splitted cluster is part of the equation. There is no dependency on other clusters or the total number of data points. We simplify the formula using the Beta function $B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ with $\Gamma(x) = (x-1)!$ the Euler Gamma function.

$$\frac{P(c^{(2)})}{P(c^{(1)})} = \alpha\frac{(n_{c_i^{(2)}}-1)!(n_{c_j^{(2)}}-1)!}{(n_{c_i^{(1)}}-1)!} = \alpha B(n_{c_i^{(2)}}, n_{c_j^{(2)}}) \tag{5}$$

The likelihood can be written as a product over all observations y_i or as a product over clusters with each cluster a product over its observations y_k:

$$L(c|y) = \prod_{c=1}^{D} \prod_{k:c_k=c} p(y_k|\theta_k)\,. \tag{6}$$

Here we write $p(y_k|\theta_k)$ rather than assuming conjugacy between the likelihood $F(\theta_k)$ and the prior distribution $H(\theta_k)$ [4]. (In the case of conjugacy we can analytically calculate $\int F(\theta_k)dH(\theta_k)$ which speeds up inference, but which restricts our choice of likelihoods and priors). The likelihood ratio becomes:

$$\frac{L(c^{(2)}|y)}{L(c^{(1)}|y)} = \frac{\prod_{k:c_k^{(2)}=c_i^{(2)}} p(y_k|\theta_k) \prod_{k:c_k^{(2)}=c_j^{(2)}} p(y_k|\theta_k)}{\prod_{k:c_k^{(1)}=c_i^{(1)}} p(y_k|\theta_k)}\,. \tag{7}$$

Only clusters to which the observations y_k belong are present in this equation. The others cancel out.

The split step determines the probability of a particular split. Given that already two data points are assigned to distinct clusters, only the remaining ones have to be assigned with equal probability to $c_i^{(2)}$ and $c_j^{(2)}$:

$$q(c^{(2)}|c^{(1)}) = \left(\frac{1}{2}\right)^{-2+n_{c_i^{(2)}}+n_{c_j^{(2)}}} = \left(\frac{1}{2}\right)^{-2+n_{c_i^{(1)}}}\,. \tag{8}$$

The probability of the reverse of the split operation is exactly 1. There is only one way in which a single cluster could have risen from a split cluster, hence:

$$\frac{q(c^{(1)}|c^{(2)})}{q(c^{(2)}|c^{(1)})} = \frac{1}{\left(\frac{1}{2}\right)^{n_{c_i^{(2)}}+n_{c_j^{(2)}}-2}} = 2^{-2+n_{c_i^{(1)}}}\,. \tag{9}$$

Only basic identies are used and the fact that the number of data items does not change after a split, $n_{c_i^{(2)}}+n_{c_j^{(2)}} = n_{c_i^{(1)}}$.

2.1.2 Acceptance for the merge step

Acceptance of a merge step consists of the same components as that of the split step. For example for a_{merge}:

$$a_{merge}(c^{(1)}, c^{(2)}) = \min\left[1, \frac{q(c^{(2)}|c^{(1)})}{q(c^{(1)}|c^{(2)})} \frac{P(c^{(1)})}{P(c^{(2)})} \frac{L(c^{(1)}|y)}{L(c^{(2)}|y)}\right]\,, \tag{10}$$

The ratios of the other equations in the merge step are similarly, the inverse of the ratios of the split step.

2.2 Sequentially-Allocated Merge-Split sampler

A variant on the conventional split-merge sampler is the Sequentially Allocated Merge-Split* (SAMS) sampler [5]. In the fourth step described at Sect. 2.1, it replaces a simple random split procedure by one that sequentially assigns observations to clusters, using the likelihood:

procedure SAMS(S, c_0, c_1) ▷ Accepts unassigned S, cluster indices c_i, and $p(y_k|\theta_{c_i})$ with $i = 0, 1$, returns cluster indices c'_m.
  T = random_shuffle(S)
  for all $m \in T$ **do**
    $p(c_m = c_0|c_0, c_1, \theta_{c_0}, \theta_{c_1}) = \frac{N_0 p(y_k|\theta_0)}{N_0 p(y_k|\theta_0)+N_1 p(y_k|\theta_1)}$
    $p(c_m = c_1|c_0, c_1, \theta_{c_0}, \theta_{c_1}) = 1 - p(c_m = c_0|c_0, c_1, \theta_{c_0}, \theta_{c_1})$
    $c'_m \sim p(c_m|c_0, c_1, \theta_{c_0}, \theta_{c_1})$
  end for
  return c'_m, the cluster assignment for S.
end procedure

In contrast to the simple random split, observations y_k are used in the SAMS to obtain cluster assignments that correspond with the data rather than cluster assignments independent of the data.

*In the naming of split-merge or merge-split samplers, the order of merge split does not bear any significance.

2.3 Triadic split-merge sampler

The triadic split-merge sampler uses up to three clusters for a split or merge step. The intuition behind the triadic split-merge sampler is twofold:

- In the dyadic sampler there is a large asymmetry between split and merge steps. There is only one way in which two clusters can be merged into one single cluster, while there are many ways in which one single cluster can be split into two clusters. This asymmetry is reduced by transitioning between two and three clusters. This is a straightforward improvement in balancing split and merge steps (for alternatives, see [6]).
- In practical optimization problems it might be useful to form a third cluster out of subsets of two other clusters. The dyadic MCMC sampler requires intermediate steps in which (1) one of these clusters is split into two, (2) the other is split into two, and (3) the two new clusters are merged. This means that (a) mixing and hence convergence will be slow and (b) the intermediate steps might have very low probability and function as an unnecessary barrier between high probable states.

We have visualized the difference between the dyadic and the triadic sampler in Fig. 1.

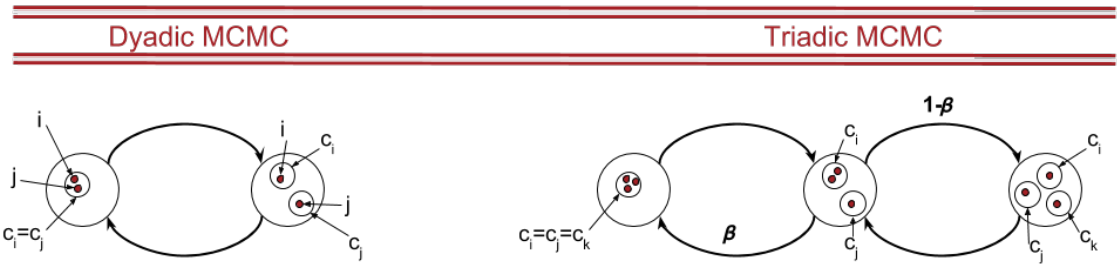

Figure 1. Right: dyadic MCMC picks two data items i, j random uniformly. If both are in the same cluster, a split towards two clusters is attempted. If both are in distinct clusters, a merge towards one cluster is attempted. Left: triadic MCMC picks three data items i, j, k random uniformly. If all three are in the same cluster, a split towards two clusters is attempted. If the three items are in two clusters, either a split into three (with probability $1 - \beta$) or a merge into a single cluster (with probability β) is attempted. If the three data items are in three distinct clusters, a merge is attempted. There are no direct transitions from a single cluster to three clusters or the other way around.

The triadic split-merge sampler in algorithmic form:

procedure TRIADIC SPLIT-MERGE SAMPLER(c) ▷ Accepts cluster assignments c of length N (besides Metropolis-Hastings acceptance factors $a(c', c)$ and a split procedure) and returns a (potentially) updated cluster assignment vector c'.
  $i \sim DU(\{1, \ldots, N\})$ ▷ Sample i random uniformly over cluster assignments.
  $j \sim DU(\{1, \ldots, N\} \cap i)$ ▷ Sample j also random uniformly, but with $j \neq i$.
  $k \sim DU(\{1, \ldots, N\} \cap \{i, j\})$ ▷ Sample k random uniformly, but with $k \neq j$, $k \neq i$.
  $S_R = \{c_i, c_j, c_k\}$ ▷ Sampled cluster indices c_i, c_j, c_k.
  $S_I = \{c_x\}$ with $c_x \in S_R$ for $x \in \{1, \ldots, N\}$ ▷ All data with same cluster indices c_i, c_j, c_k.
  $S_E = S_I \cap S_R$ ▷ All data in clusters c_i, c_j, c_k excluding S_R.
  $N_S = \text{unique}(S_R)$
  $u \sim U(0, 1)$ ▷ Sample u between 0 or 1 uniformly.
  if $N_S = 1$ **then** ▷ Case: i, j, k belong to the same cluster.
    return $c' = \text{DYADIC SPLIT-MERGE SAMPER}(c)$
  else if $N_S = 2$ **and** $u < \beta$ **then** ▷ Case: a cluster with one item and one with two items and $u < \beta$.
    return $c' = \text{DYADIC SPLIT-MERGE SAMPER}(c)$
  else if $N_S = 2$ **and** $u \geq \beta$ **then** ▷ Case: a cluster with one item and one with two items and $u \geq \beta$.
    $c_i^{(3)} = c_k$ with $c_k \notin \{c_1, \ldots, c_N\}$ ▷ Sample new cluster for $c_i^{(3)}$.
    $c_j^{(3)} = c_j^{(2)}$ ▷ Keep c_j the same.
    $c_e^{(3)} = \text{SPLITPROCEDURE}(S_E, c_i^{(3)}, c_j^{(3)})$ ▷ After $c_i^{(3)}, c_j^{(3)}$ assign S_E.

for all $m \notin S_I$ **do**
$c_m^{(3)} = c_m^{(2)}$ ▷ Data points in clusters other than c_i, c_j are not adjusted.
end for
$c' = \{c_i^{(3)}, c_j^{(3)}, c_e^{(3)}, c_m^{(3)}\}$
$a = a_{split}(c', c)$ according to Eq. 11 ▷ MH acceptance for a split.
else ▷ Case: i, j, k belong to thee different clusters $c_i \neq c_j \neq c_k$ ($N_S = 3$).
$S_L = S_I \cap \{c_i^{(3)}, c_j^{(3)}\}$ ▷ Data in clusters c_i, c_j, c_k except for i and j itself.
$\{c_i^{(2)}, c_j^{(2)}\} = \text{SAMS}(S_L, c_i^{(3)}, c_j^{(3)})$ ▷ Assign data points in c_i, c_j, c_k to c_i, c_j.
for all $m \notin S_L$ **do**
$c_m^{(2)} = c_m^{(3)}$ ▷ Data points in clusters other than S_L are not adjusted.
end for
$c' = \{c_i^{(2)}, c_j^{(2)}, c_m^{(2)}\}$
$a = a_{merge}(c', c)$ according to Eq. 18 ▷ MH acceptance for a merge.
end if
$u \sim U(0, 1)$ ▷ Sample u between 0 or 1 uniformly.
if $a < u$ **then**
$c' = c$ ▷ Reject c' by setting it to c
end if
return c', the (updated) cluster assignment vector: $c \to c'$.
end procedure

2.3.1 Acceptance for the split step

In the triadic split-merge sampler there are two splitting steps. It is possible to split according to the dyadic split-merge sampler. However, given two clusters there are (split) jumps to three states as well as (merge) jumps to single states again. To account for this asymmetry another Hastings correction is applied to establish detailed balance.

$$a_{split}(c^{(2)}, c^{(1)}) = \min\left[1, \frac{r(c^{(1)}|c^{(2)})}{r(c^{(2)}|c^{(1)})} \frac{q(c^{(1)}|c^{(2)})}{q(c^{(2)}|c^{(1)})} \frac{P(c^{(2)})}{P(c^{(1)})} \frac{L(c^{(2)}|y)}{L(c^{(1)}|y)}\right]. \tag{11}$$

Here we have one additional term compared to the split step from one cluster to two clusters:

$$\frac{r(c^{(1)}|c^{(2)})}{r(c^{(2)}|c^{(1)})} = \frac{\beta}{1}. \tag{12}$$

The parameter β is free to control, as long as $0 < \beta < 1$ (to maintain ergodicity). The transition from two states to three states is another split step:

$$a_{split}(c^{(3)}, c^{(2)}) = \min\left[1, \frac{r(c^{(2)}|c^{(3)})}{r(c^{(3)}|c^{(2)})} \frac{q(c^{(2)}|c^{(3)})}{q(c^{(3)}|c^{(2)})} \frac{P(c^{(3)})}{P(c^{(2)})} \frac{L(c^{(3)}|y)}{L(c^{(2)}|y)}\right]. \tag{13}$$

The fraction with r:

$$\frac{r(c^{(2)}|c^{(3)})}{r(c^{(3)}|c^{(2)})} = \frac{1}{1-\beta}. \tag{14}$$

The fraction with q uses the total number of data points n_c in the clusters:

$$\frac{q(c^{(2)}|c^{(3)})}{q(c^{(3)}|c^{(2)})} = \frac{\left(\frac{1}{2}\right)^{n_c-2}}{\left(\frac{1}{3}\right)^{n_c-3}} = \left(3^{n_c-3}\right)\left(2^{2-n_c}\right) = \left(\frac{3}{2}\right)^{n_c} \frac{2^2}{3^3}. \tag{15}$$

To move from 2 clusters to 3 clusters the probability is a $1/3$ for each cluster index in vector c (except for the three data items already selected randomly, hence $n_c - 3$). To move back, the probability is a $1/2$ and there are only two data items randomly assigned beforehand. The fraction with P uses the number of data points in each of the clusters before and after the step:

$$\frac{P(c^{(3)})}{P(c^{(2)})} = \alpha \frac{(n_{c_i^{(3)}} - 1)!(n_{c_j^{(3)}} - 1)!(n_{c_k^{(3)}} - 1)!}{(n_{c_i^{(2)}} - 1)!(n_{c_j^{(2)}} - 1)!} = \alpha \frac{B(n_{c_i^3}, n_{c_j^3}, n_{c_k^3})}{B(n_{c_i^2}, n_{c_j^2})}. \tag{16}$$

Here we introduced a generalized Beta function $B(a,b,c) = \Gamma(a)\Gamma(b)\Gamma(c)/\Gamma(a+b+c)$ with Γ again the Euler Gamma function. The likelihood ratio becomes:

$$\frac{L(c^{(3)}|y)}{L(c^{(2)}|y)} = \frac{\prod_{m:c_m^{(3)}=c_i^{(3)}} p(y_m|\phi) \prod_{m:c_m^{(3)}=c_j^{(3)}} p(y_m|\phi) \prod_{m:c_m^{(3)}=c_k^{(3)}} p(y_m|\phi)}{\prod_{m:c_m^{(2)}=c_i^{(2)}} p(y_m|\phi) \prod_{m:c_m^{(2)}=c_j^{(2)}} p(y_m|\phi)} . \tag{17}$$

2.3.2 Acceptance for the merge step

The merge step from two to one cluster is analogous to the split step:

$$a_{merge}(c^{(1)}, c^{(2)}) = \min\left[1, \frac{r(c^{(2)}|c^{(1)})}{r(c^{(1)}|c^{(2)})} \frac{q(c^{(2)}|c^{(1)})}{q(c^{(1)}|c^{(2)})} \frac{P(c^{(1)})}{P(c^{(2)})} \frac{L(c^{(1)}|y)}{L(c^{(2)}|y)}\right] . \tag{18}$$

The merge step from three clusters to two clusters is:

$$a_{merge}(c^{(2)}, c^{(3)}) = \min\left[1, \frac{r(c^{(3)}|c^{(2)})}{r(c^{(2)}|c^{(3)})} \frac{q(c^{(3)}|c^{(2)})}{q(c^{(2)}|c^{(3)})} \frac{P(c^{(2)})}{P(c^{(3)})} \frac{L(c^{(2)}|y)}{L(c^{(3)}|y)}\right] . \tag{19}$$

Note that all the fractions in Eq. 19 are the inverse of the fractions in Eq. 13. Hence, to calculate the merge step we can use the inverses of Eq. 14–17.

3. RESULTS

The problem we use to test our sampler is a well-known problem in computer vision, namely that of the inference of line parameters (slope and intercept) given data points. Rather than ordinary linear regression, in computer vision there is a mixture of lines that have to be estimated. Moreover, the number of lines is not known beforehand. To solve this problem we use the Dirichlet process mixture (Eq. 1) with a normal distribution $N(0, \sigma_0)$ to generate the line parameters and a likelihood function that defines points to be uniformly distributed across a line of length 20 and deviating from the line according to a normal distribution $N(0, \sigma_1)$.

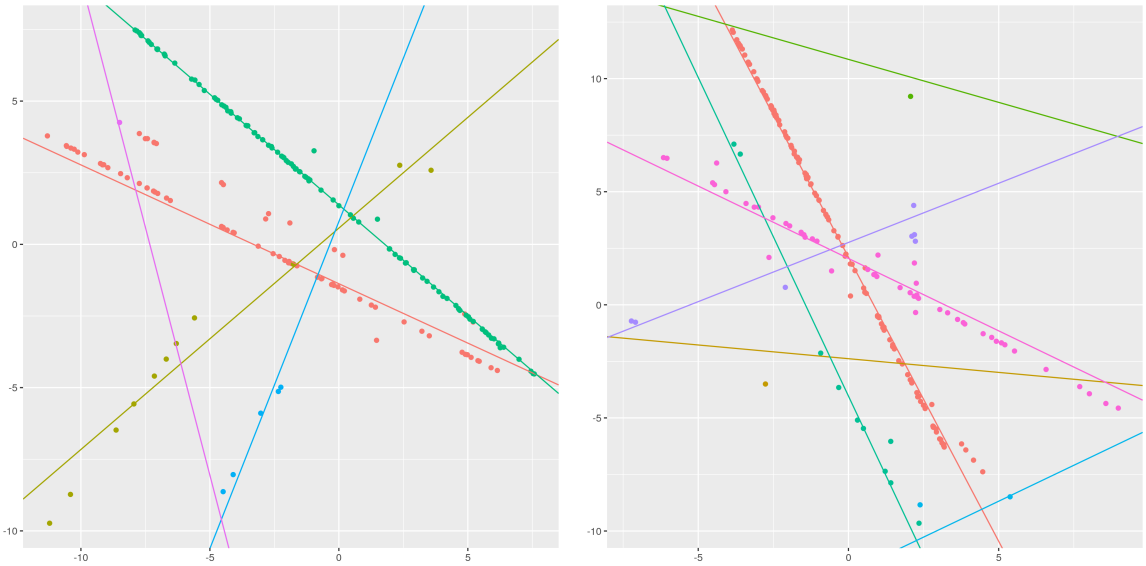

Figure 2. Two examples of fitting a mixture of lines to data items scattered over a two-dimensional space. The lines drawn are inferred using the triadic sampler. The lines are not the ground truth, but are meant to demonstrate the typical errors made by fitting methods. Note for example that there are mistakes in both the assignment of points to lines as well as the line parameters (slope and intercept). Left: In example 1 two lines with similar slope are seen as the same line. Right: In example 2 points on one vertical line are assigned to multiple lines. In example 1 and 2 slopes are not always through the points.

3.1 Implementation

The sampler is open-source[†] implemented in C++ which means that (a) it is computationally fast and (b) it can be run on embedded devices if a cross-compiler is available (and the Eigen3 library is ported). Note, that due to the fact that the simulator uses a lot of random numbers the system should use a modern compiler (g++-6 or newer) and should have enough entropy available[‡]. Rather than a random scan [7], the implementation uses a fixed scan as advocated in the literature [8].

To speed up the sampler most calculations are done in log-space. Let $v = u + 1$. The ratio with probabilities (Eq. 5 and 16) becomes:

$$\log \frac{P(c^{(v)})}{P(c^{(u)})} = \log(\alpha) + \sum_i \log \Gamma(n_{c_i^{(v)}}) - \sum_i \log \Gamma(n_{c_i^{(u)}}) \,. \tag{20}$$

The fraction with $q(\cdot)$ (Eq. 9 and 15) becomes:

$$\log \frac{q(c^{(v-1)}|c^{(v)})}{q(c^{(v)}|c^{(v-1)})} = (v - n_c - 1)\log(v-1) - (v - n_c)\log(v) \,. \tag{21}$$

The fraction with r is (Eq. 14):

$$\log \frac{r(c^{(2)}|c^{(3)})}{r(c^{(3)}|c^{(2)})} = -\log(1-\beta) \,. \tag{22}$$

The log-probability to calculate the likelihood given by a multivariate Normal distribution is well-known (not shown).

3.2 Comparison

The Triadic sampler using SAMS is compared with the Jain-Neal Dyadic sampler using SAMS and an auxiliary variable sampler with $m = 3$ (see algorithm 8 in [9]).

Method	Purity	Rand Index	Adjusted Rand Index
Dyadic sampler	0.80960	0.80580	0.56382
Auxiliary variables	0.87235	0.85879	0.68224
Triadic sampler	0.86405	0.87188	0.71067

Table 1. The purity, rand index, and adjusted rand index establishing the quality of the clustering method. The closer the values to one, the better the method performed. The purity metric assigns high values to clusters that do not have data points from other clusters (but does not penalize the number of clusters). The rand index index computes similarity between clusters taking false negatives and false positives into account. The adjusted rand index accounts for chance. The adjusted rand index is most useful in our comparison.

In Table 1 the line estimation problem is compared for the dyadic sampler, an auxiliary variables sampler, and the proposed triadic sampler. The simulation is run with $\beta = 0.1$ so that a significant number of steps are tried between two and three clusters (rather than only between one and two clusters).

[†] Code can be found at https://code.annevanrossum.nl/noparama.

[‡] On Linux this can be checked in /proc/sys/kernel/random/entropy_avail.

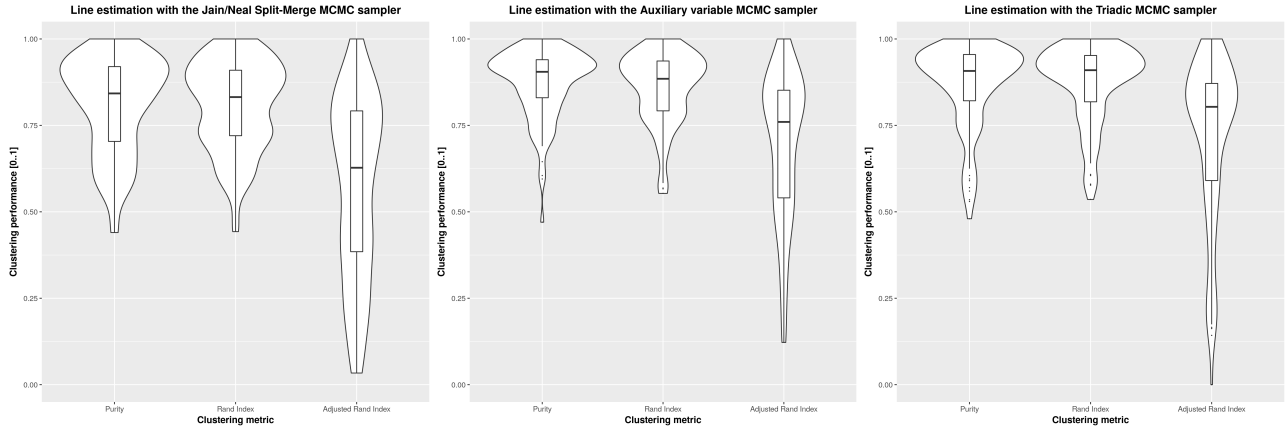

Figure 3. The same results as in Table 1, but visualized in a violin plot. The distribution over metric values are displayed in a vertical fashion. From left to right the distribution shifts up (gets closer to one), signifying better clustering performance. The clustering metrics displayed are purity, the rand index, and the adjusted rand index.

In Fig. 3 the different metrics are visualized in the form of violin plots. The improvement in clustering is especially visible with the adjusted rand index.

4. DISCUSSION

A new split-merge sampler has been introduced, implemented, and applied to the computer vision problem of line estimation. The sampler outperforms existing samplers, such as the ordinary (dyadic) split-merge sampler [1] and auxiliary variable sampler [9].

Although the proposed split-merge sampler is able to mix considerably faster through a mixture model, it does not use global jumps directly based on the data at hand. It is reasonable to suggest that MCMC methods benefit from combining the local jumps with global jumps, for example by a mixture of the local Metropolis-Hastings sampler with a Metropolized independence sampler [10].

REFERENCES

[1] S. Jain and R. M. Neal, "A Split-Merge Markov Chain Monte Carlo Procedure for the Dirichlet Process Mixture Model," Journal of Computational and Graphical Statistics **13**(1), 158–182 (2004).

[2] P. Diaconis and D. Freedman, "de Finetti's theorem for Markov chains," The Annals of Probability , 115–130 (1980).

[3] D. J. Aldous, Exchangeability and related topics, Springer (1985).

[4] D. B. Dahl, "Sequentially-Allocated Merge-Split Sampler for Conjugate and Nonconjugate Dirichlet Process Mixture Models," tech. rep., Texas A&M University (Nov. 2005).

[5] D. B. Dahl, "An Improved Merge-Split Sampler for Conjugate Dirichlet Process Mixture Models," tech. rep., University of Wisconsin–Madison (Nov. 2003).

[6] W. Wang and S. Russell, "A Smart-dumb/Dumb-smart Algorithm for Efficient Split-merge MCMC," in Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI15, 902–911, AUAI Press, Arlington, Virginia, United States (2015).

[7] S. Jain and R. M. Neal, "Splitting and Merging Components of a Nonconjugate Dirichlet Process Mixture Model," Bayesian Analysis **2**(3), 445–472 (2007).

[8] S. N. MacEachern, "Comment on "Splitting and Merging Components of a Nonconjugate Dirichlet Process Mixture Model" by Jain and Neal," Bayesian Analysis **2**(3), 483–494 (2007).

[9] R. M. Neal, "Markov Chain Sampling Methods for Dirichlet Process Mixture Models," Journal of Computational and Graphical Statistics **9**(2), 249–265 (2000).

[10] V. Jampani, S. Nowozin, M. Loper, and P. V. Gehler, "The informed sampler: A discriminative approach to bayesian inference in generative computer vision models," Computer Vision and Image Understanding **136**, 32–44 (2015).