

Трекинг при помощи фильтра частиц

Евгений Нижибицкий

10 декабря 2013 г.

- 1 Модель пространства состояний (SSM)
- 2 Методы генерации выборки
 - Rejection sampling
 - Importance sampling
 - Sequential importance sampler (SIS)
 - Resampling
- 3 Фильтр частиц
 - Sampling importance resampling (SIR)
 - Пример задачи трекинга
 - Sequential Monte Carlo (SMC)
- 4 Видеотрекинг
 - Общий план
 - Примеры моделей
 - «Видеотрекинг произвольных объектов»

1 Модель пространства состояний (SSM)

2 Методы генерации выборки

Rejection sampling

Importance sampling

Sequential importance sampler (SIS)

Resampling

3 Фильтр частиц

Sampling importance resampling (SIR)

Пример задачи трекинга

Sequential Monte Carlo (SMC)

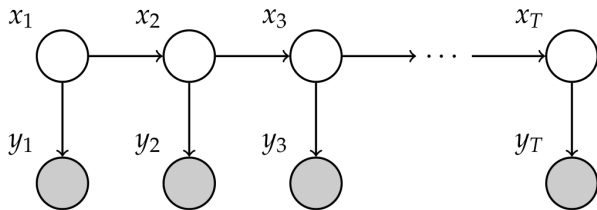
4 Видеотрекинг

Общий план

Примеры моделей

«Видеотрекинг произвольных объектов»

Модель пространства состояний (SSM)



Модель пространства состояний состоит из марковского процесса $\{x_t\}_{t \geq 1}$ и процесса измерений $\{y_t\}_{t \geq 1}$ так, что

$$\begin{aligned}x_{t+1}|x_t &\sim f_{\theta,t}(x_{t+1}|x_t, u_t), \\y_t|x_t &\sim h_{\theta,t}(y_t|x_t, u_t), \\x_1 &\sim \mu(x_1),\end{aligned}$$

где x_t отвечает за состояние, u_t — входной сигнал, y_t — наблюдаемое измерение, θ — любые неизвестные (статические) параметры.

Модель пространства состояний (SSM)

Байесовская сеть описывает совместное распределение всех задействованных переменных

$$p(x_{1:T}, y_{1:T}) = \prod_{t=1}^T p(x_t | pa(x_t)) \prod_{t=1}^T p(y_t | xpa(y_t)),$$

где за $pa(x_t)$ обозначены все родители x_t .

$$p(x_{1:T}, y_{1:T}) = \mu(x_1) \prod_{t=1}^T f_{\theta,t}(x_{t+1} | pa(x_t)) \prod_{t=1}^T h_{\theta,t}(y_{t+1} | xpa(y_t)).$$

Модель пространства состояний (SSM)

Функции плотности для наиболее часто решаемых задач вывода состояний:

Задача	Функция плотности
Фильтрация	$p(x_t y_{1:t})$
Предсказание	$p(x_{t+1} y_{1:t})$
Предсказание на k шагов	$p(x_{t+k} y_{1:t})$
Совместное сглаживание	$p(x_{1:T} y_{1:T})$
Маргинальное сглаживание	$p(x_{1:t} y_{1:T}), t \leq T$

1 Модель пространства состояний (SSM)

2 Методы генерации выборки

Rejection sampling

Importance sampling

Sequential importance sampler (SIS)

Resampling

3 Фильтр частиц

Sampling importance resampling (SIR)

Пример задачи трекинга

Sequential Monte Carlo (SMC)

4 Видеотрекинг

Общий план

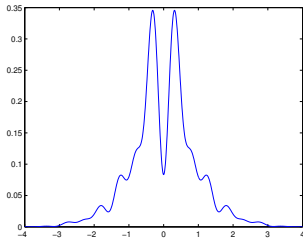
Примеры моделей

«Видеотрекинг произвольных объектов»

Rejection sampling — метод Монте-Карло, который получает независимую выборку из целевого распределения

$$\pi(z) = \frac{\tilde{\pi}(z)}{C_{\pi}},$$

где $\tilde{\pi}(z)$ можно вычислить, а C_{π} — нормировочная константа.

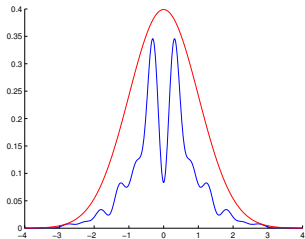


Идея: генерировать случайные числа равномерно по площади под графиком целевого распределения $\pi(z)$.

Задача так же сложна, как и начальная, но что если...

Методы генерации выборки

Rejection sampling



генерировать сэмпл \tilde{z} из предложенного распределения $q(z)$ и сэмпл $u \sim U[0, 1]$.

Сэмпл \tilde{z} тогда является целевым, если

$$u \leq \frac{\tilde{\pi}(\tilde{z})}{Bq(\tilde{z})}$$

Предположения:

- 1 Легко получать выборку из $q(z)$.
- 2 Существует константа B такая, что $\pi(z) \leq Bq(z), \forall z \in Z$.
- 3 Существенный носитель $q(z)$ включает таковой для $\pi(z)$, т.е. $q(z) > 0$ когда $\pi(z) > 0$.

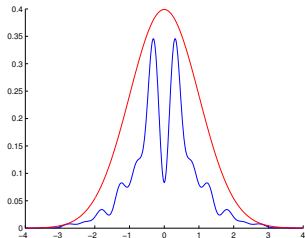
Алгоритм:

- 1 Получить $\tilde{z} \sim q(z)$.
- 2 Получить $u \sim U[0, 1]$.
- 3 Если $u \leq \frac{\tilde{\pi}(\tilde{z})}{Bq(\tilde{z})}$, то принять \tilde{z} в качестве сэмпла из $\pi(z)$ и перейти на шаг 1.
- 4 Иначе отклонить \tilde{z} и перейти на шаг 1.

Задача: сгенерировать выборку из распределения

$$\pi(z) = \frac{1}{C_\pi} e^{-\frac{1}{2}z^2} (\sin(6z)^2 + 3\cos(z)^2 \sin(4z)^2 + 1)$$

Используем rejection sampling с $q(z) = N(z|0, 1)$ и $B = 12$:



```
» disp(sum(1/12*pi(-50:0.01:50)>normpdf(-50:0.01:50,0,1)))
```

0

Алгоритм:

- 1 Сгенерировать выборку размера N $\{z^i\}_{i=1}^N$ из предложенного распределения $q(z)$.
- 2 Вычислить веса (importances)

$$\tilde{w}^i = \frac{\tilde{\pi}(z_i)}{\tilde{q}(z_i)}, i = 1, \dots, N.$$

- 3 Нормализовать веса

$$w^i = \frac{\tilde{w}^i}{\sum_{j=1}^N \tilde{w}^j}, i = 1, \dots, N.$$

Алгоритм IS не возвращает выборку целевого распределения, но выборка $\{z^i\}_{i=1}^N$ совместно с нормализованными весами $\{w^i\}_{i=1}^N$ позволяет получить эмпирическое приближение целевой функции распределения

$$\hat{\pi}(z) = \sum_{i=1}^N w^i \delta_{z^i}(z).$$

Решим задачу нелинейной фильтрации, которая состоит в вычислении $p(x_t|y_{1:t})$ для системы

$$\begin{aligned}x_{t+1}|x_t &\sim f(x_{t+1}|x_t, u_t), \\y_t|x_t &\sim h(y_t|x_t, u_t), \\x_1 &\sim \mu(x_1),\end{aligned}$$

Получаем вывод:

$$\begin{aligned}p(x_t|y_{1:t}) &= \frac{h(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}, \\p(x_t|y_{1:t-1}) &= \int f(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}.\end{aligned}$$

Реализуем решение с помощью IS.

Рассмотрим в качестве предложенной функции

$$q(x_{1:t}) = \mu(x_1) \prod_{k=1}^t f(x_k|x_{k-1})$$

Что это означает:

- В момент времени 1 генерируем $x_1 \sim \mu(x_1)$.
- В моменты времени $k = 2, \dots, t$ генерируем $x_k^i \sim f(x_k|x_{k-1}^i)$.

Это завершает шаг 1 алгоритма IS. Как теперь вычислять последовательно веса?

Алгоритм SIS для оценивания $p(x_{1:t}|y_{1:t})$:

- 1 Генерируем N начальных сэмплов $x_1^i \sim \mu(x_1)$, веса устанавливаем равными $\tilde{w}_0^i = 1/N$, $k = 0$.
- 2 Для всех k от 1 до t
 - 1 Вычисляем веса $\tilde{w}_k^i = p(y_k|x_k^i)\tilde{w}_{k-1}^i$.
 - 2 Нормализуем веса $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^N \tilde{w}_k^j}$ и сохраняем — $\{w_{1:k}^i\}_{i=1}^N = \{w_{1:k-1}^i, w_k^i\}_{i=1}^N$.
 - 3 Генерируем независимую выборку размера N , из предложенного распределения, $x_{k+1}^i \sim f(x_{k+1}|x_k^i)$, и сохраняем — $\{x_{1:k+1}^i\}_{i=1}^N = \{x_{1:k}^i, x_{k+1}^i\}_{i=1}^N$.

Рассмотрим линейную модель с гауссовым шумом

$$\begin{aligned}x_{t+1} &= x_t + v_t, & v_t &\sim \mathcal{N}(0, 0.1), \\y_t &= 0.5x_t + e_t, & e_t &\sim \mathcal{N}(0, 0.1), \\p(x_1) &= \mathcal{N}(x_1|0, 0.1).\end{aligned}$$

Применим алгоритм SIS для аппроксимации распределения

$$\hat{p}(x_t|y_{1:t}) = \sum_{i=1}^N w_t^i \delta_{x_t^i}(x_t).$$

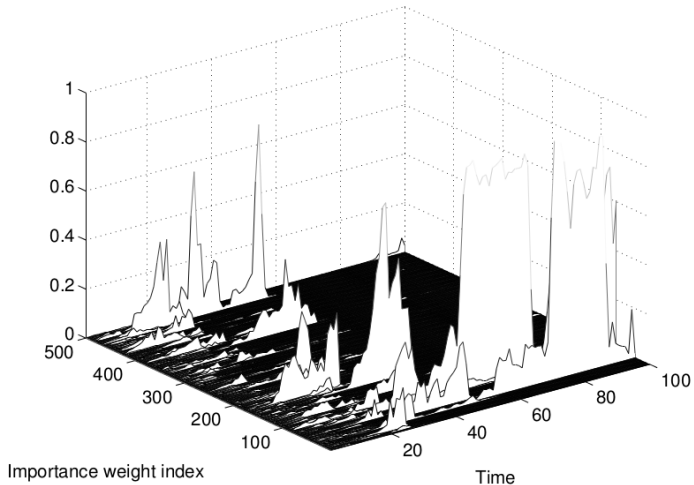
Обучаем

- Точечную оценку $\hat{x}_{t|t} = \int x_t \hat{p}(x_t|y_{1:t}) dx_t = \sum_{i=1}^N w_t^i x_t^i$.
- Веса w_t^i .

Методы генерации выборки

Sequential importance sampler (SIS)

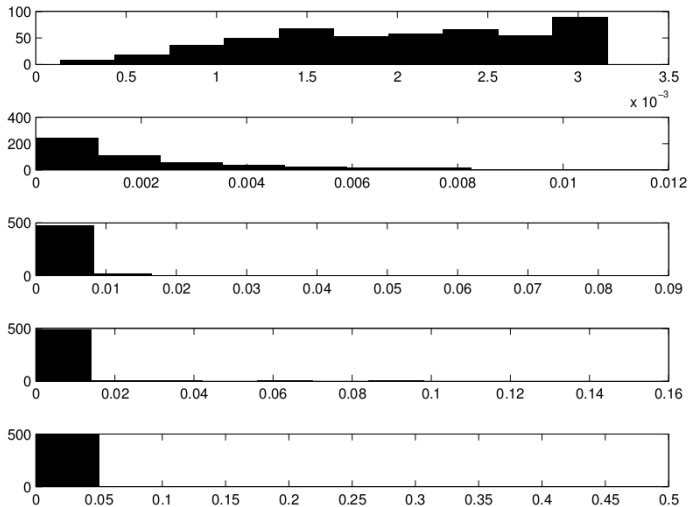
Эволюция весов во времени:



Методы генерации выборки

Sequential importance sampler (SIS)

Гистограммы весов для $t = 2, 5, 10, 20, 50$:



Что делать с вырождением? Удалим веса из представления!

SIS оценивает целевую плотность как

$$\hat{\pi}^1(z) = \sum_{i=1}^N w^i \delta_{z^i}(z).$$

Сделаем ресэмплинг с заменой — сгенерируем новую выборку z^i для всех $i = 1, \dots, N$, где

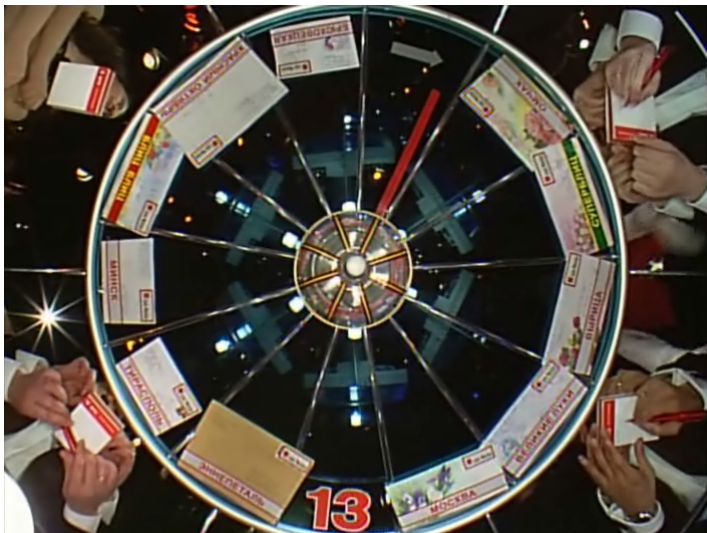
$$\mathbb{P}(z^i = \tilde{z}^j) = w^j, j = 1, \dots, N.$$

Получаем свободное от весов представление

$$\hat{\pi}^2(z) = \sum_{i=1}^N \frac{1}{N} \delta_{z^i}(z).$$

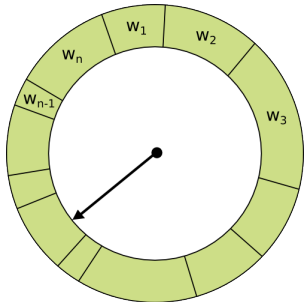
Методы генерации выборки

Resampling

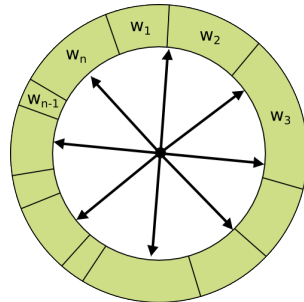


Методы генерации выборки

Resampling



- Колесо рулетки
- Бинарный поиск
- $O(n \log n)$



- Стохастическое сэмплирование
- Низкая вариативность
- $O(n)$

1 Модель пространства состояний (SSM)

2 Методы генерации выборки

Rejection sampling

Importance sampling

Sequential importance sampler (SIS)

Resampling

3 Фильтр частиц

Sampling importance resampling (SIR)

Пример задачи трекинга

Sequential Monte Carlo (SMC)

4 Видеотрекинг

Общий план

Примеры моделей

«Видеотрекинг произвольных объектов»

Алгоритм:

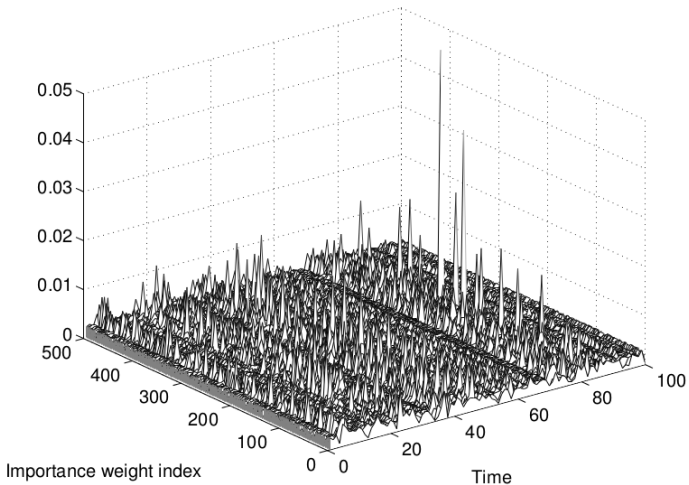
- 1 Генерируем выборку $\{\tilde{z}^i\}_{i=1}^N$ из предложенного распределения $q(z)$.
- 2 Вычисляем веса $\tilde{w}^i = \frac{\tilde{\pi}(\tilde{z}^i)}{\tilde{q}(\tilde{z}^i)}$, $i = 1, \dots, N$.
- 3 Нормализуем веса $w_k^i = \frac{\tilde{w}^i}{\sum_{j=1}^N \tilde{w}^j}$, $i = 1, \dots, N$.
- 4 Для $i = 1, \dots, N$ генерируем новый сэмпл z^i такой, что

$$\mathbb{P}(z^i = \tilde{z}^j) = w^j, j = 1, \dots, N.$$

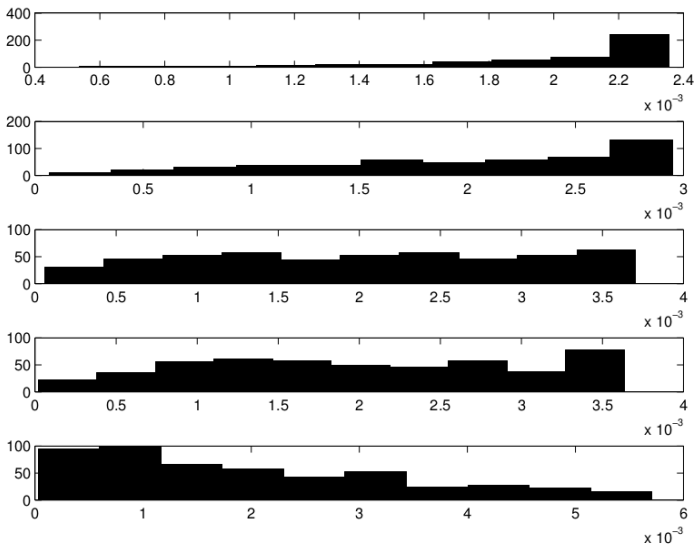
Алгоритм фильтра частиц для $p(x_{1:t}|y_{1:t})$

- 1 Генерируем N начальных сэмплов $\tilde{x}_1^i \sim \mu(x_1)$, веса устанавливаем равными $\tilde{w}_0^i = 1/N$, $k = 0$.
- 2 Для всех k от 1 до t
 - 1 Вычисляем веса $\tilde{w}_k^i = p(y_k|\tilde{x}_k^i)\tilde{w}_{k-1}^i$.
 - 2 Нормализуем веса $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^N \tilde{w}_k^j}$.
 - 3 Для $i = 1, \dots, N$ генерируем новый сэмпл x_t^i такой, что $\mathbb{P}(x_t^i = \tilde{x}_t^j) = w^j, j = 1, \dots, N$.
 - 4 Генерируем новую выборку из N элементов из предложенного распределения, $\tilde{x}_{k+1}^i \sim f(x_{k+1}|x_k^i)$.

Эволюция весов во времени:



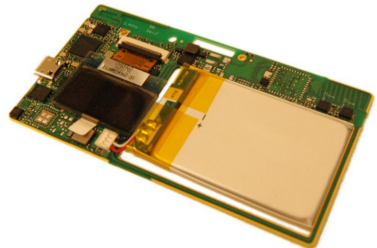
Гистограммы весов для $t = 2, 5, 10, 20, 50$:



Фильтр частиц

Пример задачи трекинга

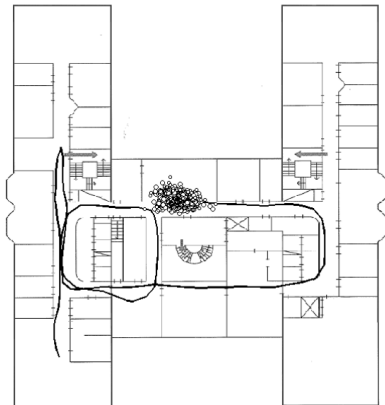
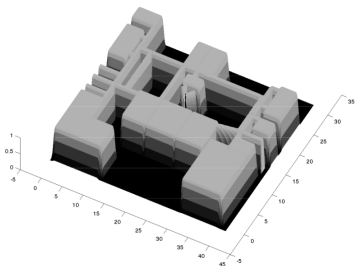
Вычислить местоположение человека, передвигающегося в помещении, на основе сенсоров в бейдже и плана здания.



Фильтр частиц

Пример задачи трекинга

Функция распределения по умолчанию для помещения и траектория человека с частицами, оценивающими последнее его местоположение.



Демонстрация

Предположим, в момент времени t имеется система частиц $\{x_{1:t}^i, w_t^i\}_{i=1}^N$, аппроксимирующая целевое распределение $p(x_{1:t}|y_{1:t})$ следующим образом:

$$\hat{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^N \frac{w_t^i}{\sum_{l=1}^N w_t^l} \delta_{x_{1:t}^i}(x_{1:t}).$$

Тогда три шага алгоритма SMC будут такими:

- 1 Ресэмплинг. $\{x_{1:t}, w_t^i\}_{i=1}^N \rightarrow \{\tilde{x}_{1:t}, 1/N\}_{i=1}^N$.
- 2 Получение новых сэмплов. $x_{t+1}^i \sim Q_t(x_{t+1}|\tilde{x}_{1:t}^i)$,
 $i = 1, \dots, N$.
- 3 Взвешивание. $w_{t+1}^i = W_t(x_{t+1}^i, \tilde{x}_{1:t}^i)$, $i = 1, \dots, N$.

SMC это комбинация SIS и ресэмплинга.

- Непараметрические рекурсивные байесовские фильтры
- Распределения характеризуются своими выборками
- Не ограничены только нормальными распределениями
- Можно генерировать новые выборки
- Хорошо работают для пространств низкой размерности

Трекинг:

- При трекинге частицы описывают местоположение объекта
- Взвешиваются соответственно правдоподобию наблюдения
- Другое название — Monte-Carlo localization (MCL)
- MCL используется в мобильной робототехнике

Распределение $p(X_t)$ возможных состояний отслеживаемого объекта приближается набором взвешенных частиц (сэмплов) $S_t = \{s_t^i\}$, $j \in \{1 \dots N\}$, где каждая частица $s_t^i = (x_t^i, \pi_t^i)$ состоит из вектора состояния x_t^i и вектора весов π_t^i .

Набор частиц обновляется при переходе от одного кадра к другому по следующей рекурсивной процедуре:

- 1 Новый набор S_t получается из предыдущего S_{t-1} , где сэмпл s_{t-1}^i из старого набора выбирается с вероятностью, пропорциональной его весу π_{t-1}^i ;
- 2 Новые состояния x_t^j получаются сэмплированием из модели движения $p(X_t | X_{t-1} = x_{t-1}^i)$;
- 3 Измерение в новом кадре Z_t используется при обновлении весов π_t^i с помощью подсчета правдоподобия наблюдения, т.е. $\pi_t^j = p(Z_t | X_t = x_t^j, Z_0, Z_1, \dots, Z_{t-1})$.

Правдоподобие зависит от всех предыдущих кадров Z_0, Z_1, \dots, Z_{t-1} , если модель наблюдения адаптируется со временем. Для статической модели мы имеем $\pi_t^j = p(Z_t | X_t = x_t^j, Z_0)$.

Состояние частицы может моделироваться вектором

$$x = (x, y, w, h)^T,$$

где x, y — позиция отслеживаемого объекта на изображении с соответствующими ей размерами целевого окна.

В состояние также могут включаться первые моменты объекта:

$$x = (x, y, w, h, v_x, v_y)^T.$$

В простом случае модель движения оценивает новое состояние путем добавления нормального шума к предыдущему состоянию:

$$\begin{aligned}x_t &= x_{t-1} + N(0, \sigma_x^2), \\y_t &= y_{t-1} + N(0, \sigma_y^2), \\w_t &= w_{t-1} + N(0, \sigma_w^2), \\h_t &= h_{t-1} + N(0, \sigma_h^2).\end{aligned}$$

При наличии первых моментов в модели состояний шум накладывается на линейную первые моменты объекта вместо его координат:

$$\begin{aligned}v_{x,t} &= v_{x,t-1} + N(0, \sigma_x^2), \\v_{y,t} &= v_{y,t-1} + N(0, \sigma_y^2), \\x_t &= x_{t-1} + v_{x,t}, \\y_t &= y_{t-1} + v_{y,t}, \\w_t &= w_{t-1} + N(0, \sigma_w^2), \\h_t &= h_{t-1} + N(0, \sigma_h^2).\end{aligned}$$

Используется состояние с первыми моментами и даже отдельным классификатором со своей историей для каждой частицы:

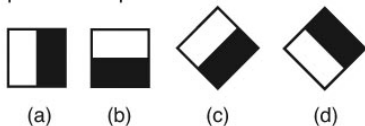
$$x = (x, y, w, h, v_x, v_y, C)^T.$$

Используется модель движения с первыми моментами, где константы определены как $\sigma_x = \sigma_y = 6.4$ и $\sigma_w = \sigma_h = 0.64$.
Правдоподобие

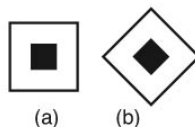
$$\begin{aligned}\pi_t^j &= p(Z_t | X_t = x_t^j, Z_0, Z_1, \dots, Z_{t-1}) \\ &= c \cdot \exp\left(\lambda \cdot C_t^j(x_t^j, y_t^j, w_t^j, h_t^j, Z_t)\right).\end{aligned}$$

Классификатор строится как композиция более слабых классификаторов на основе центральных признаков Хаара

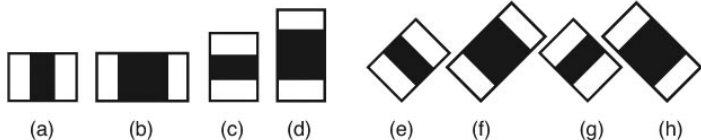
1. Граничные признаки



3. «Центральные» признаки

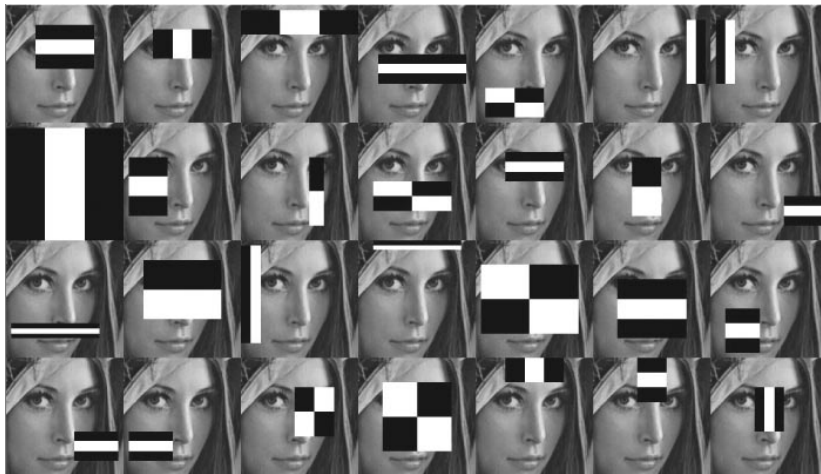


2. Линейные признаки



Видеотрекинг

«Видеотрекинг произвольных объектов»

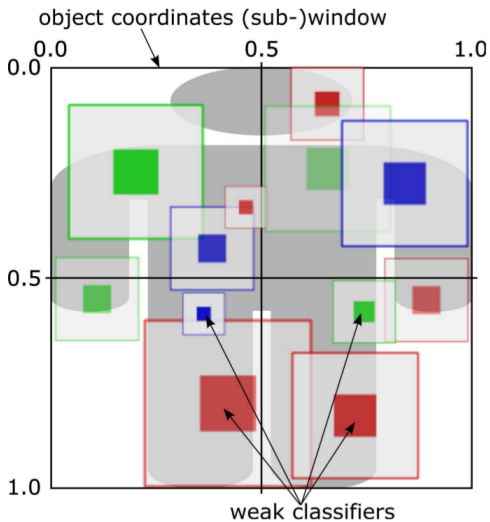


Композиция — Gentle Adaboost:

Дана размеченная выборка патчей $\{(x_1, y_1), \dots, (x_N, y_N)\}$.

- 1 Инициализируем веса $w_i = 1/N$.
- 2 Для $t = 1, \dots, T$
 - 1 Нормализуем веса
 - 2 Для каждого признака j обучаем слабый классификатор h_j .
 - 3 Выбираем классификатор, который минимизирует ошибку $\varepsilon_j = \sum_{i=1}^N w_i (y_i - f_t(x_i))^2$.
 - 4 Обновляем итоговую функцию: $F(x) = F(x) + f_t(x)$.
 - 5 Обновляем веса: $w_i = w_i e^{-y_i f_t(y_i)}$.
- 3 Получаем итоговый оценивающий классификатор $F(x) = \sum_{t=1}^T f_t(x)$.

- Используется относительная система координат для патча, промежуточные значения билинейно интерполируются
- Признаки Хаара различаются по положению и каналам изображения
- Обучается около 500 слабых классификаторов
- Из них отбираются 32 лучших
- Эвристика — заставляем GA выбрать поровну слабых классификаторов для всех четвертей изображения, в каждый момент из «пула» классификаторов убираются все, попадающие центрами в заполненную четверть

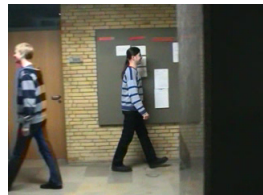


Адаптация:

- Обучение при переходе от кадра $t - 1$ к кадру t
- Храним историю признаков
- Храним как максимум 100 положительных и 20 отрицательных примеров
- При превышении лимитов на положительные патчи удаляем те, для которых получаем уверенный ответ
- В качестве исключения храним первый положительный пример с большим весом
- Генерируем заново небольшую долю отрицательных примеров из текущего фона

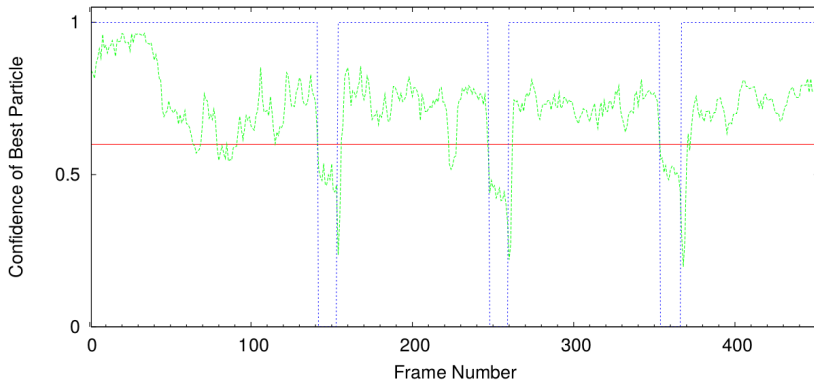
Оптимизируем:

- В бустинге набираем $n - k$ старых классификаторов и подбираем k новых из пула слабых классификаторов
- Концентрируемся на 10 лучших частицах (генерируют 50% предков), т.к. обновлять все пережившие ресэмплинг частицы в реалтайме сложно
- Не адаптируемся при уверенности ниже порога для борьбы со временными наложениями



Трем провалам и отказам от обучения для предыдущих изображений соответствуют пересечения колонн. Небольшой провал в середине — пересечение с другим человеком.

Adaption Prevention during Occlusion, Sequence G.



Демонстрация

- Dominik A. Klein, Dirk Schulz, Simone Frintrop, and Armin B. Cremers, Adaptive Real-Time Video-Tracking for Arbitrary Objects — оригинальная статья по видеотрекингу
http://www.iai.uni-bonn.de/~kleind/iros10_daklein.pdf
- BoBoT (Bonn Benchmark on Tracking) — бенчмарк на основе данных, рассмотренных в ней же
<http://www.iai.uni-bonn.de/~kleind/tracking/>
- Sequential Monte Carlo Methods & Particle Filters Resources by Arnaud Doucet — подборка ресурсов по теме
http://www.stats.ox.ac.uk/~doucet/smc_resources.html

Спасибо!