Sleeping experts and Expert Engineering

Yoav Freund

January 28, 2014

Outline

Outline

Outline

Specialists

- Also called sleeping experts

Specialists

- Also called sleeping experts
- The basic idea: specialists can associate a *confidence* with their predictions.

Specialists

- Also called sleeping experts
- The basic idea: specialists can associate a *confidence* with their predictions.
- Master's prediction depends more on the confident predictions.

Specialists

- Also called sleeping experts
- The basic idea: specialists can associate a *confidence* with their predictions.
- Master's prediction depends more on the confident predictions.
- The weight of confident experts is changed more than that of unconfident ones.

The specialists protocol

1. The adversary chooses a set $E_t \subseteq \{1, \ldots, N\}$ of specialists that are awake at iteration t.

The specialists protocol

1. The adversary chooses a set $E_t \subseteq \{1, \ldots, N\}$ of specialists that are awake at iteration t.
2. The adversary chooses a prediction $x_{t,i}$ for each awake specialist $i \in E_t$.

The specialists protocol

1. The adversary chooses a set $E_t \subseteq \{1, \ldots, N\}$ of specialists that are awake at iteration t.
2. The adversary chooses a prediction $x_{t,i}$ for each awake specialist $i \in E_t$.
3. The algorithm chooses its own prediction $\hat{y}_t$.

The specialists protocol

1. The adversary chooses a set $E_t \subseteq \{1, \ldots, N\}$ of specialists that are awake at iteration t.
2. The adversary chooses a prediction $x_{t,i}$ for each awake specialist $i \in E_t$.
3. The algorithm chooses its own prediction $\hat{y}_t$.
4. The adversary chooses an outcome y_t.

The specialists protocol

1. The adversary chooses a set $E_t \subseteq \{1, \ldots, N\}$ of specialists that are awake at iteration t.
2. The adversary chooses a prediction $x_{t,i}$ for each awake specialist $i \in E_t$.
3. The algorithm chooses its own prediction $\hat{y}_t$.
4. The adversary chooses an outcome y_t.
5. The algorithm suffers loss $\ell_A^t = L(\hat{y}_t, y_t)$ and each of the awake specialists suffers loss $\ell_i^t = L(x_{t,i}, y_t)$. Specialists that are asleep suffer no loss.

Log Loss

- Log loss is the simplest case

Log Loss

- Log loss is the simplest case
- $$L(\hat{y}, y) = \begin{cases} -\ln \hat{y} & \mathit{if}\, y = 1 \\ -\ln(1-\hat{y}) & \mathit{if}\, y = 0. \end{cases}$$

The standard Bayes algorithm (normalized weights)

Do for $t = 1, 2, \ldots, T$

1. Predict with the weighted average of the experts predictions:

$$\hat{y}_t = \sum_{i=1}^{N} p_{t,i} x_{t,i}$$

The standard Bayes algorithm (normalized weights)

Do for $t = 1, 2, \ldots, T$

1. Predict with the weighted average of the experts predictions:

$$\hat{y}_t = \sum_{i=1}^{N} p_{t,i} x_{t,i}$$

2. Observe outcome y_t

The standard Bayes algorithm (normalized weights)

Do for $t = 1, 2, \ldots, T$

1. Predict with the weighted average of the experts predictions:

$$\hat{y}_t = \sum_{i=1}^{N} p_{t,i} x_{t,i}$$

2. Observe outcome y_t
3. Calculate a new posterior distribution:

$$p_{t+1,i} = \begin{cases} \frac{p_{t,i} x_{t,i}}{\hat{y}_t} & if\, y_t{=}1 \\ \frac{p_{t,i}(1 - x_{t,i})}{1 - \hat{y}_t} & if\, y_t{=}0. \end{cases}$$

Bayes for Specialists

Do for $t = 1, 2, \ldots, T$

1. Predict with the weighted average of the predictions of the awake specialists:

$$\hat{y}_t = \frac{\sum_{i \in E_t} p_{t,i} x_{t,i}}{\sum_{i \in E_t} p_{t,i}}$$

Bayes for Specialists

Do for $t = 1, 2, \ldots, T$

1. Predict with the weighted average of the predictions of the awake specialists:

$$\hat{y}_t = \frac{\sum_{i \in E_t} p_{t,i} x_{t,i}}{\sum_{i \in E_t} p_{t,i}}$$

2. Observe outcome y_t

Bayes for Specialists

Do for $t = 1, 2, \ldots, T$

1. Predict with the weighted average of the predictions of the awake specialists:

$$\hat{y}_t = \frac{\sum_{i \in E_t} p_{t,i} x_{t,i}}{\sum_{i \in E_t} p_{t,i}}$$

2. Observe outcome y_t
3. Calculate a new posterior distribution:

   If $i \in E_t$ then

$$p_{t+1,i} = \begin{cases} \frac{p_{t,i} x_{t,i}}{\hat{y}_t} & if\, y_t{=}1 \\ \frac{p_{t,i}(1 - x_{t,i})}{1 - \hat{y}_t} & if\, y_t{=}0. \end{cases}$$

   Otherwise: $p_{t+1,i} = p_{t,i}$

Bound on Bayes for Specialists

Theorem

For any sequence of awake specialists, specialist predictions and outcomes and for any distribution $\mathbf{u}$ *over* $\{1, \ldots, N\}$, *the loss of* **SBayes** *satisfies*

$$\sum_{t=1}^{T} u(E_t) L(\hat{y}_t, y_t) \leq \sum_{t=1}^{T} \sum_{i \in E_t} u_i L(x_{t,i}, y_t) + \mathrm{RE}\left(\mathbf{u} \parallel \mathbf{p}_1\right) .$$

Where

$$u(E_t) \doteq \sum_{i \in E_t} u_i$$

Proof of Theorm

- for each step:

$$\begin{aligned} & \mathrm{RE}\left(\mathbf{u} \parallel \mathbf{p}_t\right) - \mathrm{RE}\left(\mathbf{u} \parallel \mathbf{p}_{t+1}\right) \\ = \quad & u(E_t)L(\hat{y}_t, y_t) - \sum_{i \in E_t} u_i L(x_{t,i}, y_t). \end{aligned} \tag{1}$$

Proof of Theorm

- for each step:

$$\begin{aligned} \mathrm{RE}(\mathbf{u} \parallel \mathbf{p}_t) - \mathrm{RE}(\mathbf{u} \parallel \mathbf{p}_{t+1}) \\ = \quad u(E_t)L(\hat{y}_t, y_t) - \sum_{i \in E_t} u_i L(x_{t,i}, y_t). \end{aligned} \tag{1}$$

- Summing over $t = 1, \ldots, T$ and using that relative entropy is always positive:

$$\begin{aligned} \mathrm{RE}(\mathbf{u} \parallel \mathbf{p}_1) \geq \mathrm{RE}(\mathbf{u} \parallel \mathbf{p}_1) - \mathrm{RE}(\mathbf{u} \parallel \mathbf{p}_{T+1}) \\ = \sum_{t=1}^{T} u(E_t)L(\hat{y}_t, y_t) - \sum_{t=1}^{T} \sum_{i \in E_t} u_i L(x_{t,i}, y_t). \end{aligned}$$

Using general loss functions

We focus on algorithms which, like **Bayes**, maintain a distribution vector $\mathbf{p}_t \in \Delta_N$. Such algorithms are defined by two functions:

1.
$$\mathrm{pred} : \Delta_N \times [0,1]^N \rightarrow [0,1]$$
which maps the current weight vector $\mathbf{p}_t$ and instance $\mathbf{x}_t$ to a prediction $\hat{y}_t$; and

Using general loss functions

We focus on algorithms which, like **Bayes**, maintain a distribution vector $\mathbf{p}_t \in \Delta_N$. Such algorithms are defined by two functions:

1.
$$\mathrm{pred} : \Delta_N \times [0,1]^N \to [0,1]$$
which maps the current weight vector $\mathbf{p}_t$ and instance $\mathbf{x}_t$ to a prediction $\hat{y}_t$; and

2.
$$\mathrm{update} : \Delta_N \times [0,1]^N \times [0,1] \to \Delta_N$$
which maps the current weight vector $\mathbf{p}_t$, instance $\mathbf{x}_t$ and outcome y_t to a new weight vector $\mathbf{p}_{t+1}$

Generic Insomniac Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe $\mathbf{x}_t$

Generic Insomniac Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe $\mathbf{x}_t$
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t, \mathbf{x}_t)$

Generic Insomniac Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe $\mathbf{x}_t$
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t, \mathbf{x}_t)$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$

Generic Insomniac Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe $\mathbf{x}_t$
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t, \mathbf{x}_t)$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$
4. Calculate the new weight vector
   $\mathbf{p}_{t+1} = \mathrm{update}(\mathbf{p}_t, \mathbf{x}_t, y_t)$

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and $\mathbf{x}_t^{E_t}$.

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and ${\mathbf{x}_t}^{E_t}$.
2. Predict $\hat{y}_t = \mathrm{pred}({\mathbf{p}_t}^{E_t}, {\mathbf{x}_t}^{E_t})$

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and $\mathbf{x}_t{}^{E_t}$.
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t{}^{E_t}, \mathbf{x}_t{}^{E_t})$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$.

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and $\mathbf{x}_t{}^{E_t}$.
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t{}^{E_t}, \mathbf{x}_t{}^{E_t})$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$.
4. Calculate the new weight vector $\mathbf{p}_{t+1}$ so that it satisfies the following:

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and $\mathbf{x}_t^{E_t}$.
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t^{E_t}, \mathbf{x}_t^{E_t})$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$.
4. Calculate the new weight vector $\mathbf{p}_{t+1}$ so that it satisfies the following:
   4.1 $p_{t+1,i} = p_{t,i}$ for $i \notin E_t$

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and $\mathbf{x}_t{}^{E_t}$.
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t{}^{E_t}, \mathbf{x}_t{}^{E_t})$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$.
4. Calculate the new weight vector $\mathbf{p}_{t+1}$ so that it satisfies the following:
   - 4.1 $p_{t+1,i} = p_{t,i}$ for $i \notin E_t$
   - 4.2 $\mathbf{p}_{t+1}{}^{E_t} = \frac{1}{z_t}\mathrm{update}(\mathbf{p}_t{}^{E_t}, \mathbf{x}_t{}^{E_t}, y_t)$

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and $\mathbf{x}_t^{E_t}$.
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t^{E_t}, \mathbf{x}_t^{E_t})$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$.
4. Calculate the new weight vector $\mathbf{p}_{t+1}$ so that it satisfies the following:
   - 4.1 $p_{t+1,i} = p_{t,i}$ for $i \notin E_t$
   - 4.2 $\mathbf{p}_{t+1}^{E_t} = \frac{1}{z_t}\mathrm{update}(\mathbf{p}_t^{E_t}, \mathbf{x}_t^{E_t}, y_t)$
   - 4.3 $\sum_{i=1}^{N} p_{t+1,i} = 1$

Generic Specialist Algorithm

Do for $t = 1, 2, \ldots, T$

1. Observe E_t and $\mathbf{x}_t{}^{E_t}$.
2. Predict $\hat{y}_t = \mathrm{pred}(\mathbf{p}_t{}^{E_t}, \mathbf{x}_t{}^{E_t})$
3. Observe outcome y_t and suffer loss $L(\hat{y}_t, y_t)$.
4. Calculate the new weight vector $\mathbf{p}_{t+1}$ so that it satisfies the following:
   - 4.1 $p_{t+1,i} = p_{t,i}$ for $i \notin E_t$
   - 4.2 $\mathbf{p}_{t+1}{}^{E_t} = \frac{1}{z_t}\mathrm{update}(\mathbf{p}_t{}^{E_t}, \mathbf{x}_t{}^{E_t}, y_t)$
   - 4.3 $\sum_{i=1}^{N} p_{t+1,i} = 1$
   - 4.4 or Equivalently: $\sum_{i \in E_t} p_{t+1,i} = \sum_{i \in E_t} p_{t,i}$

Comparison cumulative losses for specialists

- Comparison to average loss.

$$\min_{\mathbf{u}\in\Delta_N} \sum_{t=1}^{T} L^I_{\mathbf{u}}(\mathbf{x}_t, y_t) \quad \text{where} \quad L^I_{\mathbf{u}}(\mathbf{x}_t, y_t) \doteq \frac{\sum_{i\in E_t} u_i \; L(x_{t,i}, y_t)}{\sum_{i\in E_t} u_i} \; .$$

Comparison cumulative losses for specialists

- Comparison to average loss.

$$\min_{\mathbf{u}\in\Delta_N} \sum_{t=1}^{T} L^{I}_{\mathbf{u}}(\mathbf{x}_t, y_t) \quad \text{where} \quad L^{I}_{\mathbf{u}}(\mathbf{x}_t, y_t) \doteq \frac{\sum_{i\in E_t} u_i \; L(x_{t,i}, y_t)}{\sum_{i\in E_t} u_i} \; .$$

- Comparison to average prediction.

$$\min_{\mathbf{u}\in\Delta_N} \sum_{t=1}^{T} L^{II}_{\mathbf{u}}(\mathbf{x}_t, y_t) \quad \text{where} \quad L^{II}_{\mathbf{u}}(\mathbf{x}_t, y_t) \doteq L\left(\frac{\sum_{i\in E_t} u_i x_{t,i}}{\sum_{i\in E_t} u_i}, y_t\right)$$

Analysis using relative entropy

- Log-Loss / Bayes

$$\mathrm{RE}\left(\mathbf{u} \;\|\; \mathbf{p}_t\right) - \mathrm{RE}\left(\mathbf{u} \;\|\; \mathbf{p}_{t+1}\right) = L(\hat{y}_t, y_t) - \sum_{i=1}^{N} u_i L(x_{t,i}, y_t).$$

Analysis using relative entropy

- Log-Loss / Bayes

$$\mathrm{RE}\left(\mathbf{u} \parallel \mathbf{p}_t\right) - \mathrm{RE}\left(\mathbf{u} \parallel \mathbf{p}_{t+1}\right) = L(\hat{y}_t, y_t) - \sum_{i=1}^{N} u_i L(x_{t,i}, y_t).$$

- General Vovk-style algorithm:

$$c(\mathrm{RE}\left(\mathbf{u} \parallel \mathbf{p}_t\right) - \mathrm{RE}\left(\mathbf{u} \parallel \mathbf{p}_{t+1}\right)) \geq L(\hat{y}_t, y_t) - aL_{\mathbf{u}}(\mathbf{x}_t, y_t).$$

Where L is (a, c)-achievable (Using Vovk with $\eta = a/c$)

Bound for general loss sleeping experts

For any achievable (a, c)

$$\sum_{t=1}^{T} u(E_t) L(\hat{y}_t, y_t) \leq a \sum_{t=1}^{T} u(E_t) L_{\mathbf{u}^{E_t}}({\mathbf{x}_t}^{E_t}, y_t) + c\mathrm{RE}\left(\mathbf{u} \;\|\; \mathbf{p}_1\right) \quad .$$

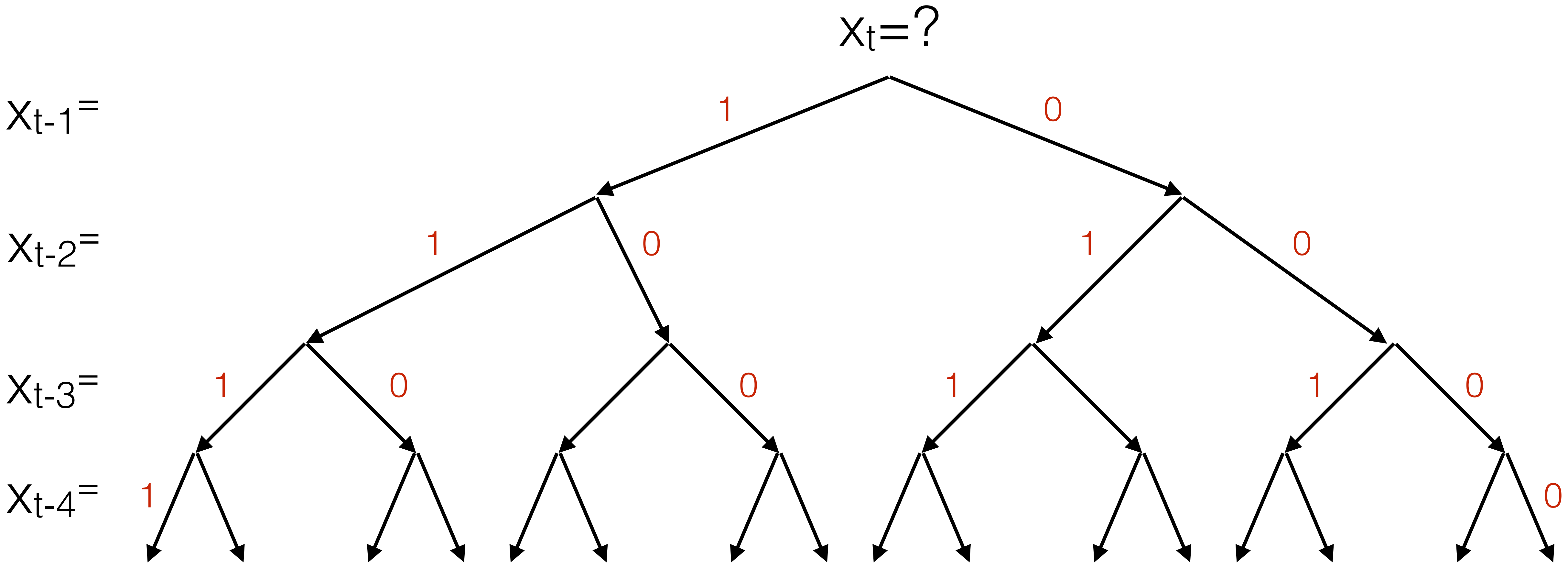
Where

$$u(E_t) \doteq \sum_{i \in E_t} u_i$$

SBayes satisfies

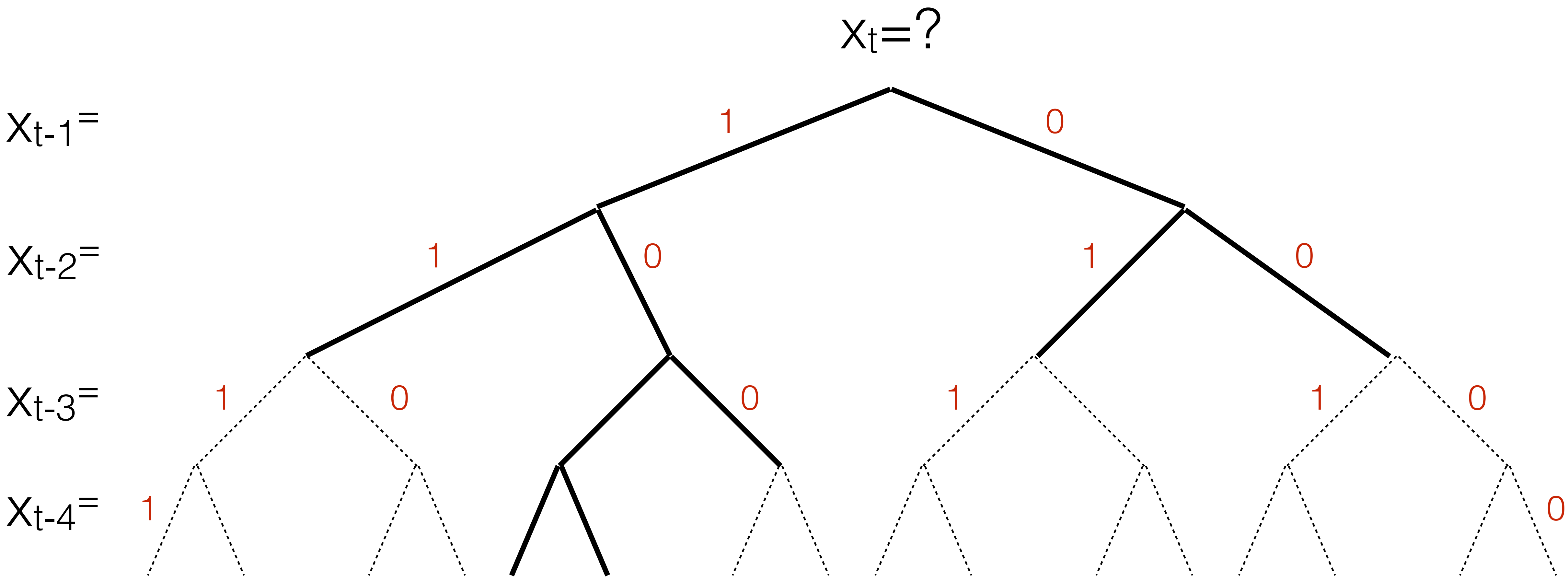
$$\sum_{t=1}^{T} u(E_t) L(\hat{y}_t, y_t) \leq \sum_{t=1}^{T} \sum_{i \in E_t} u_i L(x_{t,i}, y_t) + \mathrm{RE}\left(\mathbf{u} \;\|\; \mathbf{p}_1\right) \quad .$$

Markov Model of order 4



In each leaf node we estimate $P(x_t \mid x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4})$

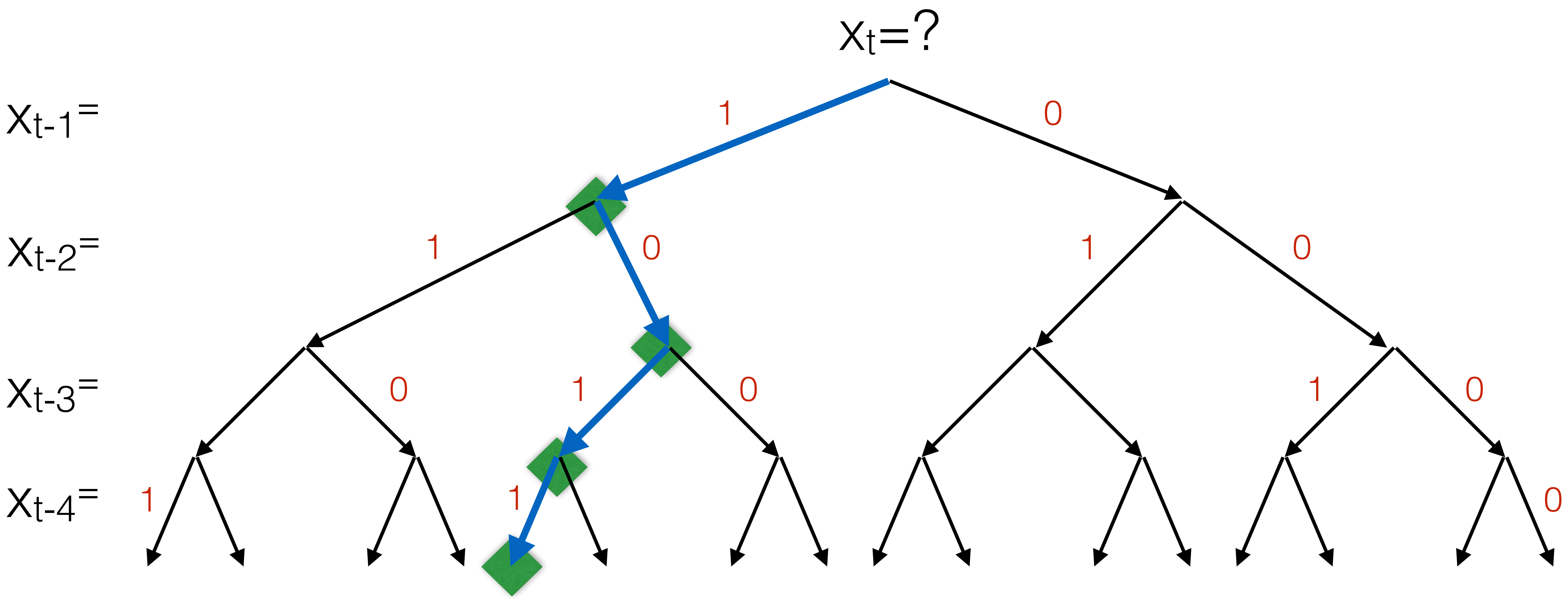
Variable Length Markov Model



- **In each leaf node we estimate** $\mathbb{P}(x_t \mid x_{t-1}, x_{t-2}, \ldots)$
- **A VMM for each prefix-free subtree**
- **An expert for each subtree**

  = An exponential number of experts

VMM using specialists



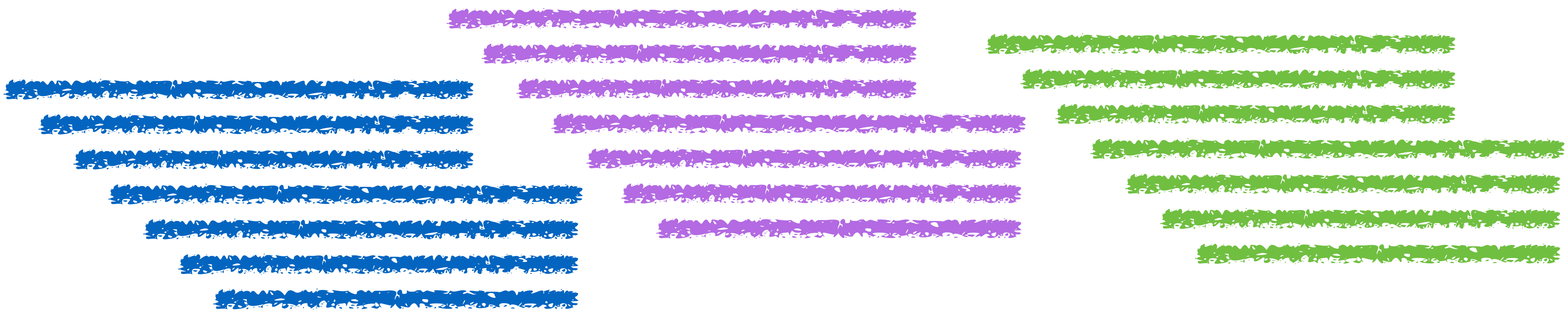
- **Each node corresponds to a specialist**
- **Each specialist estimates** $P(x_t \mid x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4})$
- **Number of specialists = number of nodes**
- **At each time t, 4 specialists are awake.**
- **Example: 1,1,0,1,?**

Switching experts

Base Experts

Combined Expert:

Low-Level specialists: ($t_1<t_2$) for each base expert



Actual algorithm maintains one weight per base expert (color),
Same as summing over all low-level specialists

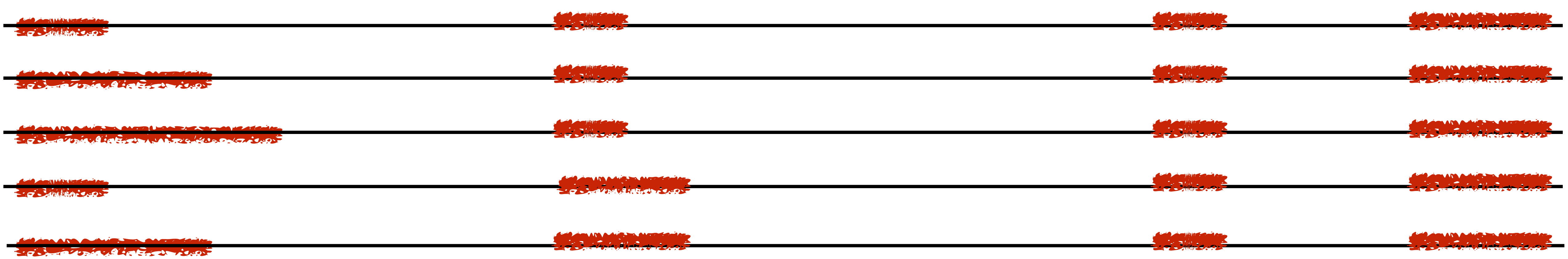
Switching within a small set of experts

time

Base Experts

Combined Expert:

Low-Level specialists: $(t_1 < t_2 < t_3 < \ldots < t_n)$ for each base expert



Actual algorithm maintains one weight per base expert (color),
Same as summing over all low-level specialists

The Text Classification Problem

Context-Sensitive Learning Methods for Text Categorization / Cohen and Singer 1999

To classify a new document d using this pool, one first finds all sparse n-grams appearing in the document, and then computes the weights of the corresponding miniexperts. For instance, in classifying the documents "prayers said for soldiers killed in ira bombing" and "taxi driver killed by ira" the relevant set of phrases would include "killed ? ira" and "bombing". The documents above are classified correctly; among the miniexperts associated with these phrases, the total weight of the miniexperts predicting $d \in$ ireland is larger than the total weight of the miniexperts predicting $d \notin$ ireland.

Using Specialists for text classification

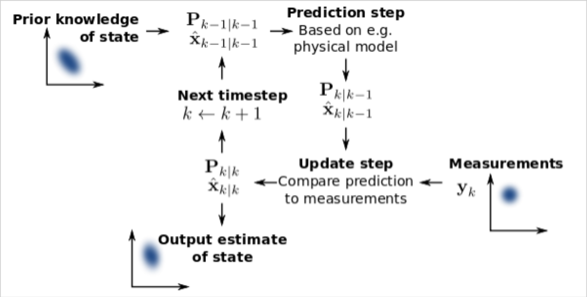
150 • W. W. Cohen and Y. Singer

Table I. Experts with Large Weights for the Category ireland

	Log-Weight		Number of Occurrences	
Phrase	$\notin$ ireland	$\in$ ireland	$\notin$ ireland	$\in$ ireland
belfast	-7.19	12.05	8	31
haughey	-6.35	11.10	2	10
ira says	-1.07	10.44	2	7
northern ireland	-7.20	10.17	18	38
catholic man	-0.87	6.03	0	3
ulster	-3.98	5.20	4	8
killed ? ira	-0.09	4.68	1	4
protestant extremists claim	-0.12	4.59	0	2
moderate catholic	-0.02	4.58	0	2
ira supporters	-3.20	3.68	0	3
sinn fein	-3.52	3.38	2	5
west belfast	-5.90	3.05	3	16

Dynamics using Kalman Filters

Too many resources to list.



The Kalman filter keeps track of the estimated state of the system and the variance or uncertainty of the estimate. The estimate is updated using a state transition model and measurements. $\hat{x}_{k|k-1}$ denotes the estimate of the system's state at time step k before the k-th measurement y_k has been taken into account; $P_{k|k-1}$ is the corresponding uncertainty.

Dynamics using Particle Filters

The unscented particle filter / R. Van Der Merwe, A. Doucet, N. De Freitas, E. Wan

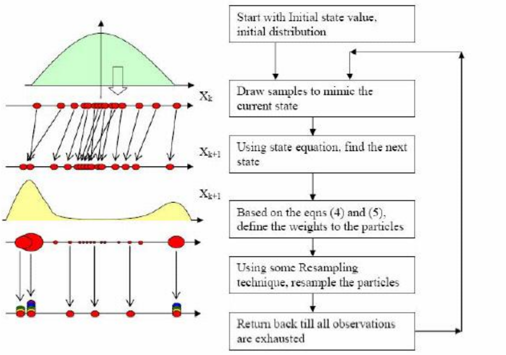


Figure 2. The flow of particles to initial distribution to the predict stage into update stage and resample stage, back to the predict stage till all the samples are exhausted.

Specialists for dynamics

- Tracking for interaction.

Specialists for dynamics

- Tracking for interaction.
- Handwriting recognition (Sunsern)

Experts for appearance modeling

- Templates - sample image patch and compare to future patches.

Experts for appearance modeling

- Templates - sample image patch and compare to future patches.
- Identify location of object using a boosted combination of low-level features. (Online Boosting)

Experts for appearance modeling

- Templates - sample image patch and compare to future patches.
- Identify location of object using a boosted combination of low-level features. (Online Boosting)
- Specialists: tracking the best appearance model.

Experts for appearance modeling

- Templates - sample image patch and compare to future patches.
- Identify location of object using a boosted combination of low-level features. (Online Boosting)
- Specialists: tracking the best appearance model.
- Within a small set: assuming that old appearances will recur.

Confidence

- Can we quantify the confidence we have in our prediction?

Confidence

- Can we quantify the confidence we have in our prediction?
- If there is a set of awake specialists that have a large weight and make similar predictions.

Confidence

- Can we quantify the confidence we have in our prediction?
- If there is a set of awake specialists that have a large weight and make similar predictions.
- In Kalman filters: covariance of the posterior distribution.

Co-Training

- When tracking, we have no ground truth - how can we train our models?

Co-Training

- When tracking, we have no ground truth - how can we train our models?
- Co-training: Train in proportion to confidence

Co-Training

- When tracking, we have no ground truth - how can we train our models?
- Co-training: Train in proportion to confidence
- When Dynamics is confident: use it to train appearance.

Co-Training

- When tracking, we have no ground truth - how can we train our models?
- Co-training: Train in proportion to confidence
- When Dynamics is confident: use it to train appearance.
- When appearance is confident: use it to train dynamics.

Co-Training

- When tracking, we have no ground truth - how can we train our models?
- Co-training: Train in proportion to confidence
- When Dynamics is confident: use it to train appearance.
- When appearance is confident: use it to train dynamics.
- Specialists can correspond to using different features, different image resolutions etc.