

# Estimation, Tracking and control using **Hedge**( $\eta$ )

Yoav Freund

January 18, 2010

# Outline

## Estimation

# Outline

Estimation

Tracking

# Outline

Estimation

Tracking

Control

# The static discrete estimation problem

- ▶ A system has an internal unobservable **state**:  
 $s \in \{0, 1, \dots, K\}$

## The static discrete estimation problem

- ▶ A system has an internal unobservable **state**:  
 $s \in \{0, 1, \dots, K\}$
- ▶ **observations** are corrupted versions of the state:  
 $o_1, o_2, \dots, o_t \in \{0, 1, \dots, K\}$

## The static discrete estimation problem

- ▶ A system has an internal unobservable **state**:  
 $s \in \{0, 1, \dots, K\}$
- ▶ **observations** are corrupted versions of the state:  
 $o_1, o_2, \dots, o_t \in \{0, 1, \dots, K\}$
- ▶ **Goal**: given  $o_1, o_2, \dots, o_t$  compute prediction  $p_{t+1}$  that is close to  $o_{t+1}$

# Generative modeling

- ▶ Prob. of observation  $o$  conditioned on state  $s$ :  
 $P(O = o | S = s)$



## Generative modeling

- ▶ Prob. of observation  $o$  conditioned on state  $s$ :  
 $P(O = o | S = s)$
- ▶ Given  $o_1, o_2, \dots, o_T$  define likelihood of state  $s$  as  
 $\prod_{t=1}^T P(O = o_t | S = s)$ .

## Generative modeling

- ▶ Prob. of observation  $o$  conditioned on state  $s$ :  
 $P(O = o | S = s)$
- ▶ Given  $o_1, o_2, \dots, o_T$  define likelihood of state  $s$  as  
 $\prod_{t=1}^T P(O = o_t | S = s)$ .
- ▶ max-likelihood estimator  $\hat{s}$ :  $s$  that maximized likelihood or log-likelihood:  
 $\sum_{t=1}^T \log P(O = o_t | S = s)$

## Generative modeling

- ▶ Prob. of observation  $o$  conditioned on state  $s$ :  
 $P(O = o | S = s)$
- ▶ Given  $o_1, o_2, \dots, o_T$  define likelihood of state  $s$  as  
 $\prod_{t=1}^T P(O = o_t | S = s)$ .
- ▶ max-likelihood estimator  $\hat{s}$ :  $s$  that maximized likelihood or log-likelihood:  
 $\sum_{t=1}^T \log P(O = o_t | S = s)$
- ▶ Predict using the estimate:  $p_{t+1} = \hat{s}$

## Generative modeling

- ▶ Prob. of observation  $o$  conditioned on state  $s$ :  
 $P(O = o | S = s)$
- ▶ Given  $o_1, o_2, \dots, o_T$  define likelihood of state  $s$  as  
 $\prod_{t=1}^T P(O = o_t | S = s)$ .
- ▶ max-likelihood estimator  $\hat{s}$ :  $s$  that maximized likelihood or log-likelihood:  
 $\sum_{t=1}^T \log P(O = o_t | S = s)$
- ▶ Predict using the estimate:  $p_{t+1} = \hat{s}$
- ▶ Real-world problem - often  $P(O = o | S = s)$  - the correct conditional distribution - is not known!

## Estimation by minimizing loss

- ▶ Define a loss function to measure the discrepancy between the prediction  $p_{t+1}$  and observation  $o_{t+1}$ .

## Estimation by minimizing loss

- ▶ Define a loss function to measure the discrepancy between the prediction  $p_{t+1}$  and observation  $o_{t+1}$ .
- ▶ Examples:  $\ell_1(p, o) = |p - o|$ ,  $\ell_2(p, o) = (p - o)^2$

## Estimation by minimizing loss

- ▶ Define a loss function to measure the discrepancy between the prediction  $p_{t+1}$  and observation  $o_{t+1}$ .
- ▶ Examples:  $\ell_1(p, o) = |p - o|$ ,  $\ell_2(p, o) = (p - o)^2$
- ▶ Given  $o_1, o_2, \dots, o_T$  use the minimal loss estimate.

## Estimation by minimizing loss

- ▶ Define a loss function to measure the discrepancy between the prediction  $p_{t+1}$  and observation  $o_{t+1}$ .
- ▶ Examples:  $\ell_1(p, o) = |p - o|$ ,  $\ell_2(p, o) = (p - o)^2$
- ▶ Given  $o_1, o_2, \dots, o_T$  use the minimal loss estimate.
- ▶ No such thing as a “correct” loss.



## Estimation by minimizing loss

- ▶ Define a loss function to measure the discrepancy between the prediction  $p_{t+1}$  and observation  $o_{t+1}$ .
- ▶ Examples:  $\ell_1(p, o) = |p - o|$ ,  $\ell_2(p, o) = (p - o)^2$
- ▶ Given  $o_1, o_2, \dots, o_T$  use the minimal loss estimate.
- ▶ No such thing as a “correct” loss.
- ▶ No direct relation to estimation of hidden state.

## Estimation by minimizing loss

- ▶ Define a loss function to measure the discrepancy between the prediction  $p_{t+1}$  and observation  $o_{t+1}$ .
- ▶ Examples:  $\ell_1(p, o) = |p - o|$ ,  $\ell_2(p, o) = (p - o)^2$
- ▶ Given  $o_1, o_2, \dots, o_T$  use the minimal loss estimate.
- ▶ No such thing as a “correct” loss.
- ▶ No direct relation to estimation of hidden state.
- ▶ Often useful to define an outlier insensitive loss:  
 $\ell(p, o) = \min(c, |p - o|)$

## Estimation by minimizing loss

- ▶ Define a loss function to measure the discrepancy between the prediction  $p_{t+1}$  and observation  $o_{t+1}$ .
- ▶ Examples:  $\ell_1(p, o) = |p - o|$ ,  $\ell_2(p, o) = (p - o)^2$
- ▶ Given  $o_1, o_2, \dots, o_T$  use the minimal loss estimate.
- ▶ No such thing as a “correct” loss.
- ▶ No direct relation to estimation of hidden state.
- ▶ Often useful to define an outlier insensitive loss:  
 $\ell(p, o) = \min(c, |p - o|)$
- ▶ Leads to non-convex optimization problems.

## Estimation using **Hedge**( $\eta$ )

- ▶ Consider each prediction  $p = 0, 1, \dots, K$  as an expert.

## Estimation using **Hedge**( $\eta$ )

- ▶ Consider each prediction  $p = 0, 1, \dots, K$  as an expert.
- ▶ **Hedge**( $\eta$ ) will define a distribution over  $0, 1, \dots, K$ .

## Estimation using **Hedge**( $\eta$ )

- ▶ Consider each prediction  $p = 0, 1, \dots, K$  as an expert.
- ▶ **Hedge**( $\eta$ ) will define a distribution over  $0, 1, \dots, K$ .
- ▶ The distribution defines our **confidence interval**.

## Estimation using **Hedge**( $\eta$ )

- ▶ Consider each prediction  $p = 0, 1, \dots, K$  as an expert.
- ▶ **Hedge**( $\eta$ ) will define a distribution over  $0, 1, \dots, K$ .
- ▶ The distribution defines our **confidence interval**.
- ▶ Guaranteed to perform almost as well as the best estimate in hindsight.

## Estimation using **Hedge**( $\eta$ )

- ▶ Consider each prediction  $p = 0, 1, \dots, K$  as an expert.
- ▶ **Hedge**( $\eta$ ) will define a distribution over  $0, 1, \dots, K$ .
- ▶ The distribution defines our **confidence interval**.
- ▶ Guaranteed to perform almost as well as the best estimate in hindsight.
- ▶ Similar to Bayesian posterior distribution, but does not assume known distribution of noise.



## The discrete **tracking** problem

- ▶ The internal state of the system changes **slowly** over time.  
 $s_1, s_2, \dots, s_t \in \{0, 1, \dots, K\}$

## The discrete **tracking** problem

- ▶ The internal state of the system changes **slowly** over time.  
 $s_1, s_2, \dots, s_t \in \{0, 1, \dots, K\}$
- ▶ **observations** are corrupted versions of the state:  
 $o_1, o_2, \dots, o_t \in \{0, 1, \dots, K\}$

## The discrete **tracking** problem

- ▶ The internal state of the system changes **slowly** over time.  
 $s_1, s_2, \dots, s_t \in \{0, 1, \dots, K\}$
- ▶ **observations** are corrupted versions of the state:  
 $o_1, o_2, \dots, o_t \in \{0, 1, \dots, K\}$
- ▶ **Goal**: for each  $t$ , given  $o_1, o_2, \dots, o_t$  compute  $p_{t+1}$  - a prediction of  $o_{t+1}$ .

## Tracking using Generative models

- ▶ To define a time varying generative model we need to know the **correct** distribution of the state transitions

## Tracking using Generative models

- ▶ To define a time varying generative model we need to know the **correct** distribution of the state transitions
- ▶ Hidden Markov Models: define  $Pr(S_{t+1}|S_t)$

## Tracking using Generative models

- ▶ To define a time varying generative model we need to know the **correct** distribution of the state transitions
- ▶ Hidden Markov Models: define  $Pr(S_{t+1}|S_t)$
- ▶ likelihood of observed sequence equal to sum over all possible hidden sequences, but can be computed efficiently using dynamic programming.

## Tracking using Generative models

- ▶ To define a time varying generative model we need to know the **correct** distribution of the state transitions
- ▶ Hidden Markov Models: define  $Pr(S_{t+1}|S_t)$
- ▶ likelihood of observed sequence equal to sum over all possible hidden sequences, but can be computed efficiently using dynamic programming.
- ▶ Popular method in speech recognition.

# Tracking by loss minimization

- ▶ Only assumption: state changes **slowly**.



## Tracking by loss minimization

- ▶ Only assumption: state changes **slowly**.
- ▶ Instead of cumulative loss, use exponentially discounted loss  $L_{t+1} = (1 - \alpha)L_t + \ell_t$

## Tracking by loss minimization

- ▶ Only assumption: state changes **slowly**.
- ▶ Instead of cumulative loss, use exponentially discounted loss  $L_{t+1} = (1 - \alpha)L_t + \ell_t$
- ▶  $\alpha = 0$  corresponds to standard cumulative loss.

## Tracking by loss minimization

- ▶ Only assumption: state changes **slowly**.
- ▶ Instead of cumulative loss, use exponentially discounted loss  $L_{t+1} = (1 - \alpha)L_t + \ell_t$
- ▶  $\alpha = 0$  corresponds to standard cumulative loss.
- ▶  $\alpha > 0$  corresponds approximately to averaging over the previous  $1/\alpha$  iterations.

## Using **Hedge**( $\eta$ ) for tracking

- Use exponentially discounted loss.

## Using **Hedge( $\eta$ )** for tracking

- ▶ Use exponentially discounted loss.
- ▶ **Hedge( $\eta$ )** guaranteed to perform almost as well as best expert **with respect to exp. discounted loss**.

## Using **Hedge( $\eta$ )** for tracking

- ▶ Use exponentially discounted loss.
- ▶ **Hedge( $\eta$ )** guaranteed to perform almost as well as best expert **with respect to exp. discounted loss**.
- ▶ Tracks state well when changes occur every  **$1/\alpha$**  examples.

## Using **Hedge( $\eta$ )** for tracking

- ▶ Use exponentially discounted loss.
- ▶ **Hedge( $\eta$ )** guaranteed to perform almost as well as best expert **with respect to exp. discounted loss**.
- ▶ Tracks state well when changes occur every  $1/\alpha$  examples.
- ▶ Choosing the learning rate  $\eta$  is a significant problem.

# Tracking dynamical systems

- ▶ Each expert can corresponds to a **trajectory** in state space.



## Tracking dynamical systems

- ▶ Each expert can corresponds to a **trajectory** in state space.
- ▶ **full state**: a description of the state that is sufficient to predict the future.

# Tracking dynamical systems

- ▶ Each expert can corresponds to a **trajectory** in state space.
- ▶ **full state**: a description of the state that is sufficient to predict the future.
- ▶ State of physical rigid object: location, speed, rotation, rotations speeds. In physics: “**phase space**”

## Tracking dynamical systems

- ▶ Each expert can corresponds to a **trajectory** in state space.
- ▶ **full state**: a description of the state that is sufficient to predict the future.
- ▶ State of physical rigid object: location, speed, rotation, rotations speeds. In physics: “**phase space**”
- ▶ Without drift (dynamic noise) the trajectory is deterministically determined from the state.

## Dealing with drift

- ▶  $s_{t+1}$  is not fully determined by  $s_t$ .

## Dealing with drift

- ▶  $s_{t+1}$  is not fully determined by  $s_t$ .
- ▶ We borrow from generative modeling.

## Dealing with drift

- ▶  $s_{t+1}$  is not fully determined by  $s_t$ .
- ▶ We borrow from generative modeling.
- ▶ Split each expert  $s_t$  into a distribution over experts:  
 $Pr(s_{t+1}|s_t)$ .

## Dealing with drift

- ▶  $s_{t+1}$  is not fully determined by  $s_t$ .
- ▶ We borrow from generative modeling.
- ▶ Split each expert  $s_t$  into a distribution over experts:  
 $Pr(s_{t+1}|s_t)$ .
- ▶ Use exponentially discounted loss to emphasize recent history.

## Dealing with drift

- ▶  $s_{t+1}$  is not fully determined by  $s_t$ .
- ▶ We borrow from generative modeling.
- ▶ Split each expert  $s_t$  into a distribution over experts:  
 $Pr(s_{t+1}|s_t)$ .
- ▶ Use exponentially discounted loss to emphasize recent history.
- ▶ Number of experts grows exponentially.



## Dealing with drift

- ▶  $s_{t+1}$  is not fully determined by  $s_t$ .
- ▶ We borrow from generative modeling.
- ▶ Split each expert  $s_t$  into a distribution over experts:  
 $Pr(S_{t+1}|S_t)$ .
- ▶ Use exponentially discounted loss to emphasize recent history.
- ▶ Number of experts grows exponentially.
- ▶ We can use dynamic programming - stil very expensive.

## Dealing with drift

- ▶  $s_{t+1}$  is not fully determined by  $s_t$ .
- ▶ We borrow from generative modeling.
- ▶ Split each expert  $s_t$  into a distribution over experts:  
 $Pr(S_{t+1}|S_t)$ .
- ▶ Use exponentially discounted loss to emphasize recent history.
- ▶ Number of experts grows exponentially.
- ▶ We can use dynamic programming - stil very expensive.
- ▶ We use monte-carlo sampling.

## The discrete **control** problem

- ▶ The internal state of the system (**plant**) changes due to:

# The discrete **control** problem

- ▶ The internal state of the system (**plant**) changes due to:
  - ▶ Internal dynamics.

## The discrete **control** problem

- ▶ The internal state of the system (**plant**) changes due to:
  - ▶ Internal dynamics.
  - ▶ External control signal.

# The discrete **control** problem

- ▶ The internal state of the system (**plant**) changes due to:
  - ▶ Internal dynamics.
  - ▶ External control signal.
  - ▶ Drift.

## The discrete **control** problem

- ▶ The internal state of the system (**plant**) changes due to:
  - ▶ Internal dynamics.
  - ▶ External control signal.
  - ▶ Drift.
- ▶ Goal of the controller given the observation history, generate a control signal to bring the plant close to desired state.

## The discrete **control** problem

- ▶ The internal state of the system (**plant**) changes due to:
  - ▶ Internal dynamics.
  - ▶ External control signal.
  - ▶ Drift.
- ▶ Goal of the controller given the observation history, generate a control signal to bring the plant close to desired state.
  - ▶ In short amount of time.



# The discrete **control** problem

- ▶ The internal state of the system (**plant**) changes due to:
  - ▶ Internal dynamics.
  - ▶ External control signal.
  - ▶ Drift.
- ▶ Goal of the controller given the observation history, generate a control signal to bring the plant close to desired state.
  - ▶ In short amount of time.
  - ▶ Using small amount of power.

## control based on generative models

- Requires generative model of plant

## control based on generative models

- ▶ Requires generative model of plant
  - ▶ Dynamics of plant.

## control based on generative models

- ▶ Requires generative model of plant
  - ▶ Dynamics of plant.
  - ▶ Distribution model of drift.

## control based on generative models

- ▶ Requires generative model of plant
  - ▶ Dynamics of plant.
  - ▶ Distribution model of drift.
  - ▶ Distribution of observation noise.

## control based on generative models

- ▶ Requires generative model of plant
  - ▶ Dynamics of plant.
  - ▶ Distribution model of drift.
  - ▶ Distribution of observation noise.
- ▶ Analysis: combine controller and plant into a single dynamic system and analyze its properties under the generative model.

## control based on experts

- ▶ Expert is a mapping from past observations to a control signal.

## control based on experts

- ▶ Expert is a mapping from past observations to a control signal.
- ▶ Loss is the difference between observation of desired observation (corresponding to desired state).



## control based on experts

- ▶ Expert is a mapping from past observations to a control signal.
- ▶ Loss is the difference between observation of desired observation (corresponding to desired state).
- ▶ Similar to driving a car without understanding mechanics.

## control based on experts

- ▶ Expert is a mapping from past observations to a control signal.
- ▶ Loss is the difference between observation of desired observation (corresponding to desired state).
- ▶ Similar to driving a car without understanding mechanics.
- ▶ Experts **can** be based on generative models. **Can** estimate the state of the plant, etc.

## control based on experts

- ▶ Expert is a mapping from past observations to a control signal.
- ▶ Loss is the difference between observation of desired observation (corresponding to desired state).
- ▶ Similar to driving a car without understanding mechanics.
- ▶ Experts **can** be based on generative models. **Can** estimate the state of the plant, etc.
- ▶ Experts can be also be dumb!

## control based on experts

- ▶ Expert is a mapping from past observations to a control signal.
- ▶ Loss is the difference between observation of desired observation (corresponding to desired state).
- ▶ Similar to driving a car without understanding mechanics.
- ▶ Experts **can** be based on generative models. **Can** estimate the state of the plant, etc.
- ▶ Experts can be also be dumb!
- ▶ When system is complex, dumb expert is likely to perform better than complex expert.