Combining infinite sets of experts

Yoav Freund

January 24, 2006

Outline

Outline

Outline

Outline

Outline

Outline

The online Bayes Algorithm

- Total loss of expert i

$$L_i^t = -\sum_{s=1}^{t} \log p_i^s(c^s); \quad L_i^0 = 0$$

The online Bayes Algorithm

- Total loss of expert i

$$L_i^t = -\sum_{s=1}^{t} \log p_i^s(c^s); \quad L_i^0 = 0$$

- Weight of expert i

$$w_i^t = w_i^1 e^{-L_i^{t-1}} = w_i^1 \prod_{s=1}^{t-1} p_i^s(c^s)$$

The online Bayes Algorithm

- Total loss of expert i

$$L_i^t = -\sum_{s=1}^{t} \log p_i^s(c^s); \quad L_i^0 = 0$$

- Weight of expert i

$$w_i^t = w_i^1 e^{-L_i^{t-1}} = w_i^1 \prod_{s=1}^{t-1} p_i^s(c^s)$$

- Freedom to choose initial weights.
  $w_t^1 \geq 0, \sum_{i=1}^{n} w_i^1 = 1$

The online Bayes Algorithm

- Total loss of expert i

$$L_i^t = -\sum_{s=1}^{t} \log p_i^s(c^s); \quad L_i^0 = 0$$

- Weight of expert i

$$w_i^t = w_i^1 e^{-L_i^{t-1}} = w_i^1 \prod_{s=1}^{t-1} p_i^s(c^s)$$

- Freedom to choose initial weights.
  $w_t^1 \geq 0$, $\sum_{i=1}^n w_i^1 = 1$
- Prediction of algorithm A

$$\mathbf{p}_A^t = \frac{\sum_{i=1}^N w_i^t \mathbf{p}_i^t}{\sum_{i=1}^N w_i^t}$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^{N} w_i^t$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t}$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t}$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t} = p_A^t(c^t)$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t} = p_A^t(c^t)$$

$$-\log \frac{W^{t+1}}{W^t} = -\log p_A^t(c^t)$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t} = p_A^t(c^t)$$

$$-\log \frac{W^{t+1}}{W^t} = -\log p_A^t(c^t)$$

$$-\log \frac{W^{T+1}}{W^1} = -\sum_{t=1}^T \log p_A^t(c^t)$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t} = p_A^t(c^t)$$

$$-\log \frac{W^{t+1}}{W^t} = -\log p_A^t(c^t)$$

$$-\log \frac{W^{T+1}}{W^1} = -\sum_{t=1}^T \log p_A^t(c^t) = L_A^T$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t} = p_A^t(c^t)$$

$$-\log \frac{W^{t+1}}{W^t} = -\log p_A^t(c^t)$$

$$-\log W^{T+1} = -\log \frac{W^{T+1}}{W^1} = -\sum_{t=1}^{T} \log p_A^t(c^t) = L_A^T$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t} = p_A^t(c^t)$$

$$-\log \frac{W^{t+1}}{W^t} = -\log p_A^t(c^t)$$

$$-\log W^{T+1} = -\log \frac{W^{T+1}}{W^1} = -\sum_{t=1}^T \log p_A^t(c^t) = L_A^T$$

EQUALITY not bound!

Simple Bound

- Use non-uniform initial weights $\sum_i w_i^1 = 1$

Simple Bound

- Use non-uniform initial weights $\sum_i w_i^1 = 1$
- Total Weight is at least the weight of the best expert.

$$L_A^T = -\log W^{T+1}$$

Simple Bound

- Use non-uniform initial weights $\sum_i w_i^1 = 1$
- Total Weight is at least the weight of the best expert.

$$L_A^T = -\log W^{T+1}$$

Simple Bound

- Use non-uniform initial weights $\sum_i w_i^1 = 1$
- Total Weight is at least the weight of the best expert.

$$L_A^T \quad = \quad -\log W^{T+1} = -\log \sum_{i=1}^{N} w_i^{T+1}$$

Simple Bound

- Use non-uniform initial weights $\sum_i w_i^1 = 1$
- Total Weight is at least the weight of the best expert.

$$
\begin{aligned}
L_A^T &= -\log W^{T+1} = -\log \sum_{i=1}^{N} w_i^{T+1} \\
&= -\log \sum_{i=1}^{N} w_i^1 e^{-L_i^T}
\end{aligned}
$$

Simple Bound

- Use non-uniform initial weights $\sum_i w_i^1 = 1$
- Total Weight is at least the weight of the best expert.

$$
\begin{aligned}
L_A^T &= -\log W^{T+1} = -\log \sum_{i=1}^{N} w_i^{T+1} \\
&= -\log \sum_{i=1}^{N} w_i^1 e^{-L_i^T} \leq -\log \max_i \left(w_i^1 e^{-L_i^T} \right)
\end{aligned}
$$

Simple Bound

- Use non-uniform initial weights $\sum_i w_i^1 = 1$
- Total Weight is at least the weight of the best expert.

$$
\begin{aligned}
L_A^T &= -\log W^{T+1} = -\log \sum_{i=1}^{N} w_i^{T+1} \\
&= -\log \sum_{i=1}^{N} w_i^1 e^{-L_i^T} \le -\log \max_i \left(w_i^1 e^{-L_i^T} \right) \\
&= \min_i \left(L_i^T - \log w_i^1 \right)
\end{aligned}
$$

Standardizing online prediction algorithms

- Fix a universal Turing machine U.

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0,1\}^t$

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0,1\}^t$
  - runs finite time and outputs

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0,1\}^t$
  - runs finite time and outputs
  - A prediction for the next bit $p(\vec{X}) \in [0,1]$.

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0, 1\}^t$
  - runs finite time and outputs
  - A prediction for the next bit $p(\vec{X}) \in [0, 1]$.
  - To ensure p has a finite description. Restrict to rational numbers n/m

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0,1\}^t$
  - runs finite time and outputs
  - A prediction for the next bit $p(\vec{X}) \in [0,1]$.
  - To ensure p has a finite description. Restrict to rational numbers n/m
- Any online prediction algorithm can be represented as code $\vec{b}(E)$ for U. The code length is $|\vec{b}(E)|$.

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0,1\}^t$
  - runs finite time and outputs
  - A prediction for the next bit $p(\vec{X}) \in [0,1]$.
  - To ensure p has a finite description. Restrict to rational numbers n/m
- Any online prediction algorithm can be represented as code $\vec{b}(E)$ for U. The code length is $|\vec{b}(E)|$.
- Most sequences do not correspond to valid prediction algorithms.

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0, 1\}^t$
  - runs finite time and outputs
  - A prediction for the next bit $p(\vec{X}) \in [0, 1]$.
  - To ensure p has a finite description. Restrict to rational numbers n/m
- Any online prediction algorithm can be represented as code $\vec{b}(E)$ for U. The code length is $|\vec{b}(E)|$.
- Most sequences do not correspond to valid prediction algorithms.
- $V(\vec{b}, \vec{X}, t) = 1$ if the program $\vec{b}$, given $\vec{X}$ as input, halts within t steps and outputs a well-formed prediction. Otherwise $V(\vec{b}, \vec{X}, t) = 0$

Standardizing online prediction algorithms

- Fix a universal Turing machine U.
- An online prediction algorithm E is a program that
  - given as input The past $\vec{X} \in \{0,1\}^t$
  - runs finite time and outputs
  - A prediction for the next bit $p(\vec{X}) \in [0,1]$.
  - To ensure p has a finite description. Restrict to rational numbers n/m
- Any online prediction algorithm can be represented as code $\vec{b}(E)$ for U. The code length is $|\vec{b}(E)|$.
- Most sequences do not correspond to valid prediction algorithms.
- $V(\vec{b},\vec{X},t) = 1$ if the program $\vec{b}$, given $\vec{X}$ as input, halts within t steps and outputs a well-formed prediction. Otherwise $V(\vec{b},\vec{X},t) = 0$
- $V(\vec{b},\vec{X},t)$ is computable (recursively enumerable).

A universal prediction machine

- Assign to the code $\vec{b}$ the initial weight $w_{\vec{b}}^1 = 2^{-|\vec{b}|-\log_2|\vec{b}|}$.

A universal prediction machine

- Assign to the code $\vec{b}$ the initial weight $w_{\vec{b}}^{1} = 2^{-|\vec{b}| - \log_2 |\vec{b}|}$.
- The total initial weight over all finite binary sequences is one.

A universal prediction machine

- Assign to the code $\vec{b}$ the initial weight $w^1_{\vec{b}} = 2^{-|\vec{b}|-\log_2 |\vec{b}|}$.
- The total initial weight over all finite binary sequences is one.
- Run the Bayes algorithm over “all” prediction algorithms.

A universal prediction machine

- Assign to the code $\vec{b}$ the initial weight $w^1_{\vec{b}} = 2^{-|\vec{b}|-\log_2 |\vec{b}|}$.
- The total initial weight over all finite binary sequences is one.
- Run the Bayes algorithm over “all” prediction algorithms.
- technical details: On iteration t, $|\vec{X}| = t$. Use the predictions of programs $\vec{b}$ such that $|\vec{b}| \leq t$ and for which $V(\vec{b}, \vec{X}, 2^t) = 1$. Assing the remaining mass the prediction $1/2$ (insuring a loss of 1)

Performance of the universal prediction algorithm

- Using $L_A \leq \min_i \left(L_i - \log w_i^1\right)$

Performance of the universal prediction algorithm

- Using $L_A \leq \min_i \left(L_i - \log w_i^1 \right)$
- Assume E is a prediction algorithm which generates the tth prediction in time smaller than 2^t

Performance of the universal prediction algorithm

- Using $L_A \leq \min_i \left(L_i - \log w_i^1\right)$
- Assume E is a prediction algorithm which generates the tth prediction in time smaller than 2^t
- When $t \leq |\vec{b}(E)|$ the algorithm is not used and thus it's loss is 1

Performance of the universal prediction algorithm

- Using $L_A \leq \min_i \left(L_i - \log w_i^1\right)$
- Assume E is a prediction algorithm which generates the tth prediction in time smaller than 2^t
- When $t \leq |\vec{b}(E)|$ the algorithm is not used and thus it's loss is 1
- We get that the loss of the Universal algorithm is at most $2|\vec{b}(E)| + \log_2 |\vec{b}(E)| + L_E$

Performance of the universal prediction algorithm

- Using $L_A \leq \min_i \left(L_i - \log w_i^1\right)$
- Assume E is a prediction algorithm which generates the tth prediction in time smaller than 2^t
- When $t \leq |\vec{b}(E)|$ the algorithm is not used and thus it's loss is 1
- We get that the loss of the Universal algorithm is at most $2|\vec{b}(E)| + \log_2 |\vec{b}(E)| + L_E$
- More careful analysis can reduce $2|\vec{b}(E)| + \log_2 |\vec{b}(E)|$ to $|\vec{b}(E)|$

Performance of the universal prediction algorithm

- Using $L_A \leq \min_i \left(L_i - \log w_i^1\right)$
- Assume E is a prediction algorithm which generates the tth prediction in time smaller than 2^t
- When $t \leq |\vec{b}(E)|$ the algorithm is not used and thus it's loss is 1
- We get that the loss of the Universal algorithm is at most $2|\vec{b}(E)| + \log_2 |\vec{b}(E)| + L_E$
- More careful analysis can reduce $2|\vec{b}(E)| + \log_2 |\vec{b}(E)|$ to $|\vec{b}(E)|$
- Code length is arbitrarily close to the Kolmogorov Complexity of the sequence.

Performance of the universal prediction algorithm

- Using $L_A \leq \min_i \left(L_i - \log w_i^1\right)$
- Assume E is a prediction algorithm which generates the tth prediction in time smaller than 2^t
- When $t \leq |\vec{b}(E)|$ the algorithm is not used and thus it's loss is 1
- We get that the loss of the Universal algorithm is at most $2|\vec{b}(E)| + \log_2 |\vec{b}(E)| + L_E$
- More careful analysis can reduce $2|\vec{b}(E)| + \log_2 |\vec{b}(E)|$ to $|\vec{b}(E)|$
- Code length is arbitrarily close to the Kolmogorov Complexity of the sequence.
- Ridiculously bad running time.

Bayes coding is better than two part codes

- Simple bound as good as bound for two part codes (MDL) but enables online compression

Bayes coding is better than two part codes

- Simple bound as good as bound for two part codes (MDL) but enables online compression
- Suppose we have K copies of each expert.

Bayes coding is better than two part codes

- Simple bound as good as bound for two part codes (MDL) but enables online compression
- Suppose we have K copies of each expert.
- Two part code has to point to one of the KN experts
  $L_A \leq \log NK + \min_i L_i^T = \log NK + \min_i L_i^T$

Bayes coding is better than two part codes

- Simple bound as good as bound for two part codes (MDL) but enables online compression
- Suppose we have K copies of each expert.
- Two part code has to point to one of the KN experts
  $L_A \leq \log NK + \min_i L_i^T = \log NK + \min_i L_i^T$
- If we use Bayes predictor + arithmetic coding we get:

$$L_A = -\log W^{T+1} \leq \log K \max_i \frac{1}{NK} e^{-L_i^T} = \log N + \min_i L_i^T$$

Bayes coding is better than two part codes

- Simple bound as good as bound for two part codes (MDL) but enables online compression
- Suppose we have K copies of each expert.
- Two part code has to point to one of the KN experts
  $L_A \leq \log NK + \min_i L_i^T = \log NK + \min_i L_i^T$
- If we use Bayes predictor + arithmetic coding we get:

$$L_A = -\log W^{T+1} \leq \log K \max_i \frac{1}{NK} e^{-L_i^T} = \log N + \min_i L_i^T$$

- We don't pay a penalty for copies.

Bayes coding is better than two part codes

- Simple bound as good as bound for two part codes (MDL) but enables online compression
- Suppose we have K copies of each expert.
- Two part code has to point to one of the KN experts $L_A \leq \log NK + \min_i L_i^T = \log NK + \min_i L_i^T$
- If we use Bayes predictor + arithmetic coding we get:

$$L_A = -\log W^{T+1} \leq \log K \max_i \frac{1}{NK} e^{-L_i^T} = \log N + \min_i L_i^T$$

- We don't pay a penalty for copies.
- More generally, the regret is smaller if many of the experts perform well.

The biased coins set of experts

- Each expert corresponds to a biased coin, predicts with a fixed $\theta \in [0, 1]$.

The biased coins set of experts

- Each expert corresponds to a biased coin, predicts with a fixed $\theta \in [0, 1]$.
- Set of experts is uncountably infinite.

The biased coins set of experts

- Each expert corresponds to a biased coin, predicts with a fixed $\theta \in [0, 1]$.
- Set of experts is uncountably infinite.
- Only countably many experts can be assigned non-zero weight.

The biased coins set of experts

- Each expert corresponds to a biased coin, predicts with a fixed $\theta \in [0, 1]$.
- Set of experts is uncountably infinite.
- Only countably many experts can be assigned non-zero weight.
- Instead, we assign the experts a Density Measure.

The biased coins set of experts

- Each expert corresponds to a biased coin, predicts with a fixed $\theta \in [0, 1]$.
- Set of experts is uncountably infinite.
- Only countably many experts can be assigned non-zero weight.
- Instead, we assign the experts a Density Measure.
- $L_A \leq \min_i \left(L_i - \log w_i^1\right)$ is meaningless.

The biased coins set of experts

- Each expert corresponds to a biased coin, predicts with a fixed $\theta \in [0, 1]$.
- Set of experts is uncountably infinite.
- Only countably many experts can be assigned non-zero weight.
- Instead, we assign the experts a Density Measure.
- $L_A \leq \min_i \left(L_i - \log w_i^1\right)$ is meaningless.
- Can we still get a meaningful bound?

Bayes Algorithm for biased coins

- Replace the initial weight by a density measure
  $w(\theta) = w^1(\theta)$, $\int_0^1 w(\theta) d\theta = 1$

Bayes Algorithm for biased coins

- Replace the initial weight by a density measure $w(\theta) = w^1(\theta)$, $\int_0^1 w(\theta)d\theta = 1$
- Relationship between final total weight and total log loss remains unchanged:

$$L_A = \ln \int_0^1 w(\theta) e^{-L_\theta^{T+1}} d\theta$$

Bayes Algorithm for biased coins

- Replace the initial weight by a density measure $w(\theta) = w^1(\theta)$, $\int_0^1 w(\theta) d\theta = 1$
- Relationship between final total weight and total log loss remains unchanged:

$$L_A = \ln \int_0^1 w(\theta) e^{-L_\theta^{T+1}} d\theta$$

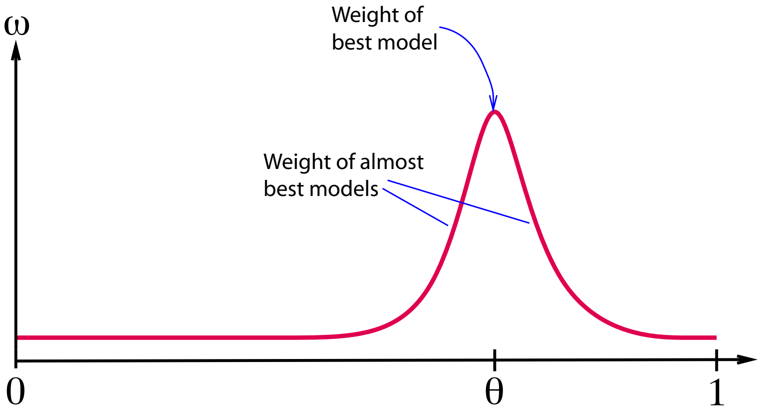
- We need a new lower bound on the final total weight

Main Idea

If $w^t(\theta)$ is large then $w^t(\theta + \epsilon)$ is also large.

Main Idea

If $w^t(\theta)$ is large then $w^t(\theta + \epsilon)$ is also large.



Expanding the exponent around the peak

- For log loss the best θ is empirical distribution of the seq.

$$\hat{\theta} = \frac{\#\{x^t = 1; \;\; 1 \leq t \leq T\}}{T}$$

Expanding the exponent around the peak

- For log loss the best θ is empirical distribution of the seq.

$$\hat{\theta} = \frac{\#\{x^t = 1; \;\; 1 \le t \le T\}}{T}$$

- The total loss scales with T

$$L_\theta = T \cdot (\hat{\theta}\ell(\theta, 1) + (1 - \hat{\theta})\ell(\theta, 0)) \doteq T \cdot g(\hat{\theta}, \theta)$$

Expanding the exponent around the peak

- For log loss the best θ is empirical distribution of the seq.

$$\hat{\theta} = \frac{\#\{x^t = 1; \;\; 1 \leq t \leq T\}}{T}$$

- The total loss scales with T

$$L_\theta = T \cdot (\hat{\theta}\ell(\theta, 1) + (1 - \hat{\theta})\ell(\theta, 0)) \doteq T \cdot g(\hat{\theta}, \theta)$$

Expanding the exponent around the peak

- For log loss the best θ is empirical distribution of the seq.

$$\hat{\theta} = \frac{\#\{x^t = 1; \;\; 1 \leq t \leq T\}}{T}$$

- The total loss scales with T

$$L_\theta = T \cdot (\hat{\theta}\ell(\theta, 1) + (1 - \hat{\theta})\ell(\theta, 0)) \doteq T \cdot g(\hat{\theta}, \theta)$$

$$L_A - L_{\min} \quad \leq \quad \ln \int_0^1 w(\theta) e^{-L_\theta} d\theta - \ln e^{L_{\min}}$$

Expanding the exponent around the peak

- For log loss the best θ is empirical distribution of the seq.

$$\hat{\theta} = \frac{\#\{x^t = 1;\ \ 1 \leq t \leq T\}}{T}$$

- The total loss scales with T

$$L_\theta = T \cdot (\hat{\theta}\ell(\theta, 1) + (1 - \hat{\theta})\ell(\theta, 0)) \doteq T \cdot g(\hat{\theta}, \theta)$$

$$\begin{aligned} L_A - L_{\min} &\leq \ln \int_0^1 w(\theta) e^{-L_\theta} d\theta - \ln e^{L_{\min}} \\ &= \ln \int_0^1 w(\theta) e^{-(L_\theta - L_{\min})} d\theta \\ pause &= \ln \int_0^1 w(\theta) e^{T(g(\hat{\theta},\theta) - g(\hat{\theta},\hat{\theta}))} d\theta \end{aligned}$$

Laplace approximation (idea)

- Taylor expansion of $g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta})$ around $\theta = \hat{\theta}$.

Laplace approximation (idea)

- Taylor expansion of $g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta})$ around $\theta = \hat{\theta}$.
- First and second terms in the expansion are zero.

Laplace approximation (idea)

- Taylor expansion of $g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta})$ around $\theta = \hat{\theta}$.
- First and second terms in the expansion are zero.
- Third term gives a quadratic expression in the exponent

Laplace approximation (idea)

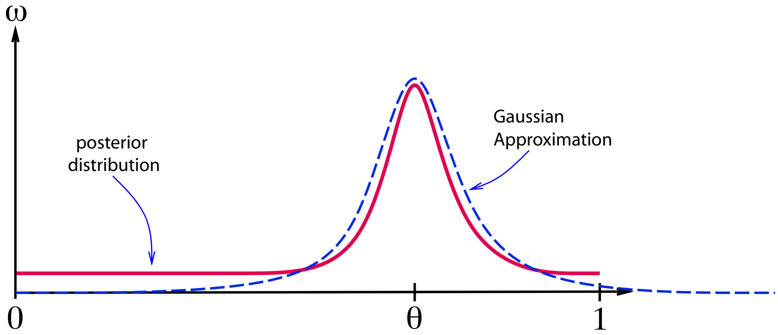
- Taylor expansion of $g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta})$ around $\theta = \hat{\theta}$.
- First and second terms in the expansion are zero.
- Third term gives a quadratic expression in the exponent
- $\Rightarrow$ a gaussian approximation of the posterior.

Laplace approximation (idea)

- Taylor expansion of $g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta})$ around $\theta = \hat{\theta}$.
- First and second terms in the expansion are zero.
- Third term gives a quadratic expression in the exponent
- $\Rightarrow$ a gaussian approximation of the posterior.

Laplace approximation (idea)

- Taylor expansion of $g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta})$ around $\theta = \hat{\theta}$.
- First and second terms in the expansion are zero.
- Third term gives a quadratic expression in the exponent
- $\Rightarrow$ a gaussian approximation of the posterior.



Laplace Approximation (details)

$$\int_0^1 w(\theta) e^{T(g(\hat{\theta},\theta)-g(\hat{\theta},\hat{\theta}))} d\theta$$

Laplace Approximation (details)

$$\int_0^1 w(\theta) e^{T(g(\hat{\theta},\theta)-g(\hat{\theta},\hat{\theta}))} d\theta$$

$$= \quad w(\hat{\theta}) \sqrt{\frac{-2\pi}{T \left. \frac{d^2}{d\theta^2} \right|_{\theta=\hat{\theta}} (g(\hat{\theta},\theta) - g(\hat{\theta},\hat{\theta}))}} + O(T^{-3/2})$$

Choosing the optimal prior

- Choose $w(\theta)$ to maximize the worst-case final total weight

$$\min_{\hat{\theta}} w(\hat{\theta}) \sqrt{\frac{-2\pi}{T \left.\frac{d^2}{d\theta^2}\right|_{\theta=\hat{\theta}} (g(\hat{\theta},\theta) - g(\hat{\theta},\hat{\theta}))}}$$

Choosing the optimal prior

- Choose $w(\theta)$ to maximize the worst-case final total weight

$$\min_{\hat{\theta}} w(\hat{\theta}) \sqrt{\frac{-2\pi}{T \left.\frac{d^2}{d\theta^2}\right|_{\theta=\hat{\theta}} (g(\hat{\theta},\theta) - g(\hat{\theta},\hat{\theta}))}}$$

- Make bound equal for all $\hat{\theta} \in [0,1]$ by choosing

$$w^*(\hat{\theta}) = \frac{1}{Z} \sqrt{\frac{\left.\frac{d^2}{d\theta^2}\right|_{\theta=\hat{\theta}} (g(\hat{\theta},\theta) - g(\hat{\theta},\hat{\theta}))}{-2\pi}} ,$$

where Z is the normalization factor:

$$Z = \sqrt{\frac{1}{2\pi}} \int_0^1 \sqrt{\left.\frac{d^2}{d\theta^2}\right|_{\theta=\hat{\theta}} (g(\hat{\theta},\hat{\theta}) - g(\hat{\theta},\theta))} \; d\hat{\theta}$$

The bound for the optimal prior

- Plugging in we get

$$
\begin{aligned}
L_A - L_{\min} &\leq \ln \int_0^1 w^*(\theta) e^{T(g(\hat{\theta},\theta)-g(\hat{\theta},\hat{\theta}))} d\theta \\
&= \ln \left(\sqrt{\frac{2\pi Z}{T}} + O(T^{-3/2}) \right) \\
&= \frac{1}{2} \ln \frac{T}{2\pi} - \frac{1}{2} \ln Z + O(1/T) .
\end{aligned}
$$

Solving for log-loss

- The exponent in the integral is

$$g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta}) = \hat{\theta} \ln \frac{\hat{\theta}}{\theta} + (1 - \hat{\theta}) \ln \frac{1 - \hat{\theta}}{1 - \theta} = D_{KL}(\hat{\theta} || \theta)$$

Solving for log-loss

- The exponent in the integral is

$$g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta}) = \hat{\theta} \ln \frac{\hat{\theta}}{\theta} + (1 - \hat{\theta}) \ln \frac{1 - \hat{\theta}}{1 - \theta} = D_{KL}(\hat{\theta} || \theta)$$

- The second derivative

$$\left. \frac{d^2}{d\theta^2} \right|_{\theta = \hat{\theta}} D_{KL}(\hat{\theta} || \theta) = \hat{\theta}(1 - \hat{\theta})$$

Is called the empirical Fisher information

Solving for log-loss

- The exponent in the integral is

$$g(\hat{\theta}, \theta) - g(\hat{\theta}, \hat{\theta}) = \hat{\theta} \ln \frac{\hat{\theta}}{\theta} + (1 - \hat{\theta}) \ln \frac{1 - \hat{\theta}}{1 - \theta} = D_{KL}(\hat{\theta} || \theta)$$

- The second derivative

$$\left. \frac{d^2}{d\theta^2} \right|_{\theta = \hat{\theta}} D_{KL}(\hat{\theta} || \theta) = \hat{\theta}(1 - \hat{\theta})$$

  Is called the empirical Fisher information
- The optimal prior:

$$w^*(\hat{\theta}) = \frac{1}{\pi \sqrt{\hat{\theta}(1 - \hat{\theta})}}$$

  Known in general as Jeffrey's prior. And, in this case, the Dirichlet-$(1/2, 1/2)$ prior.

The cumulative log loss of Bayes using Jeffrey's prior

- $$L_A - L_{\min} \leq \frac{1}{2}\ln(T+1) + \frac{1}{2}\ln\frac{\pi}{2} + O(1/T)$$

But what is the prediction rule?

- As luck would have it the Dirichlet prior is the conjugate prior for the Binomial distribution.

But what is the prediction rule?

- As luck would have it the Dirichlet prior is the conjugate prior for the Binomial distribution.
- Observed t bits, n of which were 1. The posterior is:

$$\frac{1}{Z\sqrt{\theta(1-\theta)}}\theta^{n}(1-\theta)^{t-n} = \frac{1}{Z}\theta^{n-1/2}(1-\theta)^{t-n-1/2}$$

But what is the prediction rule?

- As luck would have it the Dirichlet prior is the conjugate prior for the Binomial distribution.
- Observed t bits, n of which were 1. The posterior is:

$$\frac{1}{Z\sqrt{\theta(1-\theta)}}\theta^n(1-\theta)^{t-n} = \frac{1}{Z}\theta^{n-1/2}(1-\theta)^{t-n-1/2}$$

- The posterior average is:

$$\frac{\int_0^1 \theta^{n+1/2}(1-\theta)^{t-n-1/2}d\theta}{\int_0^1 \theta^{n-1/2}(1-\theta)^{t-n-1/2}d\theta} = \frac{n+1/2}{t+1}$$

But what is the prediction rule?

- As luck would have it the Dirichlet prior is the conjugate prior for the Binomial distribution.
- Observed t bits, n of which were 1. The posterior is:

$$\frac{1}{Z\sqrt{\theta(1-\theta)}}\theta^{n}(1-\theta)^{t-n} = \frac{1}{Z}\theta^{n-1/2}(1-\theta)^{t-n-1/2}$$

- The posterior average is:

$$\frac{\int_0^1 \theta^{n+1/2}(1-\theta)^{t-n-1/2}d\theta}{\int_0^1 \theta^{n-1/2}(1-\theta)^{t-n-1/2}d\theta} = \frac{n+1/2}{t+1}$$

- This is called the Trichevsky Trofimov prediction rule.

Laplace Rule of Succession

- Laplace suggested using the uniform prior, which is also a conjugate prior.

Laplace Rule of Succession

- Laplace suggested using the uniform prior, which is also a conjugate prior.
- In this case the posterior average is:

$$\frac{\int_0^1 \theta^{n+1}(1-\theta)^{t-n} d\theta}{\int_0^1 \theta^{n}(1-\theta)^{t-n} d\theta} = \frac{n+1}{t+2}$$

Laplace Rule of Succession

- Laplace suggested using the uniform prior, which is also a conjugate prior.
- In this case the posterior average is:

$$\frac{\int_0^1 \theta^{n+1}(1-\theta)^{t-n} d\theta}{\int_0^1 \theta^{n}(1-\theta)^{t-n} d\theta} = \frac{n+1}{t+2}$$

- The bound on the cumulative log loss is worse:

$$L_A - L_{\min} = \ln T + O(1)$$

Laplace Rule of Succession

- Laplace suggested using the uniform prior, which is also a conjugate prior.
- In this case the posterior average is:

$$\frac{\int_0^1 \theta^{n+1}(1-\theta)^{t-n}d\theta}{\int_0^1 \theta^{n}(1-\theta)^{t-n}d\theta} = \frac{n+1}{t+2}$$

- The bound on the cumulative log loss is worse:

$$L_A - L_{\min} = \ln T + O(1)$$

- Suffers larger regret when $\hat{\theta}$ is far from $1/2$

Shtarkov Lower bound

- What is the optimal prediction when T is know in advance?

Shtarkov Lower bound

- What is the optimal prediction when T is know in advance?
- $$L_*^T - \min_\theta L_\theta^T \geq \frac{1}{2}\ln(T+1) + \frac{1}{2}\ln\frac{\pi}{2} - O(\frac{1}{\sqrt{T}})$$

Multinomial Distributions

- For a distribution over k elements (Multinomial) [Xie and Barron]

Multinomial Distributions

- For a distribution over k elements (Multinomial) [Xie and Barron]
- Use the add 1/2 rule (KT).

$$p(i) = \frac{n_i + 1/2}{t + k/2}$$

Multinomial Distributions

- For a distribution over k elements (Multinomial) [Xie and Barron]
- Use the add 1/2 rule (KT).

$$p(i) = \frac{n_i + 1/2}{t + k/2}$$

- Bound is

$$L_A - L_{\min} \leq \frac{k-1}{2} \ln T + C + o(1)$$

Multinomial Distributions

- For a distribution over k elements (Multinomial) [Xie and Barron]
- Use the add 1/2 rule (KT).

$$p(i) = \frac{n_i + 1/2}{t + k/2}$$

- Bound is

$$L_A - L_{\min} \leq \frac{k-1}{2} \ln T + C + o(1)$$

- The constant C is optimal.

Exponential Distributions

- For any set of distributions from the exponential family defined by k parameters (Some technical conditions on closure of set??) [Rissanen??]

Exponential Distributions

- For any set of distributions from the exponential family defined by k parameters (Some technical conditions on closure of set??) [Rissanen??]
- Use Bayes Algorithm with Jeffrey's prior:

$$w^*(\hat{\theta}) = \frac{1}{Z} \frac{1}{\sqrt{\left| \mathbf{H}\left(D_{KL}(\hat{\theta} || \theta) \right) \Big|_{\theta=\hat{\theta}} \right|}}$$

H denotes the Hessian.

Exponential Distributions

- For any set of distributions from the exponential family defined by k parameters (Some technical conditions on closure of set??) [Rissanen??]
- Use Bayes Algorithm with Jeffrey's prior:

$$w^*(\hat{\theta}) = \frac{1}{Z} \frac{1}{\sqrt{\left| \mathbf{H}\left(D_{KL}(\hat{\theta}||\theta) \right) \Big|_{\theta=\hat{\theta}} \right|}}$$

H denotes the Hessian.

- $$L_A - L_{\min} \le \frac{k-1}{2} \ln T - \ln Z + o(1)$$

General Distributions

- Characterize distribution family by metric entropy.

General Distributions

- Characterize distribution family by metric entropy.
- Fixed parameter set usually corresponds to polynomial metrix entropy

$$N(1/\epsilon) = O\left(\frac{1}{\epsilon^d}\right)$$

d is the number of parameters.

General Distributions

- Characterize distribution family by metric entropy.
- Fixed parameter set usually corresponds to polynomial metrix entropy
$$N(1/\epsilon) = O\left(\frac{1}{\epsilon^{d}}\right)$$
d is the number of parameters.
- [Haussler and Opper] show that the coefficient in front of $\ln T$ is optimal for distribution families where the metric entropy is up to
$$N(1/\epsilon) = O\left(e^{\epsilon^{-\alpha}}\right)$$
For all $\alpha \leq 5/2$.

next Class

- Variable-length markov models - a set of distributions with increasing number of parameters.

next Class

- Variable-length markov models - a set of distributions with increasing number of parameters.
- THe context algorithm: An efficient implementation of the Bayes algorithm which achieves close-to-optimal worst case bounds.