In The Name of God

Sharif University of Technology

Electrical Engineering Department

Deep Generative Models

Assignment 1

Fall 2024

Instructor: Dr. S. Amini

Due on Mehr 26, 1403 at 23:55

Matin M.babaei 400102114

1 Autoregressive Models of Order p

In this question, you will explore the process of Maximum Likelihood Estimation (MLE) for linear Autoregressive (AR) models. Given a time series dataset, your task is to estimate the parameters of an AR model using MLE.

An Autoregressive (AR) model of order p (denoted as AR(p)) is used to describe a time-dependent process and is a classical linear model. The value at any given time is expressed as a linear function of its previous values plus some error term.

This model can be written as:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t$$

where:

- y_t is the value at time t,
- $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters (coefficients) of the model,
- ϵ_t is the error term at time t, assumed to be independently and identically distributed (i.i.d.) with a normal distribution $N(0, \sigma^2)$.

Now, you are supposed to estimate the parameters $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2 that maximize the likelihood function given the observed data $y_1, y_2, \ldots, y_n$.

1.1 Log-Likelihood Function

Formulate the likelihood function for the AR(p) model. Then, derive the log-likelihood function.

1.2 Maximum Likelihood Estimation

Find the parameter values that maximize the log-likelihood function and provide the final estimates for $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2. During your procedure, derive detailed derivation of the likelihood and log-likelihood functions.

1.1) $\epsilon_t = 1\, y_t - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \cdots - \phi_p y_{t-p}$

$\epsilon_t \sim \sim (0, \sigma^2)$:

$$f(y_t \mid \{\phi_i\}_1^p, \sigma^2) = f(\epsilon_t \mid \{\phi_i\}_1^p, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_t - \sum_{i=1}^{p} \phi_i y_{t-i})^2}{2\sigma^2}\right)$$

why? $p(y_1, \ldots, y_n) = \prod_{t=1}^{n} p(y_t \mid y_{t-1}, \ldots, y_1)$

* دقت کنید صرفاً داده های مشاهده شده $\{y_1, y_2, \ldots, y_n\}$ بیان شده است، بنابراین از $y_0 = const$ چشم پوشی کردیم
بدون آن مشکلی نیز ایجاد نمی کند، صرفاً $t=p$ به $t=p+1$ تبدیل خواهد شد.

* $E[y_k \mid y_{k-1}, \dots y_{k-p}] = \sum_{i=1}^{p} \phi_i y_{k-i}$ $\quad Var[y_k \mid y_{k-1}, \dots y_{k-p}] = \sigma^2$

$$= \prod_{t=p+1}^{n} P(y_t \mid y_{t-1}, \dots y_{t-p}) \overset{*}{=} \prod_{t=1}^{N} N\left(\sum_{i=1}^{p} \phi_i y_{t-i}, \sigma^2\right)$$

$$L(\{\phi_i\}_1^p, \sigma^2) = \prod_{t=p}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_t - \sum_{i=1}^{p} \phi_i y_{t-i})^2}{2\sigma^2}\right)$$

$$\log L(\{\phi_i\}_1^p, \sigma^2) = \sum_{t=p}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \sum_{t=p}^{n} \frac{(y_t - \sum_{i=1}^{p} \phi_i y_{t-i})^2}{2\sigma^2}$$

So:

$$LL(\{\phi_i\}_1^p, \sigma^2) = -\frac{n-p}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=p+1}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i}\right)^2$$

1.2)

Maximizing Log-Likelihood:

$$\frac{\partial \log L}{\partial \sigma^2} = 0 \rightarrow -\frac{n-p}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \overbrace{\sum_{t=p}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i}\right)^2}^{A} = 0$$

$$\rightarrow -(n-p)\sigma^2 + A = 0 \rightarrow \sigma^{2*} = \frac{A}{n-p} = \frac{\sum_{t=p}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i}\right)^2}{n-p}$$

$$\frac{\partial LL}{\partial \phi_j} = 0 \rightarrow -\frac{1}{2\sigma^2} \sum_{t=p}^{n} \frac{\partial}{\partial \phi_j} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i}\right)^2 = 0$$

$$\sum_{t=p}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i}\right) \left(-\frac{\partial}{\partial \phi_j} \sum_{i=1}^{p} \phi_i y_{t-i}\right) = 0$$

$$\rightarrow \sum_{t=p}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i}\right) \left(-y_{t-j}\right) = 0$$

$$\rightarrow \sum_{t=p}^{n} y_{t-j} y_t = \sum_{i=1}^{p} \phi_i \sum_{t=p}^{n} y_{t-i} y_{t-j}$$

So we have: $R\Phi = r$

- $R \rightarrow$ Autocorrelation matrix $\rightarrow R_{i,j} = \sum_{t=p}^{n} y_{t-i}\, y_{t-j}$
- $\Phi = (\phi_1, \phi_2, \dots \phi_p)$
- $r \rightarrow$ Correlation between y_t and lagged values:

$$\rightarrow r = \left(\sum_{t=p}^{n} y_t y_{t-1},\ \sum_{t=p}^{n} y_t y_{t-2},\ \dots\ \sum_{t=p}^{n} y_t y_{t-p} \right)$$

So, now how we solve for Φ?

Note: $\Phi = R^{-1} r \rightarrow \Phi$ ✓ $\rightarrow \{\phi_i\}_{i=1}^{p}$ ✓

به شکل تا صریح هم می‌توان گفت:

$$\forall j;\ \frac{\partial L(\theta)}{\partial \phi_j} = \frac{y_{t-j}}{\sigma^2} \sum_{t=p+1}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i} \right) = 0$$

$$\rightarrow y_{t-j} = 0 \quad \text{یا} \quad \sum_{t=p+1}^{n} \left(y_t - \sum_{i=1}^{n} \phi_i y_{t-i} \right) = 0$$

2 Autoregressive Models

Given a sequence of random variables $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, where each x_t is a continuous-valued random variable, consider the autoregressive (AR) model where the joint probability distribution of the sequence is factorized as:

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)\cdots p(x_T|x_1, x_2, \ldots, x_{T-1})$$

Now, assume that the conditional distributions $p(x_t|x_1, x_2, \ldots, x_{t-1})$ for all $t \in \{1, \ldots, T\}$ are modeled by a neural network f_θ with parameters θ such that:

$$p(x_t|x_1, x_2, \ldots, x_{t-1}) = \mathcal{N}(f_\theta(x_1, x_2, \ldots, x_{t-1}), \sigma^2)$$

where $\mathcal{N}(\mu, \sigma^2)$ represents a Gaussian distribution with mean μ and fixed variance $\sigma^2 > 0$.

2.1 Maximum Likelihood Estimation

Derive the log-likelihood function for the joint distribution $p(\mathbf{x})$ in terms of θ, and express how you would compute gradients for training the neural network f_θ using backpropagation.

2.2 Predictive Sampling

Describe the process for generating a new sequence $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_T)$ from the model by sampling from the conditional distributions. Explicitly account for how the autoregressive structure influences the sampling process.

2.3 KL Divergence between AR Models

Consider two autoregressive models, one parameterized by f_θ and another by g_ϕ. Write down the expression for the Kullback-Leibler (KL) divergence between the two distributions $p_\theta(\mathbf{x})$ and $q_\phi(\mathbf{x})$. Discuss the computational challenges of directly evaluating this divergence and propose a practical approximation method for high-dimensional sequences.

Hint: You can describe *Monte Carlo* sampling here.

2.4 Stationarity and Long-Term Dependencies

Analyze whether the autoregressive model structure as described can capture long-term dependencies in the sequence. If it cannot, provide a mathematical explanation. Suggest a modification to the autoregressive model (e.g., using a recurrent neural network or Transformer) that might address this limitation and discuss its probabilistic interpretation.

2.5 Bonus

Given the autoregressive factorization, if you observe that the model's predictive performance degrades for longer sequences, hypothesize potential reasons for this degradation. Provide at least two hypotheses grounded in the model's probabilistic structure and optimization process, and suggest corresponding solutions.

2.1)

$$p(x) = p(x_1)\, p(x_2|x_1) \ldots p(x_T|x_1, x_2, \ldots x_{T-1})$$

$$* \; p(x) = N(\mu = f_\theta(x_1, \ldots x_{t-1}), \sigma^2)$$

$$\log p(x) = \sum_{t=1}^{T} \log p(x_t|x_1, x_2, \ldots x_{t-1})$$

$$p(x_t|x_1, x_2, \ldots x_{t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_t - f_\theta(x_1, x_2, \ldots x_{t-1}))^2}{2\sigma^2}\right)$$

So:

$$\log p(x) = -\frac{T}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(x_t - f_\theta(x_1, x_2, \dots, x_{t-1}))^2$$

- Backpropagation: Maximizing the log-likelihood w.r.t θ can be done using GD, the gradient of the log-likelihood w.r.t θ can be computed as:

$$\frac{\partial \log p(x)}{\partial \theta} = \frac{1}{\sigma^2}\sum_{t=1}^{T}(x_t - f_\theta(x_1, x_2, \dots x_{t-1}))\frac{\partial f_\theta(x_1, x_2 \dots x_{t-1})}{\partial \theta}$$

2.2)

Sampling Process:

1) Initialize x'_1 by sampling from prior $p(x_1)$ (may be $N(0,1)$)

2) At step t, Sample using different samplings from

$$p(x'_t \mid x'_1, x'_2, \dots x'_{t-1})$$

This conditional distribution is parametrized by $f_\theta(x'_1, x'_2, \dots x'_{t-1})$ and fixed σ^2. In other words, $x'_t \sim N(f_\theta(x'_1, x'_2, \dots x'_{t-1}), \sigma^2)$

In autoregressive models, each sample at time t depends on all previous samples. This creates a dependency chain, which can influence how errors propagate through the sequence. Errors in early prediction can compound in later time steps, affecting long time accuracy.

2.3)

KL-Divergence between 2 joint distributions:

$$D_{KL}(f_\theta \| q_\varphi) = E_{f_\theta(x)}\left[\log \frac{f_\theta(x)}{q_\phi(x)}\right]$$

$$D_{KL}(f_\theta \| q_\varphi) = \sum_{t=1}^{T} E_{f_\theta(x_{1:t})}\left[\log \frac{f_\theta(x_t | x_{t-1}, \dots x_1)}{q_\varphi(x_t | x_{t-1}, \dots x_1)}\right]$$

Using Monte-Carlo Est:

$$D_{KL}(f_\theta \| q_\phi) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\left[\log \frac{f_\theta(x_t^{(n)} | x_{t-1}^{(n)}, \dots x_1^{(n)})}{q_\varphi(x_t^{(n)} | x_{t-1}^{(n)}, \dots x_1^{(n)})}\right]$$

Computing this phrase can be computationally intractable, Esp in high-dimensional sequences. Instead you can approximate the expectation using Monte Carlo Sampling.

Note: we are not given anything about f_θ, q_φ but if we know $f_\theta \sim \mathcal{N}(\mu_1, \sigma^2)$, $q_\phi \sim \mathcal{N}(\mu_2, \sigma^2)$:

$$D_{KL}(f_\theta \| q_\varphi) = E_{x\sim f_\theta}\left(\log \frac{f_\theta(x)}{q_\phi(x)}\right) = E_{x\sim f_\theta}\left[\log \frac{f_\theta(x_1)\prod_{k=2}^{T} f_\theta(x_k | x_{k-1}, \dots x_1)}{q_\phi(x_1)\prod_{k=2}^{T} q_\phi(x_k | x_{k-1}, \dots x_1)}\right]$$

$$= \mathop{E}_{x\sim f_\theta(x)} \log\left[\frac{f_\theta(x_1)}{q_\phi(x_1)}\right] + \mathop{E}_{x\sim f_\theta}\left[\sum_{k=2}^{T} \log\left(\frac{f_\theta(x_k|x_{k-1}\dots x_1)}{q_\phi(x_k|x_{k-1}\dots x_1)}\right)\right]$$

$$= \mathop{E}_{x_1\sim f_\theta(x)} D_{KL}(f_\theta(x_1)\|q_\phi(x_1)) + \sum_{k=2}^{T} \mathop{E}_{x_1,\dots x_{k-1}\sim f_\theta(x)} \left[D_{KL}(f_\theta(x_k|x_{k-1},\dots x_1)\|q_\phi(x_k|x_{k-1},\dots x_1))\right]$$

$$= \mathop{E}_{x_1,\dots x_{k-1}\sim f_\theta(x)} \frac{1}{2\sigma^2}\sum_{k=1}^{T}(\mu_1-\mu_2)^2$$

* $\mu_1-\mu_2 = f_\theta(x_1,\dots x_{k-1}) - g_\phi(x_1,\dots x_{k-1})$

what we do for estimation is Monte Carlo estimation:

$$D_{KL}(f_\theta\|q_\phi) \simeq \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2\sigma^2}\sum_{k=1}^{T}\left(f_\theta(x_1^{(n)},\dots x_{k-1}^{(n)}) - g_\phi(x_1^{(n)},\dots x_{k-1}^{(n)})\right)^2$$

* Computation is still soo hard!

Q. 4)

First consider the structure of the model, in base AR models, the dependency of x_t on past values decreases with time due to some factors:

1) Vanishing Gradient: In NNs, when we

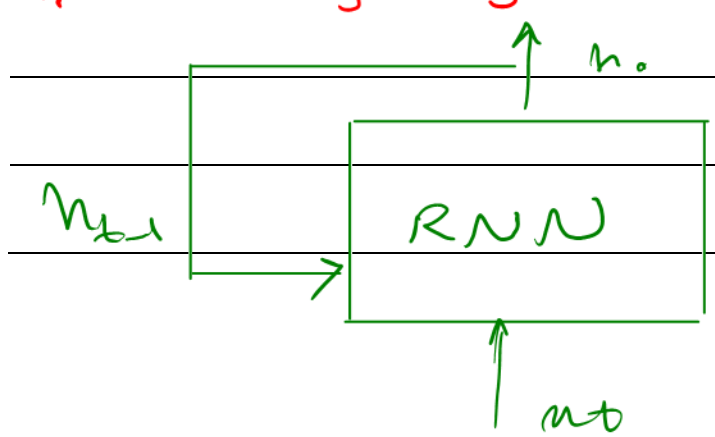
backpropagate through long sequences, the gradients vanishes, leading to weaker long term dependencies (Disscused in ML course in detail due to activation functions)

2.) Memory Constraints: Model is somehow biased of last values and weighs them more than far-past values if it doesn't have a memory mechanism.

How to come over these challenges?

To address this issues, we can use a RNN structure. Each RNN model maintains a hidden state that encapsulates past infos.

Rather than referencing the whole history of observations $P(x_k | x_{k-1} \dots x_1)$, it is based on a compressed version known as the Hidden state $P(x_k | h_{k-1})$ which is updated regularly.



$$P(x_t) = \prod_{t=1}^{T} P(x_t | h_t)$$

$$h_t = f(h_{t-1}, x_t)$$

2.6)

My hypotheses:

1) Error Propagation:

Solution: Techniques like beam search or using an ensemble of models could mitigate the Error propagation.

2) Exploding Uncertainty

Solution: Incorporating a regularization technique that controls the variance of predictions (Dropout or Bayesian Methods)

As we discussed:

Vanishing Gradients - Memory Constraints

3 Real NADE Parameters

In this problem, you will study an extension of the Real NADE model. Recall that, given an autoregressive model

$$p(\boldsymbol{x}) = p(x_1)p(x_2 \mid \boldsymbol{x}_{<2}) \dots p(x_i \mid \boldsymbol{x}_{<i}) \dots p(x_n \mid \boldsymbol{x}_{<n}), \tag{1}$$

and Real NADE models the conditional distribution as

$$p(x_1) = \mathcal{N}(x_1 \mid \mu_1, \exp(s_1)) \tag{2}$$

$$\dots$$

$$p(x_i \mid \boldsymbol{x}_{<i}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{v}_i, b_i, \boldsymbol{u}_i, d_i) = \mathcal{N}\left(x_i \mid \boldsymbol{v}_i^\top \boldsymbol{h}_i + b_i, \exp(\boldsymbol{u}_i^\top \boldsymbol{h}_i + d_i)\right), \tag{3}$$

where $\boldsymbol{h}_i, \boldsymbol{c}, \boldsymbol{v}_i, \boldsymbol{u}_i \in \mathbb{R}^d$. Now, we would like to make $p(x_i \mid \boldsymbol{x}_{<i})$ follow a mixture of Gaussian

$$p(x_i \mid \boldsymbol{x}_{<i}) = \sum_{c=1}^{C} \pi_i^c \mathcal{N}(\mu_i^c, (\sigma_i^c)^2), \tag{4}$$

where $\sum_{c=1}^{C} \pi_i^c = 1$.

Now the question is: How do you propose to parameterize $\pi_i^c, \mu_i^c, \sigma_i^c, \forall c \in \{1, \dots, C\}$ as a function of $\boldsymbol{h}_i$? Describe the parameters required and the total number of parameters required for a single $p(x_i \mid \boldsymbol{x}_{<i})$.

To parametrize π_i^c, μ_i^c and σ_i^c as a function of h_i:

1) Mixing Coeffient: using a softmax over C outputs:

$$\pi_i^c = \frac{\exp(W_c^T h_i + b_i^c)}{\sum_{k=1}^{C} \exp(W_c^T h_i + b_k^c)}$$

$W \in \mathbb{R}^{C \times D}$

number of parameters for π_i^c: $C \times D + D$

2) Mean of Gaussian Components:

$$\mu_i^c = (u_i^c)^T h_i + b_i^c$$

number of parameters for μ_i^c: $C \times D + D$

3) STD of each Guassian component:

$$\sigma_i^c = \exp((u_i^c)^T h_i + d_i^c)$$

num of params for σ_i^c : $C \times d + C$

Total : $(C \times d + C) + (C \times d + C) + (C \times d + C) = 3C \times d + 3C$

Check with Articles :

Finally, the mixture of Gaussians parameters for the d-th conditional, $\boldsymbol{\theta}_d = \{\boldsymbol{\alpha}_d, \boldsymbol{\mu}_d, \boldsymbol{\sigma}_d\}$, are set by:

$$K \text{ mixing fractions,} \qquad \boldsymbol{\alpha}_d = \text{softmax}\left(\boldsymbol{V}_d^{\alpha\top}\boldsymbol{h}_d + \boldsymbol{b}_d^{\alpha}\right) \tag{6}$$

$$K \text{ component means,} \qquad \boldsymbol{\mu}_d = \boldsymbol{V}_d^{\mu\top}\boldsymbol{h}_d + \boldsymbol{b}_d^{\mu} \tag{7}$$

$$K \text{ component standard deviations,} \qquad \boldsymbol{\sigma}_d = \exp\left(\boldsymbol{V}_d^{\sigma\top}\boldsymbol{h}_d + \boldsymbol{b}_d^{\sigma}\right), \tag{8}$$

where free parameters $\boldsymbol{V}_d^{\alpha}$, $\boldsymbol{V}_d^{\mu}$, $\boldsymbol{V}_d^{\sigma}$ are $H \times K$ matrices, and $\boldsymbol{b}_d^{\alpha}$, $\boldsymbol{b}_d^{\mu}$, $\boldsymbol{b}_d^{\sigma}$ are vectors of size K. The *softmax* [18] ensures the mixing fractions are positive and sum to one, the exponential ensures the standard deviations are positive.

4 Monte Carlo Estimation

4.1 A Quick Warm Up

What is the value of this expectation $\mathbb{E}_{x\sim\mathcal{N}(0,2)}\left[x^2+x+1\right]$? How do you propose to estimate this quantity with Monte Carlo estimation?

4.2 Variance of K-sample Estimator

What is the variance of the Monte Carlo estimator using K samples?

4.3 Objective Minimization

Now assume we are interested in minimizing the objective

$$F(\theta)=\sum_{n=1}^{N} w_n f(\theta;n)+\lambda R(\theta), \text{where } w_n>0, \forall n.$$

What is the asymptotic complexity of evaluating $F(\theta)$ and $\nabla F(\theta)$ given θ (assume the function call and gradient evaluation of $f(\theta;n)$ is constant)? How do you propose to use Monte Carlo estimation to acquire an *unbiased* estimation of the objective and its gradient?

4.1)

$x \sim N(0,2)$:

$E[(x-\mu)^2] = \sigma^2 = 2 \longrightarrow E[x^2] = \sigma^2 = 2$

$E[x] = \mu = 0 \longrightarrow E[x] = 0$

$E[x^2+x+1] = \underset{x}{E}[x^2] + \underset{x}{E}[x] + 1 = 2 + 0 + 1 = 3$

$x \sim N(0,2)$

4.2) $\forall n$; $Var(f(x)) = E[f(x)^2] - (E[f(x)])^2$

Monte Carlo Estimation $\longrightarrow E[g(x)] \approx \frac{1}{N}\sum_{i}^{N} g(x_i)$

Using K samples:

$\longrightarrow Var(f(x)) \approx \frac{1}{K}\left(\sum_{k=1}^{K}(f(x_k))^2\right) - \left(\frac{1}{K}\sum_{k=1}^{K} f(x_k)\right)^2$

First I thought that 4.1 & 4.2 are independant but then I through maybe the variance wanted is actually the variance of the last

part. Assuming this, I continue to solve it.

$Y = X^2 + X + 1$

$$\mathrm{Var}[Y] = E[(Y - \underbrace{E[Y]}_{3})^2] = E[Y^2 + 9 - 6Y]$$

$$= E[Y^2] - 6E[Y] + 9$$

$$= E[Y^2] - 9$$

Note

$$\boxed{E\left[X^{2n}\right] = \sigma^{2n}(2n-1)(2n-3)\cdots(3)}$$

$$E[Y^2] = E[(X^2+X+1)(X^2+X+1)]$$

$$= E[X^4 + 2X^3 + 3X^2 + 2X + 1]$$

$$= E[X^4] + 2\underbrace{E[X^3]}_{0} + 3E[X^2] + 2\underbrace{E[X]}_{0} + 1$$

$$= \sigma^4(3) + 3(\sigma^2) + 1 \quad \Big| \; \sigma^2 = 2$$

$$= 3(4) + 6 + 1$$

$$= 12 + 6 + 1$$

$$= 19$$

$$\rightarrow \mathrm{Var}(\bar{Y}) = \frac{19 - 9}{K} = \frac{10}{K}$$

4.8)

Assuming that the cost function of evaluating $f(\theta; n)$ is const, since there are N terms in the sum, the complexity for both the function evaluation and its gradient, are $O(N)$.

To use Monte Carlo estimation, we can approximate the sum by sampling a subset of the terms, instead of summing over all N terms, you can take a random subset of size K and compute

$$\hat{F}(\theta) = \frac{N}{K} \sum_{k}^{K} w_k f(\theta; k) + \lambda R(\theta)$$

But how should we sample, the sampling is about to be uniform, meaning that: $\quad k \leftarrow \mathrm{Bin}(N, K)$

$w_s \sim \mathrm{Unif}(\{1, 2, \dots N\}) \qquad E[\hat{F}(\theta)] = E\left[\frac{N}{K} \sum_{s} w_s f(\theta; k) + \lambda R(\theta)\right]$

$\rightarrow s \sim \mathrm{Binomial}(N) \longrightarrow P(w_s \neq 0) = \frac{K}{N}$

$\rightarrow P(w_s = 0) = 1 - \frac{K}{N}$

$\xrightarrow{\forall s;} E[w_s] = \frac{K}{\cancel{K}N} w_k$

$\rightarrow E[\hat{F}(\theta)] = \frac{N}{K} \sum_{k}^{K} \frac{K}{N} w_k f(\theta; k) + \lambda R(\theta)$

$= \sum_{k}^{K} w_k f(\theta; k) + \lambda R(\theta) = F(\theta)$

$\longrightarrow E[\hat{F}(\theta)] = F(\theta) \longrightarrow$ So $O(N) \longrightarrow O(K)$

بنابراین امید $\hat{F}(\theta)$ ها $F(\theta)$ هست و میتوانیم جای محاسبه $F(\theta)$ با $O(N)$

به سراغ $\hat{F}(\theta)$ با $O(K)$ رفت.

با شکر، خسته نباشید :)