Ordered SGD: A New Stochastic Optimization Framework for Empirical Risk Minimization (AISTATS 2020)

Kenji Kawaguchi[1] Haihao Lu[2]

[1]MIT, [2]Google Research

May 4, 2022

arXiv Link

1 Introduction

2 Intuition and Algorithm

3 Optimization Theory

4 Experiments

5 Conclusion

Optimization Problem

- Consider the Optimization problem for minimizing the average of loss function with regularizer, let $L(\theta)$,

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L_i(\theta) + R(\theta)$$

where θ is the parameter vector of the parameterized model and ${L_i}'s$ are loss of the i-th data sample, and $R(\theta) \geq 0$ is regularizer.

- We can use Gradient based methods which iteratively update the parameters as follows

$$\theta_{k+1} = \theta_k - \alpha \nabla L(\theta_k)$$

Gradient Descent

- As we aleady know, the Gradient Descent method updates the parameter by using all the gradients of dataset

$$\theta_{k+1} = \theta_k - \alpha \nabla \frac{1}{n} \sum_{i=1}^{n} L_i(\theta_k)$$

Gradient Descent

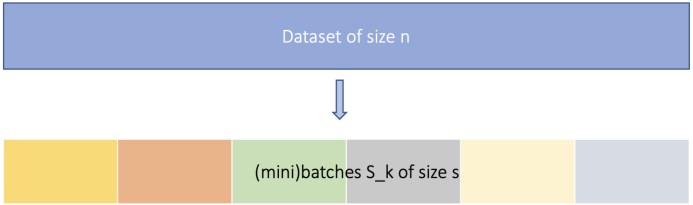
- As we aleady know, the Gradient Descent method updates the parameter by using all the gradients of dataset. i.e. full gradient

$$\theta_{k+1} = \theta_k - \alpha \nabla \frac{1}{n} \sum_{i=1}^{n} L_i(\theta)$$

- Computational expensive, inefficient

Stochastic Gradient Descent

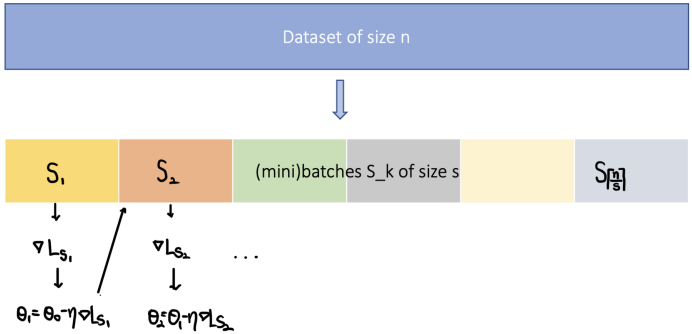
- To overcome the inefficiency, we devide the dataset to mini-batch and we replace the full gradient with mini-batch gradient(gradient estimator)



$$\theta_{k+1} = \theta_k - \alpha \nabla \frac{1}{s} \sum_{i=1}^{s} L_{s_{k+1}(i)}(\theta_k)$$

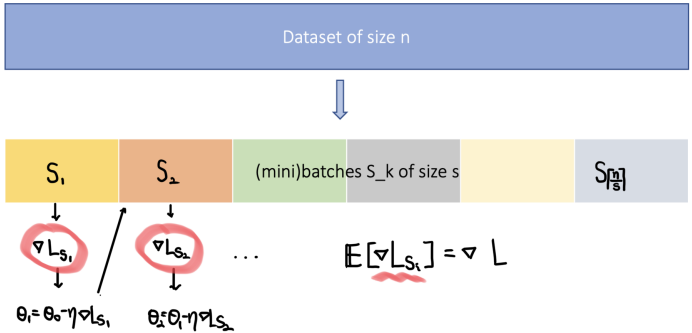
Stochastic Gradient Descent

- Is this a reasonable method?



Stochastic Gradient Descent

- Sure

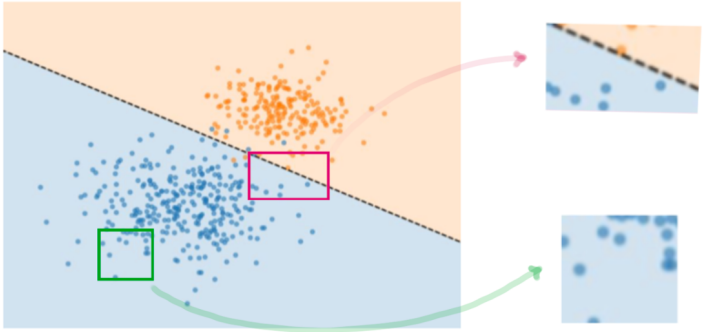


SGD is a unbiased method

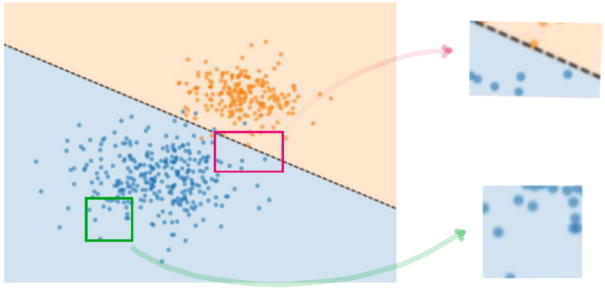
- Quite efficient and sometimes it is better than GD for use in DNN, which is non-convex.

Intuition

- When determining a decision boundary, there seem to be more impactful, helpful samples

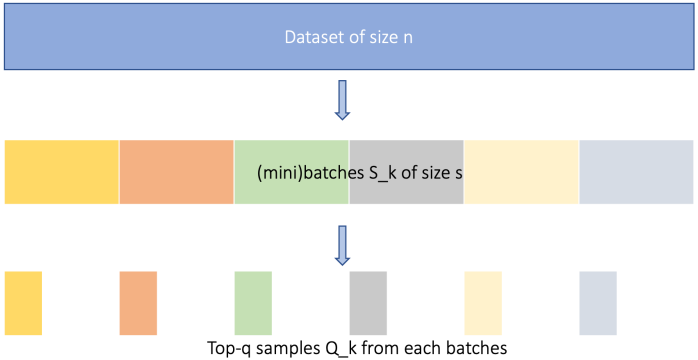


Intuition



- Is uniform sampling always better?
- If not,how can we get better?
- Loss values indicate how violate from the answer.

Intuition



- Let's use only top-q loss valued samples within a batch, instead of using all of a batch

Algorithm

Definition 1

Given a set of n real numbers $(a_1, a_2, \cdots, a_n)$, an index subset $S \subseteq \{1, 2, \cdots, n\}$, and a positive integer number $q \leq |S|$,
we define q-argmax$_{j \in S} a_j$ such that $Q \in$ q-argmax$_{j \in S} a_j$ is a set of q indices of the q-largest values of $(a_j)_{j \in S}$; i.e., q-argmax$_{j \in S} a_j =$ argmax$_{Q \subseteq S, |Q|=q} \sum_{i \in Q} a_i$

- For example, let $(a_1 = -5, a_2 = 10, a_3 = -4, a_4 = 6, a_5 = -1, a_6 = 5)$,
  if we want to know the 3 highest value indices of index subset $S = \{1, 2, 3, 4, 5\}$,
  find the $3 -$ argmaxset$_{j \in S} a_j = \{2, 4, 5\}$
  In our case, It returns the top-q largest loss valued data indices. Then we can use only these data samples.

Algorithm

Algorithm 1 Ordered Stochastic Gradient Descent(OSGD)

Input : problem data $L(x)$, step sizes sequence $(\alpha_k)_{k\in\mathbb{N}\cup\{0\}}$ and initialization θ_0

for $k = 0, 1, ...$ **do**

Sample a mini-batch uniformly: $S \subseteq \{1, 2, ..., n\}$ with $|S| = s$

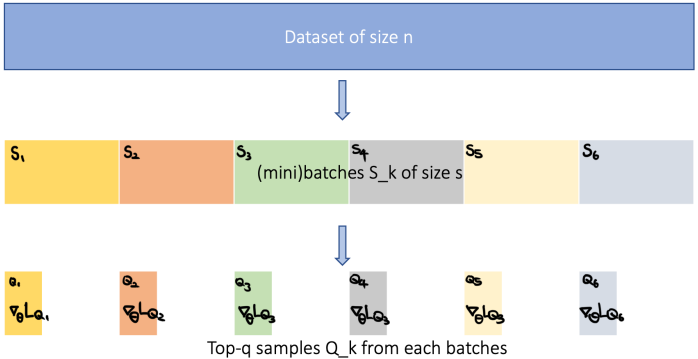
Find a set Q_k of top-q samples in S in term of loss values: $Q_k \in$ q-argmax$_{t\in S} L_t(\theta_k)$

Compute a gradient $\tilde{g}_k = \nabla_\theta L_{Q_k}(x_k)$ where $L_{\theta_k}(x_k) = \frac{1}{q}\sum_{t\in Q_k} L_t(\theta_k)$

Update parameter $\theta_{k+1} = \theta_k - \alpha_k \tilde{g}_k$

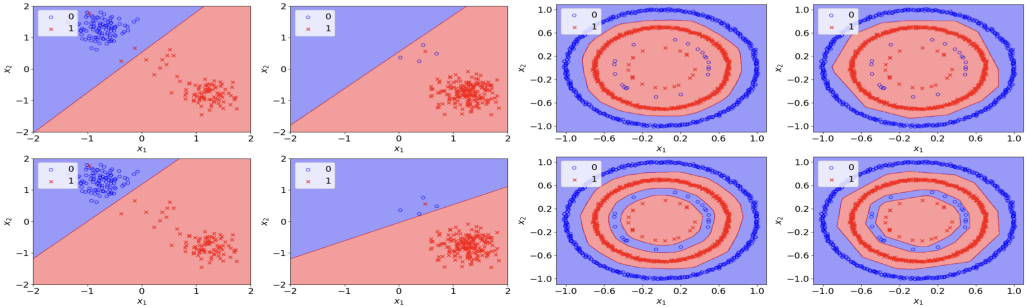
end for

Algorithm



- Notice that $\nabla_\theta L_{Q_k}$ is a biased gradient estimator

Toy examples



(a) with linear classifier (b) with linear classifier (c) with small ANN (d) with tiny ANN

Figure 1: Decision boundaries of mini-batch SGD predictors (**top** row) and ordered SGD predictors (**bottom** row) with 2D synthetic datasets for binary classification. In these examples, ordered SGD predictors correctly classify more data points than mini-batch SGD predictors, because a ordered SGD predictor can focus more on a smaller yet informative subset of data points, instead of focusing on the average loss dominated by a larger subset of data points.

Ordering notation

- Let's bring a notation similar to order statistics, but descending order.
  the notation of ordered indices : given a model parameter θ, let

$$L_{(1)}(\theta) \geq L_{(2)}(\theta) \geq \cdots \geq L_{(n)}(\theta)$$

  be the decreasing values of the individual losses $L_1(\theta), \cdots, L_n(\theta)$, where $(j) \in \{1, \cdots, n\}$. That is, $\{(1), \cdots, (n)\}$ as a permutation of $\{1, \cdots, n\}$ defines the order of sample indices by loss values.

- For example, let $L_1(\theta) = 1, L_2(\theta) = 10, L_3(\theta) = 2, L_4(\theta) = 6, L_5(\theta) = 3, L_6(\theta) = 5$, then we get the ordering notation by given θ,
  $L_{(6)}(\theta) = 1, L_{(1)}(\theta) = 10, L_{(5)}(\theta) = 2, L_{(2)}(\theta) = 6, L_{(4)}(\theta) = 3, L_{(3)}(\theta) = 5$

What we actually optimized is

Theorem 1

Consider the following objective function:

$$L_q(\theta) := \frac{1}{q}\sum_{t=1}^{n} \gamma_t L_{(t)}(\theta)$$

b where the parameter γ_t depends on the hyper parameter tuple (n, s, q), and is defined by

$$\gamma_t := \frac{\sum_{l=0}^{q-1} \binom{t-1}{l}\binom{n-t}{s-l-1}}{\binom{n}{s}}$$

Then, Ordered Stochastic Gradient Descent is a stochastic first-order method for minimizing $L_q(x)$ in sense that $\tilde{g}_k$ is used in OSGD is an unbiased estimator of a gradient of $L_q(x)$.

Example

- $L_q(\theta) := \frac{1}{q}\sum_{t=1}^{n}\gamma_t L_{(t)}(\theta)$ where $\gamma_t := \frac{\sum_{l=0}^{q-1}\binom{t-1}{l}\binom{n-t}{s-l-1}}{\binom{n}{s}}$

- For the case of (n=12, s=4, q=2),

$$\gamma_1 = 0.3333 \quad \gamma_2 = 0.3333 \quad \gamma_3 = 0.3152 \quad \gamma_4 = 0.2828 \quad \gamma_5 = 0.2404 \quad \gamma_6 = 0.1919$$
$$\gamma_7 = 0.1414 \quad \gamma_8 = 0.0929 \quad \gamma_9 = 0.0505 \quad \gamma_{10} = 0.0182 \quad \gamma_{11} = 0 \quad \gamma_{12} = 0$$

Take a close look

- $L_q(\theta) := \frac{1}{q}\sum_{t=1}^{n} \gamma_t L_{(t)}(\theta) = \sum_{t=1}^{n} \frac{\gamma_t}{q} L_{(t)}(\theta)$
- For the case of (n=12, s=4, q=2),

$$\frac{1}{q}\gamma_1 = 0.1667 \quad \frac{1}{q}\gamma_2 = 0.1667 \quad \frac{1}{q}\gamma_3 = 0.1576 \quad \frac{1}{q}\gamma_4 = 0.1414 \quad \frac{1}{q}\gamma_5 = 0.1202 \quad \frac{1}{q}\gamma_6 = 0.0808$$
$$\frac{1}{q}\gamma_7 = 0.0707 \quad \frac{1}{q}\gamma_8 = 0.0464 \quad \frac{1}{q}\gamma_9 = 0.0253 \quad \frac{1}{q}\gamma_{10} = 0.0091 \quad \frac{1}{q}\gamma_{11} = 0 \quad \frac{1}{q}\gamma_{12} = 0$$

$$\sum_{t=1}^{n} \frac{\gamma_t}{q} = 1$$

Asymptotic behavior of the γ_t

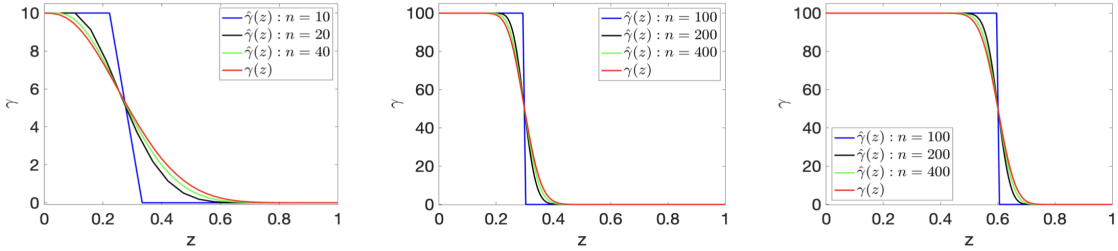
Proposition 1

Denote $z = \frac{j}{n}$ and $\gamma(z) := \sum_{l=0}^{q-1} z^l (1-z)^{s-l-1} \frac{s!}{l!(s-l-1)!}$. Then it holds that

$$\lim_{j,n\to\infty} \gamma_j = \frac{1}{n}\gamma(z)$$

Moreover, it holds that $1 - \frac{1}{s}\gamma(z)$ is the cumulative distribution function of $Beta(z; q, s-q)$.

Asymptotic behavior of the γ_t



(a) $(s, q) = (10, 3)$ (b) $(s, q) = (100, 30)$ (c) $(s, q) = (100, 60)$

Figure 2: $\hat{\gamma}(z)$ and $\gamma(z)$ for different (n, s, q) where $\hat{\gamma}$ is a rescaled version of γ_j: $\hat{\gamma}(j/n) = n\gamma_j$.

Convergence Analysis

Theorem 2

Let $\{\theta_t\}_{t=0}^{T}$ be a sequence generated by ordered SGD (Algorithm1). Suppose that $L(\cdot)$ is G_1-Lipschitz continuous for i = 1,$\cdots$,n, and $R(\cdot)$ is G_2-Lipschitz continuous. Suppose that there exists a finite $\theta_\star \in \text{argmin}_\theta L_q(\theta)$ and $L_q(\theta_\star)$ is finite. Then, the following two statements hold:

1. (Convex setting). If $L_i(\cdot)$ and $R(\cdot)$ are both convex, for any step-size η_t, it holds that

$$\min_{0\le t\le n} \mathbb{E}[L_q(\theta_t) - L_q(\theta_\star)] \le \frac{2(G_1^2 + G_2^2)\sum_{t=0}^{T}\eta_t^2 + ||\theta_\star - \theta_0||^2}{2\sum_{t=0}^{T}\eta_t}$$

Convergence Analysis

Theorem 2

Let $\{\theta_t\}_{t=0}^{T}$ be a sequence generated by ordered SGD [Algorithm1]. Suppose that $L(\cdot)$ is G_1-Lipschitz continuous for i $= 1, \cdots$,n, and $R(\cdot)$ is G_2-Lipschitz continuous. Suppose that there exists a finite $\theta_\star \in \text{argmin}_\theta L_q(\theta)$ and $L_q(\theta_\star)$ is finite. Then, the following two statements hold:

1. (Convex setting). If $L_i(\cdot)$ and $R(\cdot)$ are both convex, for any step-size η_t, it holds that

   if we choose $\eta_t \sim \mathcal{O}(\frac{1}{\sqrt{t}})$,

   the optimality gap $\min_{0 \le t \le n} (L_q(\theta_t) - L_q(\theta_\star))$ decay at the rate of $\tilde{\mathcal{O}}(\frac{1}{\sqrt{t}})$

Hyper-parameter setting

- Basically, OSGD uses the adaptive q-value setting which the default setting: q=s at the beginning of training, q=$\lfloor \frac{s}{2} \rfloor$ once train acc$\geq 80\%$, q=$\lfloor \frac{s}{4} \rfloor$ once train acc$\geq 90\%$, q=$\lfloor \frac{s}{8} \rfloor$ once train acc$\geq 95\%$, and q$\lfloor \frac{s}{16} \rfloor$ once train acc$\geq 99.5\%$, where train acc represents training accuracy.

- This rule was derived based on the intuition that in the early stage of training, all samples are informative to build a rough model, while the samples around the boundary (with larger losses) are more helpful to build the final classifier in later stage.

Experimental results

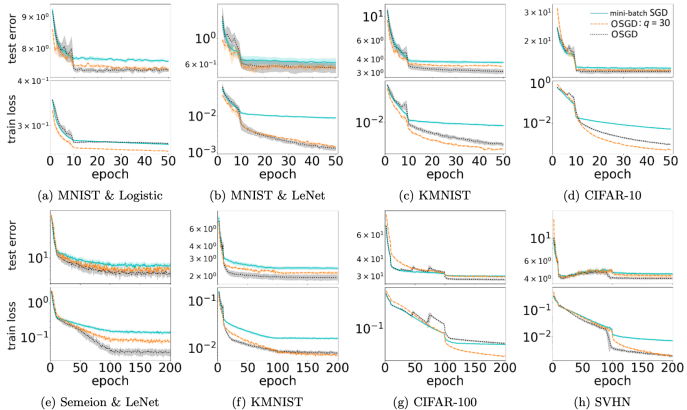
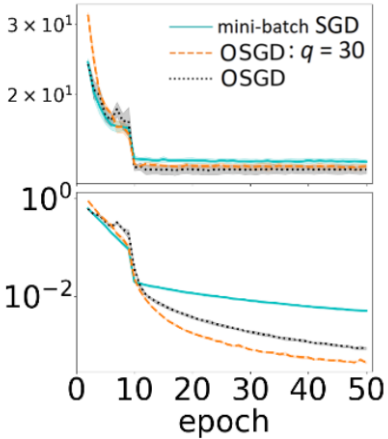


Figure 3: Test error and training loss (in log scales) versus the number of epoch. These are without data augmentation in subfigures (a)-(d), and with data augmentation in subfigures (e)-(h). The lines indicate the mean values over 10 random trials, and the shaded regions represent intervals of the sample standard deviations.

Take a close look



(d) CIFAR-10

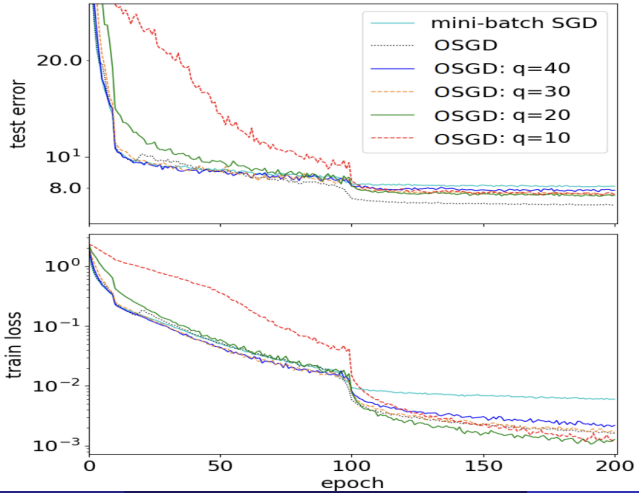
- Model = PreActResNet18
- (n,s,q) = (50000, 64, 64 or 30 or adaptive)
- the initial LR = 0.01 and decay with some policy
- 10 trial each and averaged

Wall-clock time

Table 4: Average wall-clock time (seconds) per epoch.

Data Aug	Datasets	Model	mini-batch SGD	ordered SGD	difference
No	Semeion	Logistic model	0.15 (0.01)	0.15 (0.01)	0.00
No	MNIST	Logistic model	7.16 (0.27)	7.32 (0.24)	-0.16
No	Semeion	SVM	0.17 (0.01)	0.17 (0.01)	0.00
No	MNIST	SVM	8.60 (0.31)	8.72 (0.29)	-0.12
No	Semeion	LeNet	0.18 (0.01)	0.18 (0.01)	0.00
No	MNIST	LeNet	9.00 (0.34)	9.12 (0.27)	-0.12
No	KMNIST	LeNet	9.23 (0.33)	9.04 (0.55)	0.19
No	Fashion-MNIST	LeNet	8.56 (0.48)	9.45 (0.31)	-0.90
No	CIFAR-10	PreActResNet18	45.55 (0.47)	43.72 (0.93)	1.82
No	CIFAR-100	PreActResNet18	46.83 (0.90)	43.95 (1.03)	2.89
No	SVHN	PreActResNet18	71.95 (1.40)	66.94 (1.67)	5.01
Yes	Semeion	LeNet	0.28 (0.02)	0.28 (0.02)	0.00
Yes	MNIST	LeNet	14.44 (0.54)	14.77 (0.41)	-0.32
Yes	KMNIST	LeNet	12.17 (0.33)	11.42 (0.29)	0.75
Yes	Fashion-MNIST	LeNet	12.23 (0.40)	12.38 (0.37)	-0.14
Yes	CIFAR-10	PreActResNet18	48.18 (0.58)	46.40 (0.97)	1.78
Yes	CIFAR-100	PreActResNet18	47.37 (0.84)	44.74 (0.91)	2.63
Yes	SVHN	PreActResNet18	72.29 (1.23)	67.95 (1.54)	4.34

Varying q-size



Conclusion

- This purposely biased gradient estimator perform well not only empirical risk minimization but also perspective of generalization and computational efficiency
- This variant of SGD also guaranteed to converge

Thanks for your attention

Appendix

- proof of Theorem 1

  Need to find the function that the $\tilde{g}_k$ become an unbiased estimator for subgradient
  Taking expectation to $\tilde{g}_k$, it holds that

$$\mathbb{E}[\tilde{g}_k] = \frac{1}{q}\mathbb{E}[\sum_{i \in Q_k} g_i] = \frac{1}{q}\sum_{i=1}^{n} P(i \in Q_k) g_i = \frac{1}{q}\sum_{j=1}^{n} P((j) \in Q_k) g_{(j)}$$

  Define index set $A_j = \{(1), (2), ..., (j-1)\}$, denote that given order is measured from whole n sample losses at current parameter x_k, then

Appendix

$$
\begin{aligned}
P((j) \in Q) &= P((j) \in \text{q-argmax}_{t \in S} L_t(x_k)) \\
&= P((j) \in S \text{ and } S \text{ contains at most q-1 items in } A_j) \\
&= P((j) \in S) P(S \text{ contains at most q-1 items in } A_j | (j) \in S) \\
&= P((j) \in S) \sum_{l=0}^{q-1} P(S \text{ contains } l \text{ items in } A_j | (j) \in S)
\end{aligned}
$$

Then there are $\binom{n}{s}$ different sets S s.t $|S| = s$ and $\binom{n-1}{s-1}$ different sets S contains index (j). So $P((j) \in S) = \frac{\binom{n-1}{s-1}}{\binom{n}{s}}$.

Appendix

And given the condition $(j) \in S$, let S contains l items in A_j which implies $s-l-1$ items in $\{(j+1),(j+2),...,(n)\}$. Then it holds that

$$P(S \text{ contains } l \text{ items in } A_j | (j) \in S) = \frac{\binom{j-1}{l}\binom{n-j}{s-l-1}}{\binom{n-1}{s-1}}$$

Therefore

$$P((j) \in Q_k) = \frac{\binom{n-1}{s-1}}{\binom{n}{s}} \sum_{l=0}^{q-1} \frac{\binom{j-1}{l}\binom{n-j}{s-l-1}}{\binom{n-1}{s-1}} = \frac{\sum_{l=0}^{q-1}\binom{j-1}{l}\binom{n-j}{s-l-1}}{\binom{n}{s}} =: \gamma_j$$

So the expectation of ordered gradient is

Appendix

$$\mathbb{E}[\tilde{g}_k] = \frac{1}{q}\sum_{j=1}^{n} P((j) \in Q_k) g_{(j)} = \frac{1}{q}\sum_{j=1}^{n} \gamma_j g_{(j)}$$

which desired. Then the proof is done.