Concentration inequalities and central limit theorems with dependency graphs

Master thesis of
Sebastian Gerstner
under the supervision of
Prof. Dr. Ashkan Nikeghbali
University of Zurich

Finished September 2023

Abstract

Many classical results of probability theory assume that all random variables considered are independent. But it is often possible to generalise them to certain dependent cases. In this Master's thesis, we present the approach of dependency graphs. We show how dependency graphs and related notions have been used to prove generalisations of concentration inequalities and of the central limit theorem. We sometimes present these results in a slightly stronger formulation than found in the literature. We compare the upper bounds provided by the concentration inequalities to various concrete examples, which leads us to conjecture sharper inequalities. We round off by presenting sample applications of the results.

Acknowledgements

I would like to thank Maximilian Janisch for his many-sided help: He provided ideas and feedback, helped me with my mathematical questions, and all the while spread cheerfulness at the institute. I would also like to thank Nis Bohne for his help with LaTeX. The positive atmosphere at the institute also owes much to my fellow students who were working next to me; Noam Szyfer in particular was there almost whenever I was there.

I am also thankful that I did not live alone during this time, but had my flatmates, Becky, Nerea, Simon and Tabea. They listened to my joys and frustrations, and just made me feel at home. In the same vein, I am thankful to all the friends I have seen during this time.

I also owe a lot of thanks to my parents: they, too, supported me in many ways, listened to me and made me feel at home - for twenty-four years and counting. And they financed my studies, so my work would not have been possible without them.

All of this would only have been hard and frustrating without the faith that "life, the universe, and everything" is ultimately purposeful, and that this purpose does not hinge on anybody's achievements.

Notation 1 (Basic notations and definitions)**.** *Unless stated otherwise, we consider a general probability space* $(\Omega, \mathcal{F}, P)$*. We implicitly assume all random variables are defined on this same probability space.*
We write a.s. for "almost sure(ly)" and i.i.d for "independent and identically distributed".
For a real-valued random variable X*, we denote* $Q_X(u) = \inf\{t \in \mathbb{R} : P(X > t) \leq u\}$*.*
Apart from the usual probability distributions known from textbooks (Bernouilli, binomial, Poisson, uniform, normal / Gaussian, exponential, Cauchy), we define the Rademacher distribution as the uniform distribution on the set of two elements $\{-1, 1\}$*.*

Contents

1 Introduction **6**
1.1 A gentle start . . . 7

2 Some classical results assuming independence **10**
2.1 Concentration inequalities . . . 10
2.1.1 A simple example: Hoeffding's inequality . . . 10
2.1.2 Subgaussian and subexponential random variables . . . 11
2.1.3 The general Hoeffding inequality and Bernstein's inequality . . . 16
2.1.4 McDiarmid's inequality . . . 17
2.1.5 Concentration inequalities in statistical learning theory . . . 17
2.2 Central Limit Theorem . . . 19

3 Quantifying dependence **23**
3.1 Quantities based on joint moments . . . 23
3.1.1 Covariance and related notions . . . 23
3.1.2 Cumulants and joint cumulants . . . 24
3.2 Dependence coefficients . . . 31

4 Dependency graphs and related notions **40**
4.1 Covers . . . 40
4.2 Dependency graphs . . . 42
4.3 HD dependence . . . 45
4.4 Weighted dependency graphs . . . 46
4.5 Soft covers and thresholded dependency graphs . . . 47

5 Concentration inequalities with dependency graphs **49**
5.1 Concentration inequalities with proper covers . . . 49
5.2 Concentration inequalities with soft covers . . . 53
5.3 Assessing optimality . . . 55
5.3.1 Proper covers: some examples and their concentration . . . 55
5.3.2 Soft covers: comparing with simpler inequalities . . . 62
5.3.3 Examples with weak dependencies and their concentration . . . 63
5.4 Applications to statistical learning . . . 65
5.4.1 Bipartite ranking . . . 65
5.4.2 Data assumed to be m-dependent . . . 66

6 Generalised Central Limit Theorems **69**
6.1 Janson's Central Limit Theorem . . . 69

6.2 Extensions 71
6.3 Applications 72

7 Conclusion and outlook **74**

Bibliography **75**

A Python code and Jupyter notebooks **77**
A.1 Useful functions: depgraph.py 77
A.2 Testing subgaussian and subexponential norms: test.ipynb 82
A.3 Simulating concentration inequalities: depgraph.ipynb 86

List of Figures

5.1 Concentration of sums of graph-dependent random variables 61
5.2 Comparison with sums of weakly graph-dependent random variables . . . 64

Chapter 1

Introduction

In basic probability theory, we learn that beautiful things happen when random events are independent of each other. For example, if we toss a fair coin a large number of times and get a point for each "head", then our total number of points behaves approximately like a normal distribution. This is known as the *Central Limit Theorem*. Even without handwavy approximations, we know the probability of getting more than t points is smaller than a certain function that behaves like $\exp(-t^2)$. Results of this type are known as *concentration inequalities*.

That was assuming one coin toss does not influence the next one: all coin tosses are independent. In this case, this is a reasonable assumption. But there are many other instances of randomness where independence does not hold.

For example, this happens whenever things influence each other across time or space. Consider the weather. Is it raining in Zurich right now? We can certainly view this as random to some extend, because we cannot perfectly predict it. But just as certainly, if it is indeed raining here, the probability rises that it is also raining in the neighbouring town of Zollikon, or that it will still rain in ten minutes.
Or think about a family tree. Our age at the time of getting children is random to some extent. Even if your own age at the birth of your first child were independent of your mother's age when she gave birth to you: in absolute terms, the birth date of your child does depend on your mother's age at childbirth.

How can we cope with such cases of dependent random variables? Are there also concentration inequalities and central limit theorems?

Fortunately, the answer is yes. In fact, there has been quite a lot of research on this topic. One major axis of this research is trying to understand stochastic processes, such as Markov chains. Some researchers have thought about ways to quantify dependence [1]

[1] In English, the nouns "dependence" and "dependency" both exists, and apparently English speakers do not agree what the difference is, if any. For example, both "dependence graph" and "dependency graph" appear in the literature. For this thesis, we have chosen the following usage: "dependence" is a property of a set of random variables, namely the negation of independence; a "dependency" is the link between dependent random variables. For example, we will write "quantifying dependence", but "dependency graph", "number of dependencies". Most authors seem to conform to this usage, consciously

(how strongly random variables are dependent of each other). Still others have examined the model of dependency graphs.
We will focus on the last approach, dependency graphs.

The basic idea behind it is that in some situations random variables are only partly dependent. To reuse the weather example: perhaps the weather right now and the weather in ten minutes have a lot to do with each other, but this is less obvious if we compare the weather today and the weather a hundred years ago. In such situations, we can represent the random variables of interest in a graph, where an edge indicates dependence. (This is not the precise definition, but it does convey the fundamental intuition.)

Today, the theory of dependency graphs is mostly scattered across many research articles. In this work we will try to gather these dispersed bits into a unified presentation - or at least the most important bits, because completeness is hardly possible. However, we will not just repeat what others have said.
In three cases (theorems 5.1, 5.6 and 6.5), we will present results which seem to be essentially known, but which we have not found explicitly in the literature.
Additionally, we will devote a large section (section 5.3) to assessing the optimality of various concentration inequalities in the dependency graph setting. This will involve theoretical reflections as well as computer simulations. Together they will lead us to propose conjectures (conjectures 1 and 2) about new concentration inequalities; in some situations these inequalities would be stronger than those already known.

Dependency graphs are not a purely theoretical construct, but have found concrete applications. We will present two examples: Concentration inequalities with dependency graphs can be applied to the theory of statistical learning, for example in bipartite ranking or in learning from m-dependent data (a concept we will explain in due time). As for central limit theorems with dependency graphs, they can help us understand the approximate behaviour of large real-world systems, such as the growth of a large cell population. We will present these applications in section 5.4 and section 6.3, respectively.

The rest of this work is organised as follows: After a gentle start where we recall basic notions about independence, we will present classical results about independent random variables, namely concentration inequalities and the central limit theorem. This will be the content of chapter 2. Then, in chapter 3, we will present various ways of quantifying dependence; this will be useful later and is also interesting on its own. With the following chapter (chapter 4), we enter our main topic, and present in detail the concept of dependency graphs, and some closely related notions as well. In the following two chapters, we use these ideas to show concentration inequalities (chapter 5) and central limit theorems (chapter 6). In both of these chapters, the last section presents applications: to statistical learning in the first case, and to a cell population model in the second case. We will conclude in chapter 7 by presenting ideas for further research.

1.1 A gentle start

Let us briefly recall the definition of independence and comment on it:

---

or not.

Definition 1.1 (Independence)**.** *(see e.g. Jacod and Protter 2004, p. 15, 65.)*
Let I be an index set.

- *A collection of* ***events*** $(A_i)_{i\in I}$ *is* ***independent*** *if for every finite subset J of I,*

$$P\left(\bigcap_{i\in J} A_i\right) = \prod_{i\in J} P(A_i).$$

- *A collection of* ***sub-sigma-algebras*** $(\mathcal{A}_i)_{i\in I}$ *of $\mathcal{F}$ is* ***independent*** *if for every finite subset J of I and all events $A_i \in \mathcal{A}_i$, the events $(A_i)_{i\in J}$ are independent.*

- *A collection of* ***random variables*** $(X_i)_{i\in I}$ *is* ***independent*** *if the generated sigma-algebras $\sigma(X_i)$ are independent.*

It is important to note that *independence is not the same as pairwise independence.* To see this, consider the following example:

Example 1.2. *We toss a fair coin three times independently. Let us denote Y_i the outcome of the i-th toss (0 or 1). Consider the three events $A_1 = \{Y_2 = Y_3\}, A_2 = \{Y_1 = Y_3\}, A_3 = \{Y_1 = Y_2\}$. Then:*

- $P(A_1) = P(\{Y_2 = 0\} \cap \{Y_3 = 0\}) + P(\{Y_2 = 0\} \cap \{Y_3 = 0\}) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ *and analogously for A_2 and A_3;*

- *the events are pairwise independent: $P(A_1 \cap A_2) = P(Y_2 = Y_3 = Y_1) = \frac{1}{8} + \frac{1}{8} = \frac{1}{4} = P(A_1)P(A_2)$, and analogously for the other pairs;*

- *but they are not independent: $P(A_1 \cap A_2 \cap A_3) = P(Y_1 = Y_2 = Y_3) = \frac{1}{4} \neq \left(\frac{1}{2}\right)^3 = P(A_1)P(A_2)P(A_3)$.*

If we consider the indicator variables X_i corresponding to A_i, we get of course the same result.

However, a slightly stronger condition does imply independence, and we will use it several times:

Lemma 1.3. *(see e.g. Lampert et al. 2018, in the proof of theorem 1.)*
Let $X_1, ..., X_n$ be random variables. Assume that for each k, X_k is independent of $(X_{k+1}, ..., X_n)$. Then $X_1, ..., X_n$ are independent.

Proof. Let $A_1, ..., A_n$ be measurable sets. Then

$$\begin{aligned} P(X_1 \in A_1, ..., X_n \in A_n) &= P(X_1 \in A_1)P(X_2 \in A_2, ..., X_3 \in A_3) \\ &= ... = P(X_1 \in A_1)...P(X_2 \in A_2). \end{aligned}$$

□

Furthermore, there is a very specific setting that has all nice properties we could hope for: Gaussian vectors. Recall the definition:

Definition 1.4 (Gaussian vector)**.** *(see e.g. Jacod and Protter 2004, Definition 16.1.) A random variable with values in $\mathbb{R}^n$ is called a* ***Gaussian vector*** *(or* ***multivariate normal****) if any linear combination of its components follows a Gaussian distribution.*

Proposition 1.5. *(see e.g. ibid., Corollary 16.1 and Theorem 16.4.)*

- *Two components of a Gaussian vector are independent if and only if they are uncorrelated.*
- *The components of a Gaussian vector are independent if and only if the covariance matrix is diagonal (i.e. every pair of distinct components is uncorrelated).*

This proposition also implies that in the components of a Gaussian vector independence is equivalent to pairwise independence.

Chapter 2

Some classical results assuming independence

In this chapter, we will review some results assuming independence of random variables. These are the ones we would like to extend to dependent variables. In each proof, we will point to the steps that need independence.

2.1 Concentration inequalities

Concentration inequalities are statements (roughly) of the form:

$$\forall S_n \in \mathbb{S} : P(S_n > t) < \delta(n, t),$$

where $\mathbb{S}$ is a set of random variables. Typically, as we will see, $\mathbb{S}$ consists of sums of n independent random variables, with some additional properties. As we will see at the end of this section, such inequalities are important in the theory of statistical learning.

Two simple concentration inequalities appear already in basic probability textbooks, e.g. Jacod and Protter 2004, Corollaries 5.1 and 5.2. They are the Markov and Chebyshev inequalities.

Proposition 2.1 (Markov and Chebyshev inequalities)**.** *Let $t > 0$.*

- *Let X be an a.s. non-negative random variable. Then $P(X \geq t) \leq \frac{E[X]}{t}$.*
- *Let X be a random variable with mean μ and variance σ^2. Then $P(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2}$.*

Proof. See Vershynin 2018, p. 9. □

2.1.1 A simple example: Hoeffding's inequality

A simple concentration inequality is Hoeffding's inequality. The proof is representative of the proof of many concentration inequalities. We will therefore state it here and sketch

the proof, with a focus on the part where independence comes in.
Afterwards, we will state a more general version of Hoeffding's inequality, and Bernstein's inequality, both without proofs.

Theorem 2.2 (Hoeffding's inequality)**.** *(see e.g. Vershynin 2018, Theorem 2.2.2) Let $X_1, ..., X_N$ be independent Rademacher random variables. Let $a = (a_1, ..., a_N) \in \mathbb{R}^N$. Then, for any $t \geq 0$, we have*

$$P\left(\sum_{k=1}^{N} a_k X_k \geq t\right) \leq \exp\left(-\frac{t^2}{\|a\|_2^2}\right).$$

Sketch of proof. We use that for any $\lambda > 0$, the function $\exp(\lambda t)$ is increasing in t:

$$P\left(\sum_{k=1}^{N} a_k X_k \geq t\right) = P\left(\exp\left(\lambda \sum_{k=1}^{N} a_k X_k\right) \geq \exp(\lambda t)\right).$$

By the Markov inequality (proposition 2.1), this is smaller or equal than

$$\exp(-\lambda t) E\left[\exp\left(\lambda \sum_{k=1}^{N} a_k X_k\right)\right].$$

Now the following step is where independence comes in: We decompose the expectation into a product. We get that

$$\exp(-\lambda t) E\left[\exp\left(\lambda \sum_{k=1}^{N} a_k X_k\right)\right] = \exp(-\lambda t) \prod_{k=1}^{N} E[\exp(\lambda a_k X_k)] = \exp(-\lambda t) \prod_{k=1}^{N} \cosh(\lambda a_k).$$

We get the result using another upper bound and optimising with respect to λ. For details, see ibid. □

The method of this proof is known as the Cramér-Chernoff method.

For this and all the following concentration inequalities, the same bound holds for $-X_k$ instead of X_k. Therefore, we also get (in this example):

$$P\left(|\sum_{k=1}^{N} a_k X_k| \geq t\right) = P\left(\sum_{k=1}^{N} a_k X_k \geq t\right) + P\left(-\sum_{k=1}^{N} a_k X_k \geq t\right) \leq 2\exp\left(-\frac{t^2}{\|a\|_2^2}\right).$$

We will not mention this from now on, since the argument is always the same.

2.1.2 Subgaussian and subexponential random variables

For the following concentration inequalities, we need definitions of two relatively large classes of probability distributions: subgaussian and subexponential.

Definition 2.3 (Subgaussian, subexponential)**.** *(see e.g. ibid., Definitions 2.5.6, 2.7.5.) Let X be a real-valued random variable.*

- *X is called **subgaussian** if there is a constant $C > 0$ such that $E[\exp(X^2/C^2)] \leq 2$. The smallest such C is called the subgaussian norm of X and denoted $\|X\|_{\psi_2}$.*

- *X is called **subexponential** if there is a constant $C > 0$ such that $E[\exp(|X|/C)] \leq 2$. The smallest such C is called the subexponential norm of X and denoted $\|X\|_{\psi_1}$.*

For both classes there are actually many equivalent definitions:

Proposition 2.4 (Subgaussian properties)**.** *(see e.g. Vershynin 2018, Proposition 2.5.2 and its proof.)*
Let X be a real-valued random variable. Then the following are equivalent:

1. *X is subgaussian.*
2. *$\exists C_2 > 0 : \forall t \geq 0 : P(|X| \geq t) \leq 2\exp(-t^2/C_2^2)$.*
3. *$\exists C_3 > 0 : \forall p \geq 1 : \|X\|_p \leq C_3\sqrt{p}$.*
4. *$\exists C_4 > 0 : \forall \lambda \in [-\frac{1}{C_4}, \frac{1}{C_4}] : E[\exp(\lambda^2 X^2)] \leq \exp(C_4^2\lambda^2)$.*

Moreover, if $E[X] = 0$, then these properties are also equivalent to the following one:

5. *$\exists C_5 > 0 : \forall \lambda \in \mathbb{R} : E[\exp(\lambda X)] \leq \exp(C_5^2\lambda^2)$.*

Moreover, the constants $C_2, ..., C_5$ differ from $\|X\|_{\psi_2}$ only by a constant factor, and this factor is the same for all suitable X. More precisely, if X is subgaussian with norm $\|X\|_{\psi_2}$, then the above properties hold (at least) with $C_2 = \|X\|_{\psi_2}, C_3 = 3\|X\|_{\psi_2}, C_4 = 6\sqrt{e}\|X\|_{\psi_2}$ and (if the mean is zero) $C_5 = 6\sqrt{e}\|X\|_{\psi_2}$.
Finally, these properties are also equivalent to the following:

6. *$E[\exp(tX^2)]$ is finite for some $t > 0$.*
7. *The moment generating function $E[\exp(tX^2)]$ is finite in a real neighbourhood of zero.*

A very similar proposition holds also for subexponential random variables:

Proposition 2.5 (Subexponential properties)**.** *(see e.g. ibid., Proposition 2.7.1.)*
Let X be a real-valued random variable. Then the following are equivalent:

1. *X is subexponential.*
2. *$\exists C_2 > 0 : \forall t \geq 0 : P(|X| \geq t) \leq 2\exp(-t/C_2)$.*
3. *$\exists C_3 > 0 : \forall p \geq 1 : \|X\|_p \leq C_3 p$.*
4. *$\exists C_4 > 0 : \forall \lambda \in [0, \frac{1}{C_4}] : E[\exp(\lambda|X|)] \leq \exp(C_4\lambda)$.*

Moreover, if $E[X] = 0$, then these properties are also equivalent to the following one:

5. *$\exists C_5 > 0 : \forall \lambda \in [-\frac{1}{C_5}, \frac{1}{C_5}] : E[\exp(\lambda X)] \leq \exp(C_5^2\lambda^2)$.*

Moreover, the constants $C_2, ..., C_5$ differ from $\|X\|_{\psi_1}$ only by a constant factor, and this factor is the same for all suitable X. More precisely, if X is subexponential with norm $\|X\|_{\psi_1}$, then the above properties hold (at least) with $C_2 = \|X\|_{\psi_1}, C_3 = 3\|X\|_{\psi_1}, C_4 = 6e\|X\|_{\psi_1}$ and (if the mean is zero) $C_5 = 6e\|X\|_{\psi_2}$.
Finally, these properties are also equivalent to the following:

6. *$E[\exp(t|X|)]$ is finite for some $t > 0$.*

7. *The moment generating function $E[\exp(tX)]$ is finite in a real neighbourhood of zero.*

For both propositions, Vershynin 2018 (in propositions 2.5.2 and 2.7.1) proves the equivalence of properties 1-5, including the factors between the various constants. We will not repeat these proofs here. But we will show that the other subexponential properties are indeed equivalent to the first five ones. (Analogous arguments apply to subgaussian properties.) Before we start, we need the following lemma, which is also of independent interest:

Lemma 2.6. *1. $\|\cdot\|_{\psi_2}$ and $\|\cdot\|_{\psi_1}$ satisfy the triangle inequality.*

2. *If X is subgaussian (or subexponential), so is $aX + b$ for any constants a, b.*

3. *In the definition (property 1) and in property 2, we could replace the constant 2 by any other constant K strictly larger than 1, and the definition would still be equivalent.*

Proof. We prove the results for subexponential X; the proof for the subgaussian case is perfectly analogous. We assume in the proof that the equivalence of properties 1-5 is already known.

1. We will use property 3. Recall that C_3 differs from $\|X\|_{\psi_1}$ only by a constant factor c independent of X. Thus, up to this constant factor, $\|X\|_{\psi_1}$ is the smallest C_3 such that $\forall p \geq 1 : \|X\|_p \leq C_3 p$. But, if $\|X\|_p \leq c\|X\|_{\psi_1} p$ and $\|Y_p\| \leq c\|Y\|_{\psi_1} p$, then $\|X+Y\|_p \leq \|X\|_p + \|Y\|_p \leq cp(\|X\|_{\psi_1} + \|Y\|_{\psi_1})$, so $\|X+Y\|_{\psi_1} \leq (\|X\|_{\psi_1} + \|Y\|_{\psi_1})$.

2. We apply the previous point to $aX + b$. We have $\|aX + b\|_{\psi_1} \leq \|aX\|_{\psi_1} + \|b\|_{\psi_1} = |a|\|X\|_{\psi_1} + |b|/\ln(2) < \infty$, so $aX + b$ is also subexponential.

3. - In the definition:
   Consider two constants $1 < K_1 < K_2$. First suppose that for some $C > 0$, $E[\exp(|X|/C)] \leq K_1$. Then clearly the same quantity is also smaller than K_2. Inversely, suppose that for some $C > 0$, $E[\exp(|X|/C)] \leq K_2$. Then by dominated convergence

   $$\lim_{c\to\infty} E[\exp(|X|/c)] = E[\lim_{c\to\infty} \exp(|X|/c)] = 1.$$

   Therefore we can find a C' such that $E[\exp(|X|/C')] \leq K_1$.

- In property 2: Consider two constants $1 < K_1 < K_2$.
  First suppose that $\exists C_2 > 0 : \forall t \geq 0 : P(|X| \geq t) \leq K_1 \exp(-t/C_2)$. Then clearly the same quantity is also smaller than $K_2 \exp(-t/C_2)$.
  Inversely, suppose that $\exists C_2 > 0 : \forall t \geq 0 : P(|X| \geq t) \leq K_2 \exp(-t/C_2)$. We need to show that $\exists D > 0 : \forall t \geq 0 : P(|X| \geq t) \leq K_1 \exp(-t/D)$. We claim that this works for $D = C_2 \ln(K_2)/\ln(K_1)$. Indeed, now we can consider two cases:

  - If $0 \leq t \leq C_2 \ln(K_2)$, then

$$P(|X| \geq t) \leq 1 = K_1 \exp\left(-\frac{C_2 \ln(K_2) \ln(K_1)}{C_2 \ln(K_2)}\right) \leq K_1 \exp(-t/D).$$

  - For the case $t \geq C_2 \ln(K_2)$, we will show that

$$K_2 \exp(-t/C_2) \leq K_1 \exp(-t/D) = K_1 \exp\left(\frac{-t \ln(K_1)}{C_2 \ln(K_2)}\right).$$

    This is equivalent to

$$\ln(K_2) - t/C_2 \leq \ln(K_1) - \frac{t \ln(K_1)}{C_2 \ln(K_2)}.$$

    This is in turn equivalent to

$$\ln(K_2) - \ln(K_1) \leq \frac{t}{C_2}\left(1 - \frac{\ln(K_1)}{\ln(K_2)}\right).$$

    This inequality does hold for $t \geq C_2/\ln(K_2)$.

□

We can now proceed to prove properties 6 and 7 of the proposition.

Partial proof of proposition 2.5.

- Equivalence of subexponentiality with property 6:
  By point 3 of the previous lemma, X is subexponential if and only if $E[\exp(|X|/C)]$ is finite for some $C > 0$. We can set $t = 1/C$.

- Equivalence of subexponentiality with property 7:

  - 7 implies 6:
    Assume property 7. Then

$$E[\exp(t|X|)] = E[\exp(tX)\mathbf{1}_{X \geq 0}] + E[\exp(-tX)\mathbf{1}_{X<0}] \leq E[\exp(tX)] + E[\exp(-tX)],$$

    and we can take t close enough to zero such that both terms are finite.

  - Subexponentiality implies 7:
    If X is subexponential, then by point 2 of the previous lemma $X - E[X]$ is also subexponential and has mean zero. Therefore, by property 5, $\exists C_5 > 0 : \forall \lambda \in [-\frac{1}{C_5}, \frac{1}{C_5}] : E[\exp(\lambda(X - E[X]))] \leq \exp(C_5^2 \lambda^2)$. Then, for the same C_5 and λ's, $E[\exp(\lambda X)] \leq \exp(\lambda E[X]) \exp(C_5^2 \lambda^2) < \infty$.

□

Remark 2.7.

- *Property 2 in both propositions is a concentration inequality.*
- *Every subgaussian random variable is also subexponential. We can see this most easily using property 3 of both propositions. Indeed, for $p \geq 1$, we have $\sqrt{p} \leq p$; therefore, if $\forall p \geq 1 : \|X\|_p \leq C_3\sqrt{p}$, then also $\forall p \geq 1 : \|X\|_p \leq C_3 p$.*
- *X is subgaussian if and only if X^2 is subexponential. This follows directly from the definition.*

We also have the following lemma:

Lemma 2.8. *(see e.g. Vershynin 2018, Lemma 2.7.7.)*
If X, Y are subgaussian, then XY is subexponential and $\|XY\|_{\psi_1} \leq \|X\|_{\psi_2}\|Y\|_{\psi_2}$.

The following examples show that these classes contain many, though not all, common probability distributions:

Example 2.9. *(ibid., example 2.5.8, exercise 2.5.9, example 2.7.8.)*

1. *Gaussian random variables are subgaussian, and hence subexponential. For $X \sim \mathcal{N}(0,1)$, this follows from property 5 and the fact that $E[\exp(\lambda X)] = \exp(\lambda^2/2)$. For other normal distributions, the result follows by point 2 of lemma 2.6.*

2. *Any bounded random variable is subgaussian, and hence subexponential. Indeed, with*
$$C = \frac{\|X\|_\infty}{\sqrt{\ln(2)}},$$
we get
$$E[\exp(X^2/C^2)] = E\left[\exp\left(\ln(2)\frac{X^2}{\|X\|_\infty^2}\right)\right] \leq E[\exp(\ln(2))] = 2.$$
For property 5 (if the mean is zero), a better constant is possible, namely
$$C_5 = \frac{1}{2\sqrt{2}}(b-a)$$
where $[a,b]$ is the (a.s.) range of X (Hoeffding 1963).

3. *The exponential distribution (with parameter λ) is subexponential, but not subgaussian.*
First of all, let us show that it is subexponential. We use property 2 of proposition 2.5:
$$P(|X| \geq t) = P(X \geq t) = \int_t^\infty \lambda \exp(-\lambda x)\, dx = \exp(-\lambda t) \leq 2\exp(-\lambda t).$$
The analogous property of subgaussian distributions does not hold. Indeed, whatever our choice of C_2, we will always find a large t such that $2\exp(-t^2/C_2^2) < \exp(-\lambda t)$.

4. *The Poisson distribution (with parameter λ) is subexponential, but not subgaussian. First, let us show that it is subexponential. We use the definition (and write $t = 1/C$). Let X follow a Poisson distribution with parameter λ. Its moment generating function is*

$$E[\exp(tX)] = \exp(-\lambda)\sum_{k=0}^{\infty}\frac{\lambda^k}{k!}\exp(tk) = \exp(-\lambda)\exp(\lambda e^t) = \exp(\lambda(e^t-1)),$$

which is always finite.
However, X does not fulfill the definition of a subgaussian distribution. Indeed,

$$E[\exp(tX^2)] = \exp(-\lambda)\sum_{k=0}^{\infty}\frac{\lambda^k}{k!}\exp(tk^2),$$

and the convergence radius of this power series is only

$$\begin{aligned}\rho &= \lim_{k\to\infty}\frac{\exp(tk^2)(k+1)!}{\exp(t(k+1)^2)k!}\\ &= \lim_{k\to\infty}(k+1)\exp(t(k^2-(k+1)^2))\\ &= \lim_{k\to\infty}(k+1)\exp(-t(2k-1)) = 0,\end{aligned}$$

i.e. $E[\exp(tX^2)] = \infty$ everywhere except $t = 0$.

5. *The Cauchy distribution is not subexponential, and hence not subgaussian either. This follows from the fact that it has no finite expectation, so it cannot satisfy property 3 of proposition 2.5.*

2.1.3 The general Hoeffding inequality and Bernstein's inequality

Now that we have the concepts, we can state fairly general concentration inequalities for subgaussian and subexponential random variables. The first is the general Hoeffding inequality, for subgaussian random variables; the second is called Bernstein's inequality and applies to subexponential ones.

Theorem 2.10 (General Hoeffding inequality)**.** *(see e.g. Vershynin 2018, Theorem 2.6.3.)*
There exists a constant $c > 0$ such that, for any choice of

- *independent (not necessarily identically distributed) subgaussian random variables with mean zero, denoted $X_1, ..., X_N$,*
- *$a = (a_1, ..., a_N) \in \mathbb{R}^N$, and*
- *$t \geq 0$,*

we have the following concentration inequality:

$$P\left(\sum_{k=1}^{N} a_k X_k \geq t\right) \leq \exp\left(-\frac{ct^2}{\|a\|_2^2 \max_k \|X_k\|_{\psi_2}}\right).$$

In particular, if the X_k take values a.s. in an interval of length c_k, we have:

$$P\left(\sum_{k=1}^{N} X_k \geq t\right) \leq \exp\left(-\frac{2t^2}{\sum_k c_k}\right).$$

The corresponding inequality for subexponential random variables is the following:

Theorem 2.11 (Bernstein's inequality)**.** *(see e.g. Vershynin 2018, Theorem 2.8.2.) There exists a constant $c > 0$ such that, for any choice of*

- *independent (not necessarily identically distributed) subexponential random variables with mean zero, denoted $X_1, ..., X_N$,*
- *$a = (a_1, ..., a_N) \in \mathbb{R}^N$, and*
- *$t \geq 0$,*

we have the following concentration inequality:

$$P\left(\sum_{k=1}^{N} a_k X_k \geq t\right) \leq \exp\left(-c \min\left(\frac{t^2}{\max_k \|X_k\|_{\psi_1}^2 \|a\|_2^2}, \frac{t}{\max_k \|X_k\|_{\psi_1} \|a\|_\infty}\right)\right).$$

For both of these theorems, the proof uses the moment generating function of the sum, just as in theorem 2.2.

2.1.4 McDiarmid's inequality

Another important concentration inequality is McDiarmid's inequality, also known as the bounded differences inequality. We will also state it here. The proof again follows the same method as in the other inequalities, and we will not present it here.

Theorem 2.12 (McDiarmid's inequality)**.** *(see e.g. ibid., Theorem 2.9.2.) Let $X = (X_1, ..., X_N)$ be a vector of independent random variables taking values in a measure space $A = A_1 \times ... \times A_N$. Let $f : A \to \mathbb{R}$ be a measurable function with the following property:*

$$\begin{aligned} \forall k \in \{1, ..., N\} : \exists c_k > 0 : \forall x_1, ..., x_N \in A, x_k' \in A_k : \\ |f(x_1, ..., x_k, .., x_N) - f(x_1, ..., x_k', ..., x_N)| \leq c_k. \end{aligned}$$

Then, for any $t \geq 0$, we have

$$P(f(X) - E[f(X)] \geq t) \leq \exp\left(-\frac{2t^2}{\sum_k c_k^2}\right).$$

2.1.5 Concentration inequalities in statistical learning theory

As already mentioned at the beginning of this section, concentration inequalities play an important role in machine learning. Here we will present a simple example of how this works.

First, let us describe the general framework of (supervised) statistical learning, following (for example) Shalev-Shwartz and Ben-David 2014. We have random vectors (X_k, Y_k) that follow a certain unknown distribution on a set $\mathcal{X} \times \mathcal{Y}$. (In the following we will sometimes write (X, Y) if only its distribution matters.) We assume for now that these random vectors are all independent of each other. Now, we are given a sample S of N datapoints $(x_k, y_k) = (X_k(\omega), Y_k(\omega))$ (for $k = 1, ..., N$). A learning algorithm is an algorithm that takes S as an input and outputs a hypothesis $h_S : \mathcal{X} \to \mathcal{Y}$. Of course, we want $h_S^(X_k)$ to be equal (or at least close) to Y_k as often as possible, even for new datapoints.
To formalise this last condition, we use a so-called loss function $L : \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$, that describes how far away two elements of $\mathcal{Y}$ are from each other. The easiest example is binary classification, where $\mathcal{Y} = \{\pm 1\}$; here a frequent choice is the so-called zero-one loss: $L(y, y') = \mathbf{1}_{y \neq y'}$. In general, of course, L should be symmetric and satisfy $L(y, y) = 0$.
Given such a loss function, for any hypothesis h we can define its "true risk" $R(h) = E[L(Y, h(X))]$. Since we do not know the true distribution, we also do not know the true risk, so we have to estimate it. We define the empirical risk as

$$R_S(h) = \frac{1}{N} \sum_k L(y_k, h(x_k)).$$

Note that $E[R_S(h)] = R(h)$.

Very generally, concentration inequalities help us ensure that $R(h_S)$ only has a small probability of being too large: The model is unlikely to be "too wrong". This holds even though we do not know the exact probability distribution from which we picked the sample. Furthermore, we can use the inequalities the other way round: Given thresholds ϵ ("What is too wrong?") and δ ("What is too likely?"), concentration inequalities can help us tell what N should be so that $P(R(h_S) > \epsilon) < \delta$.

One simple example, where we can straightforwardly apply Hoeffding's inequality (theorem 2.10), is the following (ibid., section 11.2.1): Suppose we have already trained a binary classification model and found a hypothesis function h. Now we want to validate the model by comparing it to a newly sampled validation set, independent of the original sample. Call this validation set V and denote its size by N_V. Then Hoeffding's inequality ensures that

$$P\left(|R_V(h) - R(h)| > \epsilon\right) = P\left(\left|\sum_k L(Y_k, h(X_k)) - N_V R(h)\right| > N_V \epsilon\right) \leq 2 \exp\left(-2 N_V \epsilon^2\right).$$

(Here we have used the fact that L, and hence also R, takes its values in the interval $[0, 1]$.)
In other words, for any δ, with probability at least $1 - \delta$,

$$|R_V(h) - R(h)| \leq \sqrt{\frac{\ln(2) - \ln(\delta)}{2 N_V}}.$$

This is an bound on how well the validation error approximates the true error.
Yet in other words, assume we want to estimate the true error using a validation set, and assume we are content with an estimation error of at most ϵ with probability at least

$1-\delta$. Then the size of the validation set should be

$$N_V \geq \frac{\ln(2) - \ln(\delta)}{2\epsilon^2}.$$

Things become more complicated if we do not have access to a validation set, or if we want to prove properties of an algorithm *a priori*. In this case we have to give concentration bounds for $|R_S(h_S) - R(h_S)|$, where h_S is random as well. Here is a typical workaround (Shalev-Shwartz and Ben-David 2014, chapter 4): If we can show that *for all* possible hypotheses h, $|R_S(h) - R(h)|$ is likely to be small, then certainly $|R_S(h_S) - R(h_S)|$ is also likely to be small. If moreover the learning algorithm only outputs hypotheses from a finite class $\mathcal{H}$, then we can bound this quantity using the union bound and Hoeffding's inequality. This kind of bounds are called generalisation bounds.
Proofs become even more involved when the hypothesis class $\mathcal{H}$ is infinite (see e.g. ibid., chapter 28).

2.2 Central Limit Theorem

The central limit theorem for independent and identically distributed random variables is well-known and appears in all modern probability textbooks. These books usually prove it using the convergence of characteristic functions. Here we will present another approach, the *method of moments*, because this is what we will use in the dependent case. However, as we will see, it assumes that all moments are well-defined, so in the independent case it is less powerful than the standard method of characteristic functions.

Theorem 2.13 (Method of moments)**.** *Let X_n be a sequence of random variables having all moments. Assume that $\forall p \geq 1 : E[X_n^p] \to_{n\to\infty} m_p \in \mathbb{R}$. Then there exists a distribution P_X such that $E[X^p] = m_p = \lim_n E[X_n^p]$. If additionally this distribution is unique, then the X_n converge in distribution to X.*

Proof. (Inspired by Janson 1988, lemma 1.) We present the proof in several steps:

1. *Claim:* The sequence P_{X_n} is tight, i.e.

   $$\forall \epsilon > 0 : \exists K \text{compact} : \forall n : P(X_n \notin K) < \epsilon.$$

   Proof of claim 1: Since $E[X_n^2] \to_n a_2$, we also know that this sequence is bounded: $\forall n : E[X_n^2] \leq C_2$. For $\epsilon > 0$, we can define $K = [-\sqrt{C_2/\epsilon}, \sqrt{C_2/\epsilon}]$. Then, for all n,

   $$P(X_n \notin K) = P(X_n^2 > C_2/\epsilon) < \frac{E[X_n^2]}{C_2/\epsilon} \leq \epsilon.$$

2. *Claim:* There is a subsequence X_{n_k} that converges in distribution. Actually, *any* subsequence has a sub-subsequence that converges in distribution.
   Proof of claim 2: This follows from claim 1 by Prohorov's theorem.

3. *Claim:* If a subsequence X_{n_k} converges in distribution to some X, then $E[X^p] = m_p$.
   Proof of claim 3: We write X_n instead of X_{n_k} for simplicity. We need to show

$E[X_n^p] \to_n E[X^p]$.
We have

$$\begin{aligned} E[X_n^p] &= E[X_n^p \mathbf{1}_{X_n^p \geq 0}] - E[-X_n^p \mathbf{1} X_n^p < 0] \\ &= \int_0^\infty P(X_n^p > t, X_n^p \geq 0) dt - \int_0^\infty P(-X_n^p > t, -X_n^p > 0) dt \\ &= \int_0^\infty (P(X_n^p > t) - P(-X_n^p > t)) dt. \end{aligned}$$

We want to apply the dominated convergence theorem. This is possible because

$$\begin{aligned} |P(X_n^p > t) - P(-X_n^p > t)| &\leq \min\{2, 2P(X_n^{2p} > t^2)\} \\ &\leq \min\left\{2, \frac{2E[X_n^{2p}]}{t^2}\right\} \\ &\leq \min\left\{2, \frac{2C_{2p}}{t^2}\right\}, \end{aligned}$$

which is integrable on $[0, \infty[$. (We have used $C_{2p} := \sup_n E[X_n^{2p}] < \infty$; this holds because the sequence $E[X_n^{2p}]$ converges for $n \to \infty$.)
Therefore, by dominated convergence,

$$\begin{aligned} E[X_n^p] &\to_n \int_0^\infty \lim_n (P(X_n^p > t) - P(-X_n^p > t)) dt \\ &= \int_0^\infty (P(X^p > t) - P(-X^p > t)) dt = E[X^p]. \end{aligned}$$

4. *Claim:* If the distribution of X is uniquely determined by its moments $E[X^p] = m_p$, then the whole sequence X_n converges in distribution to X.
Proof of claim 4: By claims 2 and 3, there is a subsequence that converges to X in distribution. Assume that the sequence as a whole does not converge in distribution to X. Then there is a continuous bounded function f, an $\epsilon > 0$ and a subsequence $(X_{n_k})_k$ such that $\forall k : E[f(X_{n_k})] - E[f(X)] > \epsilon$. But by claims 2 and 3, there is a sub-subsequence $(X_{n_{k_j}})_j$ that converges in distribution to X. Therefore also there exists a k such that $E[f(X_{n_k}) - E[f(X)] < \epsilon$. This is a contradiction, so the sequence as a whole converges to X in distribution.

□

Remark 2.14. *The condition that a distribution P_X is determined by its moments is relatively weak. It is fulfilled for example if X is subexponential (see e.g. Klenke 2020, Korollar 15.33.)*[1]
An important example of a distribution not determined by its moments is the log-normal distribution (see e.g. ibid., Beispiel 15.5). It is defined as the distribution of $\exp(X)$ *where X is standard normal.*

The special case for the normal distribution is as follows:

[1] Klenke formulates this sufficient condition as $\limsup_{n\to\infty} \frac{1}{n} E[|X|^n]^{1/n} < \infty$. This is equivalent to saying that the sequence $\frac{1}{n}\|X\|_n$ is bounded, which is equivalent to property 3 of proposition 2.5.

Corollary 2.15 (Method of moments for normal convergence)**.** *Let X_n be a sequence of random variables having all moments. Assume that, for p odd, $E[X_n^p] \to_n 0$ and for p even, $E[X_n^p] \to_{n\to\infty} (p-1)\cdot(p-3)...\cdot 3\cdot 1$. Then the X_n converge in distribution to $\mathcal{N}(0,1)$.*

Proof. The moments defined above are those of the standard normal distribution. This distribution is subexponential, hence determined by its moments, so we conclude with theorem 2.13. □

We can now state the classical central limit theorem (though in a weaker form than usual):

Theorem 2.16 (Central Limit Theorem)**.** *Let X_n be a sequence of i.i.d. random variables all of whose moments are finite, or at least such that*

$$\forall p \exists N \forall n \geq N : E[|X_n|^p] < \infty.$$

Let $S_n = \sum_{k=1}^n X_n$. Then

$$\frac{S_n - nE[X_0]}{\sqrt{n\,\mathrm{Var}(X_0)}} \to_n \mathcal{N}(0,1)$$

in distribution.

Proof. (See e.g. Filmus 2010)
We use corollary 2.15. By considering $(X_n - E[X_n])/\sqrt{Var(X_n)}$, we can assume without loss of generality that the X_n have mean 0 and variance 1. Then $\frac{S_n - nE[X_n]}{\sqrt{n\,\mathrm{Var}(X_n)}} = S_n/\sqrt{n}$.
Consider a general exponent p. Then $E[(S_n/\sqrt{n})^p] = n^{-p/2}E[S_n^p]$. We can rewrite $E[S_n^p]$ as a complicated sum of terms of the type $C(\{p_1,...,p_t\})\sum_{\{k_1,...,k_t\}} E[X_{k_1}^{p_1}]...E[X_{k_t}^{p_t}] =: A(n,\{p_1,...,p_t\})$; there is such a term for each set of positive integers $\{p_1,...,p_t\}$ such that $\sum p_k = p$. (Here we have split the expectation of the product into a product of expectations, thanks to independence.)
The terms in the sum in $A(n,\{p_1,...,p_t\})$ are all the same, because the X_k are identically distributed. Therefore (writing X instead of X_k for simplicity),

$$\begin{aligned} A(n,\{p_1,...,p_t\}) &= \binom{n}{t} C(\{p_1,...,p_t\})E[X^{p_1}]...E[X^{p_t}] \\ &= n(n-1)...(n-t+1)\frac{C(\{p_1,...,p_t\})}{t!}E[X^{p_1}]...E[X^{p_t}]. \end{aligned}$$

Now recall that we care about the behaviour as $n \to \infty$ of $n^{-p/2}\sum_{\{p_1,...,p_t\}} A(n,\{p_1,...,p_t\})$.
On the one hand, the terms for which $t < p/2$ go to zero, so we can ignore them from now on.
On the other hand, as soon as one of the p_k is 1, the term will equal zero because $E[X] = 0$. This happens as soon as $t > p/2$. So we can ignore these terms as well.
Therefore, if p is odd, we get $E[(S_n/\sqrt{n})^p] \to_n 0$.
If p is even, $E[(S_n/\sqrt{n})^p]$ behaves as $n^{-p/2}n(n-1)...(n-p/2+1)\frac{C(\{2,...,2\})}{(p/2)!}E[X^2]...E[X^2]$, which goes to $\frac{C(\{2,...,2\})}{(p/2)!}$. We therefore need to compute $C(\{2,...,2\})$. It is the number

of possibilities for reordering the tuple $(X_1, X_1, X_2, X_2, ..., X_{p/2}, X_{p/2})$. If we ignore that some of these are equal, we get $p!$ reorderings; then we must divide it by $2^{p/2}$. Thus, for p even,

$$]E[(S_n/\sqrt{n})^p] \to_n \frac{p!}{(p/2)!2^{p/2}} = (p-1)\cdot(p-3)\cdot...3\cdot 1.$$

The last equality follows from the fact that $p! = ((p-1)\cdot(p-3)\cdot...3\cdot 1)\cdot(p(p-2)...2)$ and $p(p-2)...2 = 2(p/2)\cdot 2(p/2-1)\cdot...2\cdot 1 = (p/2)!2^{p/2}$.
We have found precisely the moments of the standard normal distribution, so we can conclude by corollary 2.15. □

To conclude this section, we should mention that there are many generalisations of the classical Central Limit Theorem. Concretely, we can relax either the independence or the identical distribution assumptions, or both. Here is one example (without proof) of a Central Limit Theorem that assumes independence, but not identical distribution:

Theorem 2.17 (Central Limit Theorem, non-i.d. case)**.** *(Jacod and Protter 2004, theorem 21.2.)*
Let X_n be a sequence of independent random variables. Instead of identical distribution, assume

$$\exists \epsilon > 0 : \sup_n E[|X_n|^{2+\epsilon}] < \infty,$$
$$\sum_{k=1}^{\infty} \mathrm{Var}(X_k) = \infty.$$

Let $S_n = \sum_{k=1}^n X_n$. Then

$$\frac{S_n - E[S_n]}{\sqrt{\sum_{k=1}^n \mathrm{Var}(X_k)}} \to_n \mathcal{N}(0,1)$$

in distribution.

In chapter 6, we will present a specific direction of relaxing the independence assumption, namely using dependency graphs.

Chapter 3

Quantifying dependence

We would like to generalise results about sets of independent random variables to similar sets of dependent random variables. Intuitively, we would expect that "weakly dependent" random variables will have a behaviour close to independent ones. But what does "weakly dependent" mean? How can we quantify dependence?

As we will see, there is unfortunately no one-size-fits-all solution. In the following, we will examine various approaches in turn.

Ideally, we would like a quantity that can be computed for any number of random variables, not just pairs of them. However, most of the approaches we will see in this chapter concern only pairwise dependence.

3.1 Quantities based on joint moments

3.1.1 Covariance and related notions

Probably the best-known way to quantify dependence is covariance or the closely related notion of correlation. However, this method is not always trustworthy: It works mainly under the assumption that everything is a Gaussian vector; from a mathematical point of view, such an assumption is rather restrictive.

Definition 3.1 (Covariance and correlation)**.** *(see e.g. Jacod and Protter 2004, p. 91.) Let X, Y be real-valued square-integrable random variables. Then their* ***covariance*** *is*

$$\operatorname{Cov}(X,Y) := E[XY] - E[X]E[Y]$$

and their ***correlation coefficient*** *is*

$$\operatorname{Corr}(X,Y) = \frac{\operatorname{Cov}(X,Y)}{\sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}}.$$

If $\operatorname{Cov}(X,Y) = 0$ *(or equivalently* $\operatorname{Corr}(X,Y) = 0$*), X and Y are called* ***uncorrelated****.*

Proposition 3.2. *(see e.g. ibid., p.91.)*
Independence implies zero covariance (or equivalently, zero correlation).

However, the converse implication does not hold: There are pairs of uncorrelated random variables (even normally distributed ones) that are not independent.

Example 3.3. [1] *Consider a standard normal random variable X and an independent Z uniformly distributed on $\{-1, +1\}$. Define $Y = ZX$. Then:*

- *Y is also standard normal: Its characteristic function is $E[exp(itY)] = E[exp(itX)\mathbf{1}_{Z=1} + exp(-itX)\mathbf{1}_{Z=-1}] = \frac{1}{2}E[exp(itX)] + \frac{1}{2}E[exp(-itX)] = E[exp(itX)]$ (where the last step uses that $-X$ is also standard normal and thus has the same characteristic function as X).*
- *X and Y are uncorrelated: $E[XY] - E[X]E[Y] = E[XY] = E[X^2Z] = E[X^2]E[Z] = 1 \cdot 0 = 0$.*
- *However, X and Y are dependent: For example, $P(\{X \in [0,1]\} \cap \{Y \in [2,3]\}) = 0 \neq P(X \in [0,1])P(Y \in [2,3])$.*

Thus covariance has two important drawbacks: First, it can only quantify pairwise dependence. Second, even for pairs, zero covariance is no trustworthy indicator of independence.

However, as we have seen in the introduction (proposition 1.5), there is a special setting where covariance is enough to quantify dependence: Gaussian vectors. This is not the same as a vector of Gaussian variables, as the previous example shows.

In the following, we will consider a generalisation of covariance, that also takes into account more complex dependence structures than pairwise dependence: joint cumulants.

3.1.2 Cumulants and joint cumulants

Cumulants are also known as semiinvariants. They are equivalent to moments, in the sense that the cumulants are a polynomial function the moments, and vice versa. So for example, if we want to apply the method of moments (theorem 2.13), we can use the cumulants instead of the moments; this is what we will do in chapter 6. The advantage of cumulants over moments is their relationship to joint cumulants: Joint cumulants quantify dependence in a certain sense (under a relatively weak condition). We will explain the details of these relationships in this subsection.

First, let us define the cumulants of a single random variable.

Definition 3.4 (Cumulants)**.** *(see e.g. Janson et al. 2000, p.145.)*
Let X be a random variable with finite moments. The k-th cumulant (or semiinvariant) of X is

$$\kappa_k(X) = i^{-k} \frac{d^k}{dt^k} \ln E[\exp(itX)](0).$$

Remark 3.5.
- *The finite moment assumption ensures that the derivatives exist. Indeed, since $E[\exp(i0X)] = 1 \neq 0$, the only possible obstacle to computing the derivative in $\kappa_k(X)$ is that $E[\exp(itX)]$ may not be k times differentiable. However,*

[1] This example is well-known, but I have not been able to find a source for it. It appears as an exercise in Jacod and Protter 2004, p.136.

if the k-th moment is finite, we have that $|\frac{d^k}{dt^k}\exp(itX)| = |i^k X^k \exp(itX)| = |X|^k$ *is integrable, and hence also* $E[\exp(itX)]$ *is* k *times differentiable.*

- *An alternative definition is the following:* $\ln E[\exp(itX)]$ *has the Taylor series* $\sum_{k=0}^{\infty} \kappa_k(X)\frac{(it)^k}{k!}$.
- *The first two cumulants are the expectation and the variance. Indeed, assuming finite moments, we can exchange differentiation and integration, and get:*

$$\frac{d}{dt}\ln E[\exp(itX)] = \frac{1}{E[\exp(itX)]}E[iX\exp(itX)],$$

whose value at $t = 0$ *is* $iE[X]$*; and*

$$\begin{aligned}\frac{d^2}{dt^2}\ln E[\exp(itX)] &= \frac{d}{dt}\frac{1}{E[\exp(itX)]}E[iX\exp(itX)] \\ &= \frac{1}{E[\exp(itX)]}E[i^2X^2\exp(itX)] - \frac{1}{E[\exp(itX)]^2}E[iX\exp(itX)]^2,\end{aligned}$$

whose value at $t = 0$ *is* $i^2E[X^2] - i^2E[X]^2 = i^2\operatorname{Var}(X)$.

As an example, let us compute the cumulants of a standard normal distribution.

Example 3.6. *Let* $X \sim \mathcal{N}(0,1)$. *Recall that* $E[\exp(itX)] = \exp(-t^2/2)$. *It follows that* $\kappa_1(X) = 0, \kappa_2(X) = 1$ *and all higher cumulants are zero.*

We can generalise this definition to the joint cumulant of several random variables:

Definition 3.7 (Joint cumulant)**.** *(see e.g. Janson et al. 2000, p.147.)*
Let $X_1, ..., X_k$ *be random variable with finite moments. (Some or all of them may be identical.) Their joint cumulant (or mixed cumulant, or mixed semiinvariant) is*

$$\kappa(X_1, ..., X_k) = i^{-k}\frac{d^k}{dt_1...dt_k}\ln E[\exp(i(t_1X_1 + ... + t_kX_k))](0).$$

To begin with, we have the following basic results:

Lemma 3.8. *(see e.g. ibid., Proposition 6.16.)*
Let $X, X_1, ..., X_k$ *be random variables with finite moments. Then:*

1. *The* k*-th cumulant of* X *is just the joint cumulant of* X *repeated* k *times.*
2. *The joint cumulant is symmetric (invariant under permutations of the variables).*
3. *The joint moment can be expressed in terms of joint cumulants:*

$$E[X_1...X_k] = \sum_{I_1,...,I_n}\prod_{r=1}^{n}\kappa(\{X_i|i \in I_r\}),$$

where the sum is over all partitions of $\{1, ..., k\}$.

4. *The joint cumulant can be expressed in terms of joint moments:*

$$\kappa(X_1, ..., X_k) = \sum_{I_1,...,I_n} (-1)^{n-1}(n-1)! \prod_{r=1}^{n} E\left[\prod_{j \in I_r} X_j\right], \tag{3.1.1}$$

where the sum is again over all partitions of $\{1, ..., k\}$*. (In particular, the joint cumulant of two variables is their covariance.)*

5. *The joint cumulant is multilinear.*

Proof. 1. We can write the joint cumulant of X repeated k times as

$$i^{-k} \frac{d^k}{dt_1 ... dt_k} \ln E[\exp(iX \sum_{j=1}^{k} t_j)](0).$$

If we denote $f(t) := \ln E[\exp(itX)]$, we can rewrite this as

$$i^{-k} \frac{d^k}{dt_1 ... dt_k} f(\sum_{j=1}^{k} t_j)(0).$$

We claim that this is equal to

$$i^{-k} \frac{d^k}{dt^k} f(t)(0).$$

This is just a simple analysis argument: Define $S(t_1, ..., t_k) = t_1 + ... + t_j$. Then by the chain rule,

$$\frac{d}{dt_1} f(\sum_{j=1}^{k} t_j) = f'(\sum_{j=1}^{k} t_j) \cdot 1 = f'(\sum_{j=1}^{k} t_j).$$

Repeating the same argument k times, we get

$$\frac{d^k}{dt_1 ... dt_k} f(\sum_{j=1}^{k} t_j) = f^{(k)}(\sum_{j=1}^{k} t_j).$$

Evalued at $t_1 = ... = t_k = 0$, this is clearly the same as

$$\frac{d^k}{dt^k} f(t)(0).$$

2. Since everything is continuous, the order of differentiation does not matter.

3. (Leonov and Shiryaev 1959) On the one hand, $E[\exp(i(t_1 X_1 + ... + t_k X_k))]$ has the k-th order Taylor expansion

$$\sum_{\alpha \in \mathbb{N}^k, |\alpha| \leq k} i^{|\alpha|} \frac{t_1^{\alpha_1} ... t_k^{\alpha_k}}{\alpha_1! ... \alpha_k!} E[X_1^{\alpha_1} ... X_k^{\alpha_k}] + o(|t|^k), \tag{3.1.2}$$

where $|\alpha| := \sum \alpha_j$. On the other hand,

$$E[\exp(i(t_1 X_1 + ... + t_k X_k))] = \exp \ln E[\exp(i(t_1 X_1 + ... + t_k X_k))];$$

again by Taylor expansion, this becomes

$$\exp\left(\sum_{\alpha\in\mathbb{N}^k,|\alpha|\leq k} i^{|\alpha|}\frac{t_1^{\alpha_1}...t_k^{\alpha_k}}{\alpha_1!...\alpha_k!}\kappa(X_1,...,X_1,X_2,...,X_k,...,X_k)\right)+o(|t|^k),$$

where inside κ, each X_j appears α_j times. By expanding the exponential, this becomes

$$1+\sum_{n=1}^{k}\frac{1}{n!}\left(\sum_{\alpha\in\mathbb{N}^k,|\alpha|\leq k} i^{|\alpha|}\frac{t_1^{\alpha_1}...t_k^{\alpha_k}}{\alpha_1!...\alpha_k!}\kappa(X_1,...,X_1,X_2,...,X_k,...,X_k)\right)^n+o(|t|^k),$$

which can be expanded into a (complicated) sum of the same type as eq. (3.1.2). We now want to compare the terms for $\alpha_1=...=\alpha_k=1$. On the one hand we have

$$i^k t_1...t_k E[X_1...X_k].$$

On the other hand, we have a sum over all possible sets of n integer vectors summing to $(1,...,1)$, or equivalently, over all partitions of $\{1,...,k\}$ into n subsets $I_1,...,I_n$:

$$i^k t_1...t_k\sum_{n=1}^{k}\frac{1}{n!}n!\sum_{I_1,...,I_n}\prod_{r=1}^{n}\kappa(\{X_i|i\in I_r\}).$$

(The factor $n!$ comes from the $n!$ different orderings of the set $\{I_1,...,I_n\}$, all of which appear in the sum.)
It follows that

$$E[X_1...X_k]=\sum_{I_1,...,I_n}\prod_{r=1}^{n}\kappa(\{X_i|i\in I_r\}),$$

where this time the sum is over all partitions of $\{1,...,k\}$ without regard to number of subsets.

4. (Leonov and Shiryaev 1959) We apply the same argument to $\ln E[\exp(i(t_1X_1+...+t_kX_k))]$. It has the k-th order Taylor expansion

$$\sum_{\alpha\in\mathbb{N}^k,|\alpha|\leq k} i^{|\alpha|}\frac{t_1^{\alpha_1}...t_k^{\alpha_k}}{\alpha_1!...\alpha_k!}\kappa(X_1,...,X_1,X_2,...,X_k,...,X_k)+o(|t|^k), \tag{3.1.3}$$

where inside κ, each X_j appears α_j times. On the other hand, we can write it as

$$\ln\left(1+\sum_{\alpha\in\mathbb{N}^k,|\alpha|\leq k} i^{|\alpha|}\frac{t_1^{\alpha_1}...t_k^{\alpha_k}}{\alpha_1!...\alpha_k!}E[X_1^{\alpha_1}...X_k^{\alpha_k}]+o(|t|^k)\right).$$

By expanding the logarithm ($\ln(1+x)=\sum_{n=1}^{k}\frac{(-1)^{n-1}}{n}x^n+o(x^k)$), this becomes

$$\sum_{n=1}^{k}\frac{(-1)^{n-1}}{n}\left(\sum_{\alpha\in\mathbb{N}^k,|\alpha|\leq k} i^{|\alpha|}\frac{t_1^{\alpha_1}...t_k^{\alpha_k}}{\alpha_1!...\alpha_k!}E[X_1^{\alpha_1}...X_k^{\alpha_k}]\right)^n+o(|t|^k).$$

Again, this can be expanded into a complicated sum of the same type as eq. (3.1.3). We compare the terms for $\alpha_1 = .. = \alpha_k = 1$. On the one hand we have

$$i^k t_1 ... t_k \kappa(X_1, ..., X_k).$$

On the other hand, we have a sum over all partitions of $\{1, ..., k\}$ into n subsets $I_1, ..., I_n$:

$$i^k t_1 ... t_n \sum_{n=1}^{k} \frac{(-1)^{n-1}}{n} n! \sum_{I_1,...,I_n} \prod_{r=1}^{n} E\left[\prod_{j \in I_r} X_j\right].$$

It follows that

$$\kappa(X_1, ..., X_k) = \sum_{I_1,...,I_n} (-1)^{n-1}(n-1)! \prod_{r=1}^{n} E\left[\prod_{j \in I_r} X_j\right].$$

5. This follows from the previous point.

□

This lemma enables us to compute the cumulants of a linear combination of random variables via their joint cumulants. (This is also the reason why Leonov and Shiryaev 1959 derived this result in the first place.) Concretely, the j-th cumulant of $X_1 + ... + X_n$ is the joint cumulant of this sum repeated j times. We can then split up this joint cumulant using multilinearity.

Another consequence is that in the method of moments (theorem 2.13) we can take cumulants instead of moments:

Corollary 3.9 (Method of cumulants)**.** *Let X_n be a sequence of random variables having all moments.*

1. *Assume that $\forall p \geq 1 : \kappa_p(X_n) \to_{n\to\infty} \kappa_p \in \mathbb{R}$. Then there exists a distribution P_X such that $\kappa_p(X) = \kappa_p = \lim_n \kappa_p(X_n)$. If additionally this distribution is unique, then the X_n converge in distribution to X.*

2. *In particular, if $\kappa_2(X_n) \to_n 1$ and all other cumulants converge to 0, then the X_n converge in distribution to $\mathcal{N}(0, 1)$.*

Proof. 1. By point 3 of lemma 3.8, the moments are a polynomial function of the cumulants: $m_j(X_n) = f_m(\kappa_1(X_n), ..., \kappa_j(X_n))$. Therefore the moments converge as well: $m_j(X_n) \to f_m(\kappa_1, ..., \kappa_j) =: m_j$. Therefore, by theorem 2.13, there exists a distribution P_X with these moments. But then by point 4 of lemma 3.8, P_X must have the corresponding cumulants as well, namely $\kappa_j(X) = f_\kappa(m_1, ..., m_j)$. But by the same result, $\kappa_j(X_n) = f_\kappa(m_1(X_n), ..., m_j(X))$. Therefore $\kappa_j(X_n) \to \kappa_j(X)$.
If the distribution P_X is uniquely determined by its cumulants, then it is also uniquely determined by its moments: if we know the moments, we know the cumulants, so we know the distribution. Therefore we can again apply theorem 2.13.

2. We have seen in example 3.6 that these are exactly the cumulants of the standard normal distribution. This distribution is subexponential, hence determined by its moments. Therefore it is also uniquely determined by its cumulants: if we know the cumulants, we know the moments, so we know the distribution. Hence we can apply the previous point.

□

Remark 3.10. *Janson 1988 has shown that convergence to the normal distribution holds even when, for some $m \geq 3$, $\kappa_j(X_n) \to_n 0$ for all $j \geq m$. This gives less to check. However (as Janson also shows), in all of these cases we will also have $\kappa_j(X_n)$ for $j \geq 3$.*

We will now state the results that explain the relationship of joint cumulants with independence. But first, we need the following lemma, from Petersen 1982. We will state it here without proof, in a version tailored to probability distributions. [2]

Lemma 3.11. *Let $X_1, ..., X_k$ be real-valued random variables with finite moments. Assume the distrubution of each X_i is determined by its moments (i.e., if a Y_i has the same moments as X_i, then it has the same distribution). Then the joint distribution of $(X_1, ..., X_k)$ is also determined by its joint moments.*

Proposition 3.12. *Let $X_1, ..., X_k$ be random variables with finite moments, and consider a partition $\{X_1, ...X_l\} \sqcap \{X_{l+1}, ..., X_k\}$. (We may rearrange the indices to simplify the notation.)*

1. *(Janson 1988, lemma 3) Assume that the two sets $\{X_1, ..., X_l\}, \{X_{l+1}, ..., X_k\}$ are independent of each other (i.e. the variables in the first set are jointly independent of those in the second set). Then $\kappa(X_1, ..., X_k) = 0$.*

2. *(Féray 2016, proposition 2.1.) Assume that each X_i is determined by its moments, and assume that $\kappa(X_{i_1}, ..., X_{i_n}) = 0$ whenever this joint cumulant involves variables from both sets of the partition (possibly with repetitions). Then the two sets $\{X_1, ..., X_l\}, \{X_{l+1}, ..., X_k\}$ are independent of each other.*

Proof. 1. By independence,

$$\begin{aligned} &\ln E[\exp(i(t_1X_1 + ... + t_kX_k))] \\ &= \ln(E[\exp(i(t_1X_1 + ... + t_lX_l))]E[\exp(i(t_{l+1}X_{l+1} + ... + t_kX_k))]) \\ &= \ln E[\exp(i(t_1X_1 + ... + t_lX_l))] + \ln E[\exp(i(t_{l+1}X_{l+1} + ... + t_kX_k))]. \end{aligned}$$

When we differentiate along t_1, the second term becomes zero. When we differentiate along t_k, the first term becomes zero as well. Hence $\kappa(X_1, ..., X_k) = 0$.

[2] In fact, Petersen 1982 considers Radon measures on $\mathbb{R}$ or $\mathbb{R}^k$. A Radon measure is a measure that is finite on compact sets and inner regular (i.e., $\mu(A) = \sup_{C \subset A \text{ compact}} \mu(C)$). On $\mathbb{R}^k$, every probability distribution is a Radon measure. That it is finite on compacts is clear. We can see the second part using monotone convergence: For any sequence of compact sets C_n increasing to A, by the monotone convergence theorem, $\mu(A) = E[\mathbf{1}_A] = \lim E[\mathbf{1}_{C_n}] = \lim \mu(C_n)$. (Such sequences exist because we are in the Borel sigma-algebra.)

2. Define $(Y_1, ..., Y_l) = (X_1, ..., X_l)$ and take $(Y_{l+1}, ..., Y_k)$ such that this random vector is independent of $(Y_1, ..., Y_l)$ and has the same joint distribution as $\{X_{l+1}, ..., X_k\}$. Then the joint cumulants of $(Y_1, ..., Y_k)$ are as follows:

   - Those joint cumulants that involve only variables from $\{Y_1, ..., Y_l\}$ or only from $\{Y_{l+1}, ..., Y_k\}$ are the same as the corresponding ones with the X_i's, since the joint distributions are the same.

   - The joint cumulants involving both sets of the partition are zero. This follows from the first point of the present proposition and from the fact that $(Y_{l+1}, ..., Y_k)$ is independent of $(Y_1, ..., Y_l)$. But the corresponding joint cumulants with the X_i's are also zero by assumption.

   Thus $(Y_1, ..., Y_k)$ has the same joint cumulants as $(X_1, ..., X_k)$, and hence also the same joint moments. By lemma 3.11, it follows that it has the same joint distribution. Therefore, since $(Y_{l+1}, ..., Y_k)$ is independent of $(Y_1, ..., Y_l)$, also $(X_{l+1}, ..., X_k)$ is independent of $(X_1, ..., X_l)$.

□

Example 3.13. *1. Consider the three Bernouilli random variables X_1, X_2, X_3 from example 1.2. Recall that they are pairwise independent, but not independent. They cannot be partitioned into independent sets of random variables. Indeed division into singletons does not work (otherwise the variables themselves would be independent), and whenever we take two of the variables together, the third one is dependent of them.*
Bernouilli random variables are determined by their moments, so by proposition 3.12 at least one of their joint cumulants is different from zero.
We can verify this using eq. (3.1.1)*:*

$$\begin{aligned}\kappa(X_1, X_2, X_3) &= E[X_1X_2X_3] - 3E[X_1]E[X_2X_3] + 2E[X_1]E[X_2]E[X_3] \\ &= \frac{1}{4} - 3\frac{1}{2}\frac{1}{4} + 2\left(\frac{1}{2}\right)^3 = \frac{1}{8} \neq 0.\end{aligned}$$

2. Consider the two Gaussian random variables X, Y from example 3.3, which are uncorrelated but dependent.
By remark 2.14, both X and Y are determined by their moments. Therefore, by proposition 3.12, at least one of their joint cumulants (involving both X and Y) must be different from zero.
Indeed it turns out that $\kappa(X, X, Y, Y) \neq 0$. We can see this via eq. (3.1.1)*. We can ignore all the terms where one of the factors is $E[X]$ or $E[Y]$, since these terms will be zero. It then follows that*

$$\begin{aligned}\kappa(X, X, Y, Y) &= E[XXYY] - E[XX]E[YY] - 2E[XY]E[XY] \\ &= E[Z^2]E[X^4] - E[X^2]E[Z^2]E[X^2] - 2E[Z]E[X^2]E[Z]E[X^2] \\ &= E[X^4] - E[X^2]^2 - 0 = 3 - 1 = 2 \neq 0.\end{aligned}$$

In this section, we have seen that joint cumulants are doubly useful: On the one hand, they quantify dependence in a certain sense (proposition 3.12); on the other hand, they

may help compute cumulants and hence prove convergence results via the method of cumulants (corollary 3.9).

3.2 Dependence coefficients

Another way to quantify dependence is by dependence coefficients, also (more often) called *mixing coefficients* in the context of stochastic processes. They are mostly denoted by various lowercase Greek letters: $\alpha, \beta, \rho, \phi$ and others. A great advantage is that they are a faithful criterion for independence: If the coefficient takes on a certain value (typically zero), we *know* the variables are independent. However, they are defined only pairwise and thus *a priori* they tell us only about the pairwise independence of the random variables. Moreover, in practice it may often be difficult to actually compute or estimate them.

Here we will focus on two of these dependence coefficients: the α coefficient, which will become important later (section 4.5); and the so-called maximal correlation coefficient, denoted ρ, which shows how after all one can use correlation to quantify dependence in a general setting. We base this section mostly on Bradley 2005; Dedecker and Prieur 2005, who give much more detailed surveys of dependence coefficients.

Definition 3.14. *(Bradley 2005) Let $\mathcal{A}, \mathcal{B} \subset \mathcal{F}$ be sigma-algebras. Then we define*

- $\alpha(\mathcal{A}, \mathcal{B}) := \sup_{A \in \mathcal{A}, B \in \mathcal{B}} |P(A \cap B) - P(A)P(B)|$,
- $\rho(\mathcal{A}, \mathcal{B}) := \sup_{f \in \mathcal{L}^2(\mathcal{A}), g \in \mathcal{L}^2(\mathcal{B})} |\operatorname{Corr}(f, g)|$. *We also call $\rho(\mathcal{A}, \mathcal{B})$ the* ***maximal correlation coefficient****.*

To extend the definition of α to random variables, we partly follow Dedecker and Prieur 2005:[3]

Definition 3.15. *Let X, Y be real-valued random variables, and $\mathcal{A} \subseteq \mathcal{F}$ a sigma-algebra. Then we define*

$$\alpha(X, Y) := \sup_{x,y \in \mathbb{R}} |P(X > x, Y > y) - P(X > x)P(Y > y)|$$

and

$$\alpha(X, \mathcal{A}) := \sup_{x \in \mathbb{R}, A \in \mathcal{A}} |P(X > x, Y \in A) - P(X > x)P(Y \in A)|.$$

Remark 3.16. *Bradley 2005 and many others, including Lampert et al. 2018; Tanoue 2021 whom we will cite later, write instead $\alpha(X, Y) := \alpha(\sigma(X), \sigma(Y))$ and analogously for other dependence coefficients. This definition is not equivalent, as we will see in example 3.20. To avoid confusion, inspired by Dedecker and Prieur 2005, we may write $\alpha'(X, Y) := \alpha(\sigma(X), \sigma(Y))$.*
Clearly, $\alpha(X, Y) \leq \alpha(X, \sigma(Y)) \leq \alpha(\sigma(X), \sigma(Y)) = \alpha'(X, Y)$.

The following two lemmas clarify the relationship between dependence coefficients and help us compute them:

[3] Dedecker and Prieur 2005 also put a factor 2 in front.

Lemma 3.17. $0 \leq 4\alpha(\mathcal{A}, \mathcal{B}) \leq \rho(\mathcal{A}, \mathcal{B}) \leq 1$.

Proof. The first inequality is clear from the definition of α. The last one follows directly from the fact that $\mathrm{Corr}(A, B) \in [-1, 1]$. We thus still need to show that $4\alpha(\mathcal{A}, \mathcal{B}) \leq \rho(\mathcal{A}, \mathcal{B})$.
Let X be a Bernouilli random variable with success probability p. Then its variance is $p - p^2$. This function has its maximum at $p = 1/2$, with value $1/4$. Therefore, for Bernouilli random variables, $\mathrm{Var}(X) \leq 1/4$. We therefore have:

$$\begin{aligned}
\rho(\mathcal{A}, \mathcal{B}) &= \sup_{f \in \mathcal{L}^2(\mathcal{A}), g \in \mathcal{L}^2(\mathcal{B})} |\operatorname{Corr}(f, g)| \\
&\geq \sup_{A \in \mathcal{A}, B \in \mathcal{B}} \left|\frac{P(A \cap B) - P(A)P(B)}{\sqrt{\operatorname{Var}(\mathbf{1}_A) \operatorname{Var}(\mathbf{1}_B)}}\right| \\
&\geq \sup_{A \in \mathcal{A}, B \in \mathcal{B}} 4|P(A \cap B) - P(A)P(B)| = 4\alpha(\mathcal{A}, \mathcal{B}).
\end{aligned}$$

□

Lemma 3.18. *Let A, B be events. Then $P(A^c \cap B) - P(A^c)P(B) = -(P(A \cap B) - P(A)P(B))$. In particular, $\alpha(A, B) = |P(A \cap B) - P(A)P(B)|$.*

Proof. The first formula follows from:

$$P(A^c \cap B) - P(A^c)P(B) = P(B) - P(A \cap B) - (1 - P(A))P(B) = -P(A \cap B) + P(A)P(B).$$

The second one follows by applying the previous one to all four combinations (A, B), (A^c, B), (A, B^c), (A^c, B^c), which represent all elements of $\sigma(A) \times \sigma(B)$. □

Now let us prove that the coefficients defined here do quantify dependence in an unambiguous way:

Proposition 3.19. *The following are equivalent:*

1. *$\mathcal{A}$ and $\mathcal{B}$ are independent.*
2. *$\rho(\mathcal{A}, \mathcal{B}) = 0$.*
3. *$\alpha(\mathcal{A}, \mathcal{B}) = 0$.*

If $\mathcal{A} = \sigma(X)$, the previous statements are also equivalent to:

$$\alpha(X, \mathcal{A}) = 0.$$

If additionally $\mathcal{B} = \sigma(Y)$, the previous statements are also equivalent to:

$$\alpha(X, Y) = 0.$$

Proof.
- (1) ⇒ (2):
  This follows directly from the definition of the maximal correlation coefficient, since independence implies zero correlation.

- (2) $\Rightarrow$ (3):
  This follows from the previous lemma.

- (3) $\Rightarrow$ (1):
  (3) implies that $\forall A \in \mathcal{A}, B \in \mathcal{B} : |P(A \cap B) - P(A)P(B)| = 0$. In other words, for all such sets, $P(A \cap B) = P(A)P(B)$. This is precisely the definition of independence of two sigma-algebras.

It remains to prove the equivalence of (1) and $\alpha(X, Y) = 0$. (The equivalence of (1) and $\alpha(X, \mathcal{A}) = 0$ follows by a similar argument.)
The quantities $P(X > x, Y > y)$ for $x, y \in \mathbb{R}$, taken together, fully determine the joint distribution of (X, Y). Therefore, X and Y are independent if and only if $\forall x, y \in \mathbb{R} : P(X > x, Y > y) = P(X > x)P(Y > y)$, which precisely means $\alpha(X, Y) = 0$. $\square$

Now we are ready to compute the dependence coefficients in an example:

Example 3.20. *Consider again the dependent Gaussian random variables X, Y from example 3.3. We should find that $\rho(X, Y)$, $\alpha(X, Y)$ and $\alpha'(X, Y)$ are all strictly larger than zero.*

- *To compute $\alpha'(X, Y)$, consider the real number $a > 0$ such that $P(X \in [-a, a]) = 1/2$. Then*

$$\begin{aligned}\alpha(X, Y) &\geq P(X \in [-a, a], Y \in [-a, a]) - P(X \in [-a, a])P(Y \in [-a, a]) \\ &= 1/2 - (1/2)^2 = 1/4.\end{aligned}$$

  By lemma 3.17, we therefore have $\alpha'(X, Y) = 1/4$.

- *By the same lemma, it follows that $\rho(X, Y) = 1$.*

- *Now let us compute $\alpha(X, Y)$. Define $p_x := P(X > x) = P(X < -x)$. Note that $Y = ZX$ and $X = ZY$, so all computations remain the same if we exchange X and Y. Consider two real numbers x, y, and assume w.l.o.g. that $x \leq y$ (equivalently, $p_y \leq p_x$). By lemma 3.18 and because the standard normal distribution is symmetric, it is enough to consider the case $0 \leq x \leq y$ (equivalently, $p_y \leq p_x \leq 1/2$):*

$$\begin{aligned}P(X > x, Y > y) - P(X > x)P(Y > y) &= P(X > y, Z = 1) - p_x p_y \\ &= \frac{1}{2} p_y - p_x p_y = p_y \left(\frac{1}{2} - p_x \right) \geq 0.\end{aligned}$$

  The maximum on the allowed domain is on the line $p_y = p_x =: p$, and therein at $p = 1/4$ with value $1/16$. We find therefore that $\alpha(X, Y) = 1/16$. This is an example where the inequality $\alpha(X, Y) \leq \alpha'(X, Y)$ is strict.

We can also construct a pair of events (or the corresponding indicator random variables) with any given α coefficient (with $\alpha \in [0, 1/4]$ of course). To do this, let $X \sim \mathcal{U}[0, 1]$ and define $A = \{X \leq 1/2\}, B = \{\alpha + 1/4 \leq X \leq \alpha + 3/4\}$. Then, by lemma 3.18,

$$\alpha(A, B) = |P(A \cap B) - P(A)P(B)| = |\frac{1}{2} - \left(\alpha + \frac{1}{4} \right) - \frac{1}{2}\frac{1}{2}| = |-\alpha| = \alpha.$$

As already mentioned, these coefficients tell us only about pairwise dependence. If we have a set of three or more random variables $X_1, ..., X_n$, we can consider quantities such as $\alpha(\{X_1, ..., X_k\}, \{X_{k+1}, ..., X_n\}) := \alpha(\sigma(X_1, ..., X_k), \sigma(X_{k+1}, ..., X_n))$. In this context, we can note the following lemma:

Lemma 3.21. *(see Lampert et al. 2018, in the proof of theorem 1.) Let $Y_1, ..., Y_k$ be real-valued random variables, and let $X_1, ..., X_n$ be real-valued random variables that are independent of $\{Y_1, ..., Y_k\}$. Then*

$$\alpha(Y_1, \{Y_2, ..., Y_k, X_1, ..., X_n\}) \leq \alpha(Y_1, \{Y_2, ..., Y_k\}).$$

The same holds for the α' coefficient.

Proof. We prove it for the α' coefficient. For the α coefficient, the proof is the same but considers less general sets.
Denote $N := n + k - 1$ (this is the number of elements of the set $\{Y_2, ..., Y_k, X_1, ..., X_n\}$). In the following, we define an N-dimensional "rectangle" B_j to be a cartesian product of N Borel sets $B_{j,m}$ (not necessarily intervals). We write such a cartesian product as $B_j = \bigotimes_{m=1}^{N} B_{j,m}$.
Any $B \in \mathcal{B}(\mathbb{R}^N)$ can be written as a countable union of disjoint rectangles, $B = \bigsqcup_{j=1}^{\infty} B_j = \bigsqcup_{j=1}^{\infty} \bigotimes_{m=1}^{N} B_{j,m}$. Then, for another Borel set A,

$$\begin{aligned}
&P(Y_1 \in A, (Y_2, ..., Y_k, X_1, ..., X_n) \in B) \\
&= \sum_{j=1}^{\infty} P(Y_1 \in A, (Y_2, ..., Y_k, X_1, ..., X_n) \in B_j) \\
&= \sum_{j=1}^{\infty} P\left(Y_1 \in A, (Y_2, ..., Y_k) \in \bigotimes_{m=1}^{k-1} B_{j,m}, (X_1, ..., X_n) \in \bigotimes_{m=k}^{N} B_{j,m}\right) \\
&= \sum_{j=1}^{\infty} P\left(Y_1 \in A, (Y_2, ..., Y_k) \in \bigotimes_{m=1}^{k-1} B_{j,m}\right) P\left((X_1, ..., X_n) \in \bigotimes_{m=k}^{N} B_{j,m}\right).
\end{aligned}$$

Similarly,

$$\begin{aligned}
&P(Y_1 \in A) P((Y_2, ..., Y_k, X_1, ..., X_n) \in B) \\
&= P(Y_1 \in A) \sum_{j=1}^{\infty} P\left((Y_2, ..., Y_k) \in \bigotimes_{m=1}^{k-1} B_{j,m}\right) P\left((X_1, ..., X_n) \in \bigotimes_{m=k}^{N} B_{j,m}\right).
\end{aligned}$$

Therefore,

$$
\begin{aligned}
&|P(Y_1 \in A, (Y_2, ..., Y_k, X_1, ..., X_n) \in B) - P(Y_1 \in A)P((Y_2, ..., Y_k, X_1, ..., X_n) \in B)| \\
&= |\sum_{j=1}^{\infty} P\left((X_1, ..., X_n) \in \bigotimes_{m=k}^{N} B_{j,m}\right) \\
&\quad \cdot \left(P\left(Y_1 \in A, (Y_2, ..., Y_k) \in \bigotimes_{m=1}^{k-1} B_{j,m}\right) - P(Y_1 \in A)P\left((Y_2, ..., Y_k) \in \bigotimes_{m=1}^{k-1} B_{j,m}\right)\right)| \\
&\leq \sum_{j=1}^{\infty} P\left((X_1, ..., X_n) \in \bigotimes_{m=k}^{N} B_{j,m}\right) \\
&\quad \cdot |P\left(Y_1 \in A, (Y_2, ..., Y_k) \in \bigotimes_{m=1}^{k-1} B_{j,m}\right) - P(Y_1 \in A)P\left((Y_2, ..., Y_k) \in \bigotimes_{m=1}^{k-1} B_{j,m}\right)| \\
&\leq \sum_{j=1}^{\infty} P\left((X_1, ..., X_n) \in \bigotimes_{m=k}^{N} B_{j,m}\right) \alpha'(Y_1, \{Y_2, ..., Y_k\}) \\
&= P\left((X_1, ..., X_n) \in \bigsqcup_{j=1}^{\infty} \bigotimes_{m=k}^{N} B_{j,m}\right) \alpha'(Y_1, \{Y_2, ..., Y_k\}) \\
&\leq \alpha'(Y_1, \{Y_2, ..., Y_k\}).
\end{aligned}
$$

Since this holds for any suitable sets A, B, it follows that

$$\alpha'(Y_1, \{Y_2, ..., Y_k, X_1, ..., X_n\}) \leq \alpha'(Y_1, \{Y_2, ..., Y_k\}).$$

□

For sets of three or more random variables, Lampert et al. 2018 have extended the α'-dependence coefficient to an *α'-separation coefficient* (though they write α for what we call α'). Analogous definitions are possible for other mixing coefficients, and what we will actually use here is the *α-separation coefficient* (so the notation does not change, but the content does slightly).

Definition 3.22. *The* ***α-separation coefficient*** *of a set of random variables $\{X_1, ..., X_n\}$ is*

$$\vec{\alpha}(X_1, ..., X_n) = \min_{\pi \in \Pi_n} \frac{1}{n-1} \sum_{i=1}^{n-1} \alpha(X_{\pi(i)}, \{X_{\pi(i+1)}, ..., X_{\pi(n)}\}).$$

Similarly, the ***α'-separation coefficient*** *is*

$$\vec{\alpha}'(X_1, ..., X_n) = \min_{\pi \in \Pi_n} \frac{1}{n-1} \sum_{i=1}^{n-1} \alpha'(X_{\pi(i)}, \{X_{\pi(i+1)}, ..., X_{\pi(n)}\}).$$

Remark 3.23. *ibid. actually use the factor $\frac{1}{n}$ instead of $\frac{1}{n-1}$. In our opinion, it is better to use $\frac{1}{n-1}$, since there are only $n-1$ terms in the sum. For example, this enables us to say that in the case $n = 2$, $\vec{\alpha}(X_1, X_2) = \alpha(X_1, X_2)$.*

Given an even n and an $\alpha \leq \frac{1}{4(n-1)}$, Lampert et al. 2018, theorem 4, provide an explicit example of a set of n random variables with α- (or α'-) separation coefficient α.

The following result shows that α-separation is a valid measure of dependence for a set of several random variables:

Proposition 3.24. *The following are equivalent:*

1. $\vec{\alpha}(X_1, ..., X_n) = 0$.
2. $\vec{\alpha}'(X_1, ..., X_n) = 0$.
3. $X_1, ..., X_n$ *are independent.*

Proof. That (3) implies (1) and (2) is clear from the definitions.
We will now show that (1) implies (3); the proof that (2) implies (3) is analogous. Assume that $\vec{\alpha}(X_1, ..., X_n) = 0$. Then there exists a permutation π such that

$$\sum_{i=1}^{n-1} \alpha(X_{\pi(i)}, \{X_{\pi(i+1)}, ..., X_{\pi(n)}\}) = 0.$$

Without loss of generality, assume that π is the identity. Since the α coefficient is non-negative, it means that, for all i, $\alpha(X_i, \{X_{i+1}, ..., X_n\}) = 0$; in other words, X_i is independent of $\{X_{i+1}, ..., X_n\}$. But then, by lemma 1.3, $X_1, ..., X_n$ are independent. □

The α (or α') coefficient has a beautiful application, known as *coupling*: in certain cases, if we have two potentially dependent random variables X and Y, we can construct a copy of Y that is independent of X and "usually" equal to Y; what this "usually" means is quantified by a dependence coefficient of X and Y. Notable such results go back to Berbee 1979 and Bradley 1983. Here we present, without proof, a result from Rio 2013.

Proposition 3.25. *(ibid., lemma 5.2.) Let X be a random variable with values in $[a, b]$, and let $\mathcal{A}$ be a sub-sigma-algebra. Assume that the probability space is rich enough to admit a $U \sim \mathcal{U}([0, 1])$ independent of X and $\mathcal{A}$. Then there exists a random variable X^* measurable with respect to $\sigma(X, U) \vee \mathcal{A}$ such that:*

1. *X^* is independent of $\mathcal{A}$,*[4]
2. *X^* has the same distribution as X,*
3. *$E[|X - X^*|] \leq 2(b - a)\alpha(X, \mathcal{A})$*[5]

There is also an extension of this result to the case where X is unbounded; it can be found in Peligrad 2002:

Proposition 3.26. *Let X be a real-valued integrable random variable, and let $\mathcal{A}$ be a sub-sigma-algebra. Assume that the probability space is rich enough to admit a $U \sim \mathcal{U}([0, 1])$*

[4] Rio 2013 writes "independent of X", but this seems to be a typo.
[5] The factor 2 comes from Rio's different definition of α.

independent of X and $\mathcal{A}$. Then there exists a random variable X^* measurable with respect to $\sigma(X, U) \vee \mathcal{A}$ such that:

1. *X^* is independent of $\mathcal{A}$,*
2. *X^* has the same distribution as X,*
3. *$E[|X - X^*|] \leq 4 \int_0^{\alpha(X,\mathcal{A})} Q_{|X|}(u)du$ (recall notation 1).*

Remark 3.27. *Peligrad 2002 adds further properties to this list, which are bounds in terms of other dependence coefficients.*

Tanoue 2021 has extended the last two propositions to sets of several random variables, either bounded or subgaussian. We will present his result for sets of bounded random variables, and a generalised version of his second result.

Proposition 3.28. *(ibid., lemma 3.1.) Let $\mathcal{X} = \{X_1, ..., X_n\}$ be a set of random variables such that each X_i takes its values in an interval $[a_i, b_i]$. Assume that the probability space is rich enough to admit n $\mathcal{U}[0, 1]$ random variables that are mutually independent and independent of $\mathcal{X}$. Then there exists a set of random variables $\mathcal{X}^* = \{X_1^*, ..., X_n^*\}$ such that:*

1. *The variables in $\mathcal{X}^*$ are mutually independent,*
2. *Each X_i^* has the same distribution as X_i,*
3. *$E[\sum_{i=1}^n |X_i - X_i^*|] \leq 2(n-1)\vec{\alpha}(\mathcal{X}) \max_i(b_i - a_i)$.*[6]

Proof. By reordering the X_i's, we can assume without loss of generality that $\vec{\alpha}(X_1, ..., X_n) = \frac{1}{n-1} \sum_{i=1}^{n-1} \alpha(X_i, \{X_{i+1}, ..., X_n\})$.
We apply proposition 3.25 n times. First, we consider X_1 and $\mathcal{X}_1 = \sigma(X_2, ..., X_n)$. Applying the proposition yields X_1^* distributed as X_1, independent of $\mathcal{X}_1$, with $E[|X_1 - X_1^*|] \leq 2(b-a)\alpha(X_1, \{X_2, ..., X_n\})$.
We proceed to construct $X_2^*, ..., X_n^*$. More explicitly, at step i, we consider X_i and $\mathcal{X}_i = \sigma(X_1^*, ..., X_{i-1}^*, X_{i+1}, ..., X_n)$, and we construct X_i^* with the same distribution as X_i. Additionally, X_i^* is independent of $\mathcal{X}_i$ and in particular it is independent of $(X_1^*, ..., X_{i-1}^*)$. Moreover,

$$E[|X_i - X_i^*|] \leq 2(b-a)\alpha(X_i, \mathcal{X}_i) \leq 2(b-a)\alpha(X_i, \{X_{i+1}, ..., X_n\}),$$

where the second inequality follows from lemma 3.21.
In the last step, we can simply choose $X_n^* = X_n$, since X_n already fulfills the desired properties.
After fulfilling all the n steps, we have a set $\mathcal{X}^* = \{X_1^*, ..., X_n^*\}$ such that each X_i^* has the same distribution as X_i. Moreover, the variables in this set are mutually independent. This follows from lemma 1.3 and the fact that each X_i^* is independent of the previous

[6] The original in Tanoue 2021 ignores the factor 2. This comes from confusing slightly different definitions of the α coefficient. The author has been notified. Less importantly, he uses what we call α'-separation, but the sharper α-separation works just as well.

ones.
Finally,

$$E[\sum_{i=1}^{n} |X_i - X_i^*|] \leq \sum_{i=1}^{n-1} 2(b_i - a_i)\alpha(X_i, \{X_{i+1}, ..., X_n\}) \\ \leq 2(n-1)\vec{\alpha}(\mathcal{X}) \max_i (b_i - a_i).$$

□

Tanoue 2021 also presents an analogous proposition for sets of subgaussian random variables. We will present a straightforward generalisation of his argument.

Proposition 3.29. *Let $\mathcal{X} = \{X_1, ..., X_n\}$ be a set of real-valued integrable random variables. Assume that the probability space is rich enough to admit n $\mathcal{U}[0,1]$ random variables that are mutually independent and independent of $\mathcal{X}$. Then there exists a set of random variables $\mathcal{X}^* = \{X_1^*, ..., X_n^*\}$ such that:*

1. *The variables in $\mathcal{X}^*$ are mutually independent,*
2. *Each X_i^* has the same distribution as X_i,*
3. *$E[\sum_{i=1}^n |X_i - X_i^*|] \leq 4 \sum_{i=1}^{n-1} \int_0^{\alpha(X_i, \{X_{i+1}, ..., X_n\})} Q_{|X_i|}(u)du$.*

In particular:

1. *Suppose one or several of the X_i's is subgaussian, with c_i such that $P(|X_i| > t) \leq 2\exp(-t^2/c_i^2)$ (for example $c_i = \|X_i\|_{\psi_2}$, see proposition 2.4). Then we can replace $Q_{|X_i|}(u)$ with $\sqrt{-c_i^2 \ln(u/2)}$. (This corresponds to ibid., lemma 4.1.)*
2. *Suppose one or several of the X_i's is subexponential, with c_i such that $P(|X_i| > t) \leq 2\exp(-t/c_i)$ (for example $c_i = \|X_i\|_{\psi_1}$, see proposition 2.5). Then we can replace $Q_{|X_i|}(u)$ with $-c_i \ln(u/2)$.*

Proof. We follow the previous proof, but this time, it is proposition 3.26 that we apply several times.
As before, define $\mathcal{X}_1 = \sigma(X_2, ..., X_n)$. We get X_1^* distributed as X_1 and independent of $\mathcal{X}_1$, and with $E[|X_1 - X_1^*|] \leq 4 \int_0^{\alpha(X_1, \{X_1, ..., X_n\})} Q_{|X_1|}(u)du$.
We construct $X_2^*, ..., X_n^*$ as follows: to construct X_i^*, we consider X_i and $\mathcal{X}_i$ and apply the proposition. In each case, we get

$$E[|X_i - X_i^*|] \leq 4 \int_0^{\alpha(X_i, \mathcal{X}_i)} Q_{|X_i|}(u)du \leq 4 \int_0^{\alpha(X_i, \{X_{i+1}, ..., X_n\})} Q_{|X_i|}(u)du;$$

again, the second inequality follows from lemma 3.21.
The X_i^* are all independent. As in the previous proof, this follows from lemma 1.3 and the fact that each X_i^* is independent of the previous ones. Again, $X_n^* = X_n$.
Finally,

$$E[\sum_{i=1}^{n} |X_i - X_i^*|] \leq 4 \sum_{i=1}^{n-1} \int_0^{\alpha(X_i, \{X_{i+1}, ..., X_n\})} Q_{|X_i|}(u)du.$$

□

Chapter 4

Dependency graphs and related notions

In chapter 3, our approach was to get a number or a sequence of numbers to quantify the "amount of dependence" between two random variables (or within a set of several random variables). In this chapter and the next one, we will examine a different approach: we split our set of random variables into several smaller sets, such that inside each of them the random variables are independent.

4.1 Covers

We can formalise this approach with the notion of *covers*.

Definition 4.1. *Let $\mathcal{X}$ by a set of random variables.*

- *A **cover** of $\mathcal{X}$ is a set of subsets $\mathcal{X}_j$ such that $\mathcal{X} = \bigcup_j \mathcal{X}_j$. It is called a **proper cover** if each $\mathcal{X}_j$ is independent (i.e. the random variables inside $\mathcal{X}_j$ are independent of each other).*

- *A **fractional cover** of $\mathcal{X}$ is a set of pairs $(\mathcal{X}_j, w_j)$ where $\mathcal{X}_j \subseteq \mathcal{X}, w_j \in [0,1]$, such that $\forall X \in \mathcal{X} : \sum_{j:X\in\mathcal{X}_j} w_j \geq 1$. (Thus a cover is a fractional cover where each subset has the same weight 1.) It is called a **proper fractional cover** if each $\mathcal{X}_j$ is a set of independent random variables.*

- *A fractional cover is called **exact** if the previous inequality is an equality, i.e. $\forall X \in \mathcal{X} : \sum_{j:X\in\mathcal{X}_j} w_j = 1$.*

- *$\chi(\mathcal{X})$ is the size of the smallest proper cover of $\mathcal{X}$ and is called its **covering number**.*

- *$\chi^*(\mathcal{X})$ is the smallest value of $\sum_j w_j$ over all proper fractional covers of $\mathcal{X}$, and is called its **fractional covering number**.*

Let us mention two simple results about (fractional) covers:

Lemma 4.2. *(Janson 2004, lemma 3.1) Let $\{(\mathcal{X}_j, w_j)\}_j$ be an exact fractional cover of*

a set of random variables $\mathcal{X}$. Then for any numbers a_X (for $X \in \mathcal{X}$),

$$\sum_{X \in \mathcal{X}} a_X = \sum_j w_j \sum_{X \in \mathcal{X}_j} a_X.$$

In particular (with $a_X = 1$), $|\mathcal{X}| = \sum_j w_j |\mathcal{X}_j|$.

Proof.

$$\sum_j w_j \sum_{X \in \mathcal{X}_j} a_X = \sum_j w_j \sum_{X \in \mathcal{X}} \mathbf{1}_{\mathcal{A}_j}(X) a_X = \sum_{X \in \mathcal{X}} a_X \sum_j w_j \mathbf{1}_{\mathcal{A}_j}(X) = \sum_{X \in \mathcal{X}} a_X.$$

□

We can often use this result to bound the fractional covering number from below. Concretely, suppose that we have a set of N random variables and know that a subset of independent random variables has at most n elements. Then for any proper fractional cover we have $N = \sum_j w_j |\mathcal{X}_j \leq n \sum_j w_j$. This implies $\sum_j w_j \geq N/n$. Since this holds for any proper fractional cover, it follows that $\chi^*(\mathcal{X}) \geq N/n$.

Here is another basic result about fractional covers:

Lemma 4.3. *(Janson 2004, lemma 3.2) Let $\mathcal{X}$ be a finite set of random variables. Then for every fractional cover $(\mathcal{X}_j, w_j)$ of $\mathcal{X}$, there is an exact fractional cover $(\mathcal{X}_{jk}, w_{jk})$, such that $\mathcal{X}_{jk} \subseteq \mathcal{X}_j$, and with $\sum_{jk} w_{jk} = \sum_j w_j$. (More precisely, we even have $\sum_k w_{jk} = w_j$.) In particular, if $(\mathcal{X}_j, w_j)$ is proper, then so is $(\mathcal{X}_{jk}, w_{jk})$. In particular, $\chi^*(\mathcal{X})$ is also the smallest value of $\sum_j w_j$ over all exact proper fractional covers.*

Proof. The idea is to refine the cover for each $X \in \mathcal{X}$, one by one, such that the desired property holds for X. More precisely, for each X, we want that $\sum_{jk: X \in \mathcal{X}_{jk}} w_{jk} = 1$ and that the corresponding quantity for the other elements is unchanged.
To do one such step, fix X. We have that $\sum_{j: X \in \mathcal{X}_j} w_j \geq 1$. To reach 1 instead, we create one or several "wastebins" for the excess weight, which do not contain X and hence do not figure in the sum. Concretely, we replace one or several $\mathcal{X}_j$ by $\mathcal{X}_{j1} = \mathcal{X}_j$ and $\mathcal{X}_{j2} = \mathcal{X}_j \setminus X$, with appropriate weights w_{j1}, w_{j2} such that $w_{j1} + w_{j2} = w_j$. (This also ensures that for $Y \neq X$, $\sum_{jk: Y \in \mathcal{X}_{jk}} w_{jk} = \sum_{j: Y \in \mathcal{X}_j} w_j$.)
We repeat this procedure for all $X \in \mathcal{X}$, and get the result. □

To end this section, we present the fractional covering number in a specific example.

Example 4.4. *(See e.g. ibid., remark 5.2) Consider the set of random variables $X = \{X_1, X_2, X_3\}$ from example 1.2. Recall that they are pairwise independent, but not independent.*
Several covers are possible for this set of random variables. Apart from the trivial cover (three singletons), we can also take two pairs (since each pair is independent). Covering X with only one set is clearly not possible, since they are not independent. Therefore, the covering number is 2.
However, we can create a fractional cover with weight sum 3/2: Each of the three pairs

forms an independent subset with weight 1/2. From lemma 4.2 it follows that the fractional covering number cannot be smaller than 3/2. Therefore the fractional covering number is actually equal to 3/2.

4.2 Dependency graphs

We can visualise the cover approach using so-called dependency graphs.[1] The idea is roughly that random variables are represented by vertices, and that we draw an edge between any two variables that are dependent. But, again, independence is not the same as pairwise independence, so the definition is more subtle than that.

The idea was introduced by Erdös and Lovász 1975, though without the name. It seems that the terminology was introduced by Petrovskaya and Leontovich 1983 ("graph-dependence") and Janson 1988 ("dependency graph").

Actually, several different definitions are possible and in use. We will speak of strong, weak and superweak dependency graphs. Weak dependency graphs correspond to Erdös and Lovász 1975's original idea, while strong dependency graphs are the notion used by Janson 1988; Petrovskaya and Leontovich 1983. What we call superweak dependency graphs comes from a remark in Janson 2004. When we just say "dependency graph", we mean the strong one, since this seems to be the most used definition.

Definition 4.5. *Let $V = \{X_1, ..., X_N\}$ be random variables. A graph $D = (V, E)$ (with vertex set V) is*

- *a **(strong) dependency graph** for $\{X_1, ..., X_N\}$ if the following condition is satisfied:*
  For any $V_1, V_2 \subset V$, if V_1 and V_2 are disjoint and not connected by any edge of G, then the set of random variables V_1 is independent of V_2.

- *It is a **weak dependency graph** if the above condition is satisfied for all singletons $V_1 \subset V$ (but not necessarily other V_1's). In this situation we also speak of **local dependence (LD)** (Paulin 2014).*

- *It is a **superweak dependency graph** if for any $U \subset V$ such that there is no edge inside U, the random variables in U are independent.*

For a set of N random variables, the complete graph with N vertices is always a dependency graph, though of course a rather uninteresting one. Indeed, there are no two subsets that are unconnected by edges of the graph, so there is nothing to check. Ideally, we are interested in a minimal dependency graph (which is also not unique, as we will see later).

In the following, we will usually denote N the number of nodes of a graph, and M the number of edges. We will not systematically mention this.

The connection of dependency graphs to covers is that all three types of dependency graphs yield proper covers: a set that does not contain any edges will consist of inde-

[1] This mathematical concept is not the same as the "dependency graphs" of computer science.

pendent random variables. We will elaborate on this later in this section. First, let us examine the relationship between the various types of dependency graphs:

Lemma 4.6. *Every strong dependency graph is also a weak dependency graph, and every (finite) weak dependency graph is also a superweak dependency graph.*

Proof. The first implication is clear.
For the second implication, assume that D is a weak dependency graph, and consider $U \subset V$ such that there is no edge inside U. Then we can take $v_1 \in U$ and $V_2 = U \setminus \{v_1\}$ and get that v_1 is independent of V_2. We repeat the same argument on V_2 and find that some v_2 is independent of $V_3 = V_2 \setminus \{v_2\}$. After $n = |U|$ steps, we have found $U = \{v_1, ..., v_n\}$ where each v_k is independent of the following ones. We conclude by lemma 1.3. □

We will now elaborate on the connection between dependency graphs (of any type) and covers. As already mentioned, a set that does not contain any edges consists of independent random variables. This leads us to the following definitions from graph theory, analogous to definition 4.1:

Definition 4.7. *(see e.g. Zhang and Amini 2022, section 2.1.1.)*
Let $G = (V, E)$ be a graph.

- *A **colouring** of G is a set of disjoint subsets $V_j \subseteq V$ such that $V = \bigcup_j V_j$ and there is no edge within any V_j. (Intuitively, each V_j corresponds to a colour, and we want that adjacent nodes do not get the same colour.)*
- *A **fractional colouring** of G is a set of pairs (V_j, w_j) where $V_j \subseteq V, w_j \in [0, 1]$, such that there is no edge within any V_j, and such that $\forall v \in V : \sum_{j:v \in V_j} w_j \geq 1$. (Thus a colouring is a fractional colouring where each subset has the same weight 1).*
- *The **colouring number** of G the size of the smallest colouring of G, and is denoted $\chi(G)$.*
- *The **fractional colouring number** of G the smallest value of $\sum_j w_j$ over all fractional colourings of G, and is denoted $\chi^*(G)$.*

We have the following lemma (see e.g. Bollobás 1990, p.91):

Lemma 4.8. *Let D be a graph and let Δ denote its maximum degree (i.e., the maximum number of edges involving any given vertex). Then $\chi(D) \leq \Delta + 1$.*

We can directly conclude the following:

Lemma 4.9. *Let V be a set of random variables and $D = (V, E)$ a dependency graph (of any type), with maximum degree Δ. Then a (fractional) colouring is also a proper (fractional) cover. Consequently, $\chi(V) \leq \chi(D) \leq \Delta + 1$ and $\chi^*(V) \leq \chi^*(D) \leq \Delta + 1$.*

The converse does not hold: A proper cover is not necessarily a colouring of a dependency

graph. We can see this in the following example:

Example 4.10. *We consider again the three random variables from example 1.2, with the complete graph as dependency graph. There are clearly non-trivial proper covers, for example $V_1 = \{X_1, X_3\}, V_2 = \{X_2\}$, but these are not colourings of the complete graph. The complete graph admits only the trivial colouring (a different colour for each edge).*

By now, it is time to ask: When do non-trivial dependency graphs arise at all? In the following, we will present two examples, and then we will examine a rather general framework that yields dependency graphs.

Example 4.11. *(See e.g. Janson 2004, remark 5.2, and Féray et al. 2016, example 9.1.4.)*
Yet again, consider the set of random variables $X = \{X_1, X_2, X_3\}$ from example 1.2, which are pairwise independent, but not independent.
In this setting with only three random variables, strong and weak dependency graphs are the same, because for any pair of disjoint subsets, at least one of them will be a singleton. These random variables admit several (strong) dependency graphs:

- *Of course, the complete graph with three vertices is a dependency graph for X.*
- *Furthermore, any set of two edges yields a dependency graph for X. For example, let us consider the graph with edge set $E = \{\{X_1, X_2\}, \{X_2, X_3\}\}$; the same argument applies to all other sets of two edges. Here only X_1 and X_3 are not directly connected, and indeed X_1 and X_3 are independent. Thus the definition is fulfilled here as well.*
- *A graph with less than two edges cannot be a dependency graph for X. Indeed, in such a graph, at least one vertex, say X_1, would not be connected to anything else. In a dependency graph this would mean that X_1 would be independent of (X_2, X_3), and this is not the case.*

However, any graph with one edge is a superweak dependency graph for X. Indeed, a $U \subset X$ such that there is no edge inside U can have at most two elements, and any pair of two elements of X is independent.
The empty graph is not even a superweak dependency graph for X. Indeed, the whole set X has no edge inside it, but it it is not independent.

This example shows that a set of random variables may admit several dependency graphs of any type.

Another example is the following:

Example 4.12. *(see Paulin 2014, section 7.2.)*
Let $X_1, ..., X_n, Q$ be independent Rademacher random variables, and define $X_{n+k} = QX_k$. We consider $X = \{X_1, ..., X_{2n}\}$.
One possible strong dependency graph connects each pair (X_k, X_{n+k}) and all of the X_{n+k}. Indeed, if two subsets are unconnected to each other, it means that they are based on disjoint subsets of $\{X_1, ..., X_n, Q\}$, so they are independent.
One possible weak dependency graph connects only each pair (X_k, X_{n+k}). Call this graph

W. To see that it is a weak dependency graph, take a single vertex, say X_1 and consider the set $X \setminus \{X_1, X_{n+1}\}$, which is the maximal set unconnected to X_1. Then this set is independent of X_1.
However, W is not a strong dependency graph. Indeed, the sets $\{X_1, X_{n+1}\}$ and $X \setminus \{X_1, X_{n+1}\}$ are not connected to each other, but they are dependent: Indeed, if we know X_1 and QX_1, we know Q, so not all outcomes are possible any more in $X \setminus \{X_1, X_{n+1}\}$.

These two examples together show that the different notions of dependency graphs are not equivalent.

Now we can turn to the general framework of Paulin 2014's HD dependence.

4.3 HD dependence

The idea of HD dependence is that our random variables are functions of some underlying ("hidden") independent random variables. Roughly speaking, when two "visible" random variables are functions of the same underlying "hidden" random variable, they are dependent of each other, and otherwise they are not. Not only do most known examples of dependency graphs follow this pattern, but HD dependence is also useful on its own to prove a generalisation of McDiarmid's inequality (ibid., theorem 7.3.4).

Definition 4.13. *(ibid., definition 7.3.1; see also Janson et al. 2000, example 1.5.)*
*Let $\{Y_\alpha\}_{\alpha \in \mathcal{A}}$ be independent, and let $\{X_k\}_{k \in \mathcal{B}}$ be such that each X_k is a function of a subset $\{Y_\alpha\}_{\alpha \in \mathcal{A}_k}$. Then we say that the X_k are **HD dependent**, or, more precisely, $HD(\max_k |\mathcal{A}_k|, \max_\alpha |\{k : \alpha \in \mathcal{A}_k\}|)$ dependent.*

Dependency graphs appear in a natural way in this situation:

Proposition 4.14. *(Paulin 2014, proposition 7.3.3.)*
Let $\{X_k\}_{k \in \mathcal{B}}$ be $HD(m, n)$ dependent as in the above definition. Then they admit a dependency graph with maximum degree at most $m(n-1)$.

Proof. Two disjoint subsets $\{X_k\}_{k \in V_1}, \{X_k\}_{k \in V_2}$ are independent of each other if they are based on disjoint subsets of the Y_α's, i.e. $\bigcup_{k \in V_1} \mathcal{A}_k \cap \bigcup_{k \in V_2} \mathcal{A}_k = \emptyset$. Therefore we can construct a dependency graph for the X_k's as follows: Let the vertices be the X_k's, and let there be an edge between X_k and X_j if and only if $\mathcal{A}_k$ and $\mathcal{A}_j$ have a common element. Now, consider a node X_k. It depends at most on m of the Y_α's, and each of these influences at most n other X_j's (including X_k itself). Therefore X_k has edges to at most $m(n-1)$ of the other X_j's. □

However, this is not the only situation where dependency graphs arise. In the examples of the previous section, we have seen dependency graphs that do not proceed from HD dependence (though the sets of random variables we considered are HD dependent in another way).

Here is a concrete example of HD dependence:

Example 4.15. *(See e.g. Janson et al. 2000, example 1.6.)*

Consider the Erdös-Rényi random graph model: a graph with n vertices, where each possible edge, independently, is present with probability p. Let Y_{jk} be the indicator of the event that there is an edge between nodes j and k. Then the Y_{kj}'s are all independent. Now consider various random variables $X_i = \prod_{(j,k)\in\mathcal{A}_i} Y_{jk}$, i.e. the indicators that certain sets of edges are present.
Then the X_i are $HD(\max_i |\mathcal{A}_i|, \max_\alpha |\{i : \alpha \in \mathcal{A}_k\}|)$ dependent.
We can also make these X_i to be vertices of a dependency graph, with an edge between X_i and X_j if $\mathcal{A}_i$ and $\mathcal{A}_j$ share an element.

HD dependence also enables us to construct a set of random variables with a given dependency graph. To this effect, let $D = (V, E)$ be a graph, with N vertices, $V = \{1, ..., N\}$, and M edges. Construct M independent random variables Y_e indexed by the edges $e \in E$ (the distribution does not matter). Additionally, for each vertex k we may (but do not need to) construct an additional independent random variable Y_k. Now, for each vertex k, let E_k be the set of edges adjacent to k. Define $X_k = f_k(\{Y_e : e \in E_k\} \cup \{Y_k\})$ for some arbitrary functions f_k defined on the proper domain. Then the situation of example 4.15 arises, so D is a dependency graph for $\{X_k\}_{k=1,...,N}$. This is what we will do later when assessing the optimality of concentration inequalities (section 5.3.

We will now proceed to two fairly recent generalisations of dependency graphs. In both cases the idea is that the graph should also include information about the strength of dependencies. We will thus be combining dependency graphs with the concepts from chapter 3.

4.4 Weighted dependency graphs

Weighted dependency graphs were introduced by Féray 2016. The idea is that each edge gets a weight corresponding to the strength of the dependency (as measured by joint cumulants). This is very specifically tailored towards proving a generalised central limit theorem with the method of cumulants.

Before giving a formal definition, we need to present a few concepts and notations, especially from graph theory: multisets, induced subgraphs, and spanning subtrees.

Multisets are intuitively "sets with repetitions" (but without ordering). We denote them with the same curly parentheses {} as usual sets, as long as the context makes it clear that we are talking about multisets. For example, as multisets, $\{a, a, b\}$ is different from $\{a, b\}$, but $\{a, b\}$ is still the same as $\{b, a\}$.
Formally, in terms of set theory, we could define multisets as equivalence classes of tuples modulo reordering. For example, $\{a, a, b\}$ would denote the equivalence class of the tuple (a, a, b); this equivalence class would also contain the tuples (a, b, a) and (b, a, a).

Definition 4.16 (Induced subgraph)**.** *Let D be a weighted graph, and let B be a multiset of vertices of D. Then the* ***subgraph induced by D on B*** *has one vertex for each element of B (with multiplicities); two of these vertices are connected if and only if the corresponding vertices of D are connected, and the weight of this edge is the weight taken from D. This new graph is denoted $D[B]$.*

Example 4.17. *Consider the complete graph with three vertices, denoted a, b, c; suppose*

the edge $\{a, b\}$ has weight $\frac{1}{2}$ and the other two edges have weight 1. Let $B = \{a, a, b\}$. Then the induced subgraph has vertices a_1, a_2, b and edges $\{a_1, b\}, \{a_2, b\}$, and both of these edges have weight $\frac{1}{2}$.

Definition 4.18 (Spanning tree)**.** *For a weighted graph $D = (V, E)$, a **spanning tree** is a subset of the edges, $E' \subseteq E$, such that (V, E') is a tree, i.e. is connected and contains no cycles. The weight of E' is the product of weights of E'. We denote $\mathcal{M}(D)$ the maximum weight of a subtree of D.*

Example 4.19. *Let D again be the complete weighted graph from example 4.17. In this case, the spanning trees of D are exactly the sets of two edges. If one of these edges is $\{a, b\}$, the weight of the spanning tree is $\frac{1}{2}$; otherwise it is 1. Therefore the maximum weight of a subtree is $\mathcal{M}(D) = 1$.*

We are now ready to define weighted dependency graphs. Essentially, they provide a bound on the joint cumulants of the random variables involved; hence they tell us not only where we can find independence, but also how strong the dependencies are. More precisely, the definition is as follows:

Definition 4.20 (Weighted dependency graph)**.** *Let $Y_\alpha, \alpha \in A$ be random variables. Let $C = (C_n)_{n\in\mathbb{N}}$ be a sequence of positive real numbers, and Ψ be a function on multisets of elements of A. Then a graph D with vertex set A is called a (C, Ψ) **weighted dependency graph** for $\{Y_\alpha, \alpha \in A\}$ if for any multiset B of elements of A*

$$|\kappa(Y_\alpha, \alpha \in B)| \leq C_r \Psi(B) \mathcal{M}(D[B]).$$

4.5 Soft covers and thresholded dependency graphs

Let us leave abstract mathematics for a moment, and think back to real-life situations we can model as dependent random variables. For example, suppose a disease has been spreading for some time among a large group of people who don't all know each other. We can represent these people as vertices in a graph, with edges representing a relationship such as "regularly meet each other". For a probabilistic model of the epidemic, we can assume that a person may either contract the disease (independently) at random, or get infected (with a certain probability, independently) by somebody they regularly meet. Consider the random variables X_i, representing the indicator of the event "person i is infected now." Clearly, if i and j regularly meet each other, X_i and X_j are (strongly) dependent. If they are once removed in the graph, X_i and X_j are still dependent, though presumably less strongly. We can continue arguing this way, so in the end, everything is dependent of everything (as long as the "regular meetings" graph has only one connected component).

In such a situation, a classical dependency graphs is not very informative, because it would just be the complete graph. According to Zhang and Amini 2022, practitioners often assume *m-dependence*, meaning that nodes are independent if they are more than m units away from each other. This yields a classical dependency graph and one can proceed from this assumption. But of course, this assumption is often only approximately correct.

To overcome this kind of imprecision at least in principle, Lampert et al. 2018 introduced

the idea of thresholded dependency graphs, together with the closely related notion of *soft covers*. The idea is that we only consider dependencies that are stronger than a certain threshold. The threshold is measured by α-coefficients (or more precisely α'-coefficients) in Lampert et al. 2018 and we will follow this standard, but analogous definitions and results are possible with other dependence coefficients. We will use a slightly different notation to enable the use of other dependence coefficients instead of α'. In the following we will fix a threshold γ in the range of possible values of our dependence coefficient.

Definition 4.21. *(ibid., definitions 5 and 6.) Let $\mathcal{X} = \{X_1, ..., X_N\}$ be random variables.*

- *A **soft (fractional) cover** of $\mathcal{X}$ with α-threshold γ is a (fractional) cover consisting of sets $\mathcal{X}_j$ such that each of them satisfies $\vec{\alpha}(\mathcal{X}_j) \leq \gamma$.*
- *$\chi_{\alpha \leq \gamma}(\mathcal{X})$ is the size of the smallest soft cover of $\mathcal{X}$ (with α-threshold γ).*
- *$\chi^*_{\alpha \leq \gamma}(\mathcal{X})$ is the size of the smallest proper fractional cover of $\mathcal{X}$ (with α-threshold γ).*
- *A graph $D = (\mathcal{X}, E)$ is a **thresholded dependency graph** with α-threshold γ for $\{X_1, ..., X_N\}$ if the following condition is satisfied:*
  For any $U \subset \mathcal{X}$ and any $X_i \notin U$, if there is no edge between X_i and any element of U, then $\alpha(X_i, U) \leq \gamma$.

We can replace α by any other dependence coefficient (but α is the one we will actually use).

Remark 4.22. *This definition extends the weak dependency graph from definition 4.5. Analogous definitions are possible for a "strong" or a "superweak" dependency graph. Using an α-threshold gives slightly weaker definitions than ibid.'s α'-threshold, but we will see that the same results still hold.*

A similar idea is presented in Isaev et al. 2021.

Chapter 5

Concentration inequalities with dependency graphs

Now we have presented the concepts of dependency graphs, we can finally prove results. This is what we will do in this chapter and the following one. In this chapter in particular, we will examine concentration inequalities with dependency graphs.

5.1 Concentration inequalities with proper covers

Various concentration inequalities involving dependency graphs (or, more precisely, proper covers) have been proven in the last few decades. Notably, Janson 2004 generalised Hoeffding's inequality (theorem 2.2) to sums of dependent bounded random variables. Here, we will further generalise his argument and prove inequalities for sums of dependent subgaussian or subexponential random variables, analogously to theorem 2.10 and theorem 2.11. For the first case, the subgaussian one, we have not found this generalisation in the literature, but it is rather straightforward and seems to be essentially known. The subexponential case has already been published in Tanoue 2022.

Theorem 5.1 (General Hoeffding inequality, dependent case)**.** *Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a set of subgaussian random variables with $E[\exp(\lambda(X_k - E[X_k]))] \leq \exp(\lambda^2 C_k^2)$ (for example $C_k = 6\sqrt{e}\|X_k - E[X_k]\|_{\psi_2}$, see proposition 2.4). Then, for any $t \geq 0$, we have*

$$P\left(\sum_{k=1}^{N} X_k - E\left[\sum_{k=1}^{N} X_k\right] \geq t\right) \leq \exp\left(-\frac{t^2}{4\chi^*(\mathcal{X})\sum_k C_k^2}\right).$$

Remark 5.2. *If X_k a.s. takes values in an interval $[a_k, b_k]$, we can take $C_k = \frac{1}{2\sqrt{2}}(b_k - a_k)$. This yields Janson 2004's original formulation of the theorem.*

Proof. We use the same variation of the Cramér-Chernoff method (from theorem 2.2) as in ibid. The crucial modification of the classical Cramér-Chernoff method is using Jensen's inequality to bound the moment generating function.
To simplify notation, we can assume without loss of generality that $E[X_k] = 0$ (otherwise

we define $X'_k = X_k - E[X_k]$ and prove the theorem for X'_k). By lemma 4.3, it is enough to consider *exact* proper fractional covers without affecting $\chi^*(\mathcal{X})$. Thus, let $\{(\mathcal{X}_j, w_j)\}_j$ be an exact proper fractional cover.
For each j, define $Y_j = \sum_{X \in \mathcal{X}_j} X$. By lemma 4.2 we have $\sum_{k=1}^N X_k = \sum_j w_j Y_j$.
Now we are arriving at the crucial step. Let p_j be positive numbers summing to 1 (we will later choose the optimal ones). Then, by Jensen's inequality,

$$\exp\left(\lambda \sum_{k=1}^N X_k\right) = \exp\left(\sum_j p_j \frac{\lambda w_j}{p_j} Y_j\right) \leq \sum_j p_j \exp\left(\frac{\lambda w_j}{p_j} Y_j\right).$$

The same holds for the expectation:

$$E\left[\exp\left(\lambda \sum_{k=1}^N X_k\right)\right] \leq \sum_j p_j E\left[\exp\left(\frac{\lambda w_j}{p_j} Y_j\right)\right].$$

Now, Y_j is a sum of independent random variables, so we can continue as usual: the last term is equal to

$$\sum_j p_j \prod_{X_k \in \mathcal{X}_j} E\left[\exp\left(\frac{\lambda w_j}{p_j} X_k\right)\right]$$

and by subgaussianity this is less or equal to

$$\sum_j p_j \prod_{X_k \in \mathcal{X}_j} \exp\left(\frac{\lambda^2 w_j^2}{p_j^2} C_k^2\right) = \sum_j p_j \exp\left(\frac{\lambda^2 w_j^2}{p_j^2} \sum_{X_k \in \mathcal{X}_j} C_k^2\right).$$

(We will denote $a_j = \lambda^2 w_j^2 \sum_{X_k \in \mathcal{X}_j} C_k^2$ for simplicity. Note that $a_j \geq 0$.)
Now, similarly to the classical Cramér-Chernoff method, we would like to find the optimal p_j's.

> *Claim:* The optimum is at $p_i = \sqrt{a_i} / \sum_j \sqrt{a_j}$, with value $(\sum_j p_j) \exp(a_j/p_j^2) = \exp((\sum_j \sqrt{a_j})^2)$.
> *Proof:* We need to minimise $\sum_j p_j \exp(a_j/p_j^2)$ under the constraints $\sum_j p_j = 1, p_j > 0$. If there are minima, they are stationary points of the Lagrangian
>
> $$f(\{p_j\}_j, t) := \sum_j p_j \exp(a_j/p_j^2) + t(\sum_j p_j - 1).$$
>
> The derivative according to p_j is
>
> $$f_j(\{p_j\}_j, t) = \exp(a_j/p_j^2) - 2\exp(a_j/p_j^2) a_j/p_j^2 + t = (1 - 2a_j/p_j^2)\exp(a_j/p_j^2) + t.$$
>
> For all these derivatives to be zero, we need $(1 - 2a_j/p_j^2)\exp(a_j/p_j^2)$ to give the same value for all j. Now, note that the derivative of $(1-2x)\exp(x)$ is $\exp(x)(1-2x-2) = -\exp(x)(1+2x)$, which is negative for $x \geq 0$. Therefore $(1-2x)\exp(x)$ for $x \geq 0$ is injective, and the only stationary point of the Lagrangian is where a_j/p_j^2 are equal. Because $\sum_j p_j = 1$, this means $p_i = \sqrt{a_i}/\sum_j \sqrt{a_j}$. If we plug this solution into our function $\sum_j p_j \exp(a_j/p_j^2)$, we get $(\sum_j p_j)\exp(a_j/p_j^2) = \exp((\sum_j \sqrt{a_j})^2)$.

Is this a minimum or a maximum? We could make the sum much larger by choosing ridiculously small p_j's (except the last one which would be close to 1); indeed, the exponential "kills everything". What we have found is therefore a minimum.

We therefore get

$$E\left[\exp\left(\lambda\sum_{k=1}^{N}X_k\right)\right] \leq \exp((\sum_j \sqrt{a_j})^2) = \exp\left(\lambda^2\left(\sum_j w_j\sqrt{\sum_{X_k\in\mathcal{X}_j}C_k^2}\right)^2\right).$$

We use this to bound the tail probability: for $\lambda \geq 0$,

$$P\left(\sum_{k=1}^{N}X_k \geq t\right) \leq \exp(-\lambda t)E\left[\exp\left(\lambda\sum_{k=1}^{N}X_k\right)\right] \leq \exp\left(\lambda^2\left(\sum_j w_j\sqrt{\sum_{X_k\in\mathcal{X}_j}C_k^2}\right)^2 - \lambda t\right).$$

The minimum with respect to λ is where

$$0 = 2\lambda\left(\sum_j w_j\sqrt{\sum_{X_k\in\mathcal{X}_j}C_k^2}\right)^2 - t,$$

i.e.,

$$\lambda = \frac{t}{2\left(\sum_j w_j\sqrt{\sum_{X_k\in\mathcal{X}_j}C_k^2}\right)^2}.$$

Therefore,

$$P\left(\sum_{k=1}^{N}X_k \geq t\right)$$
$$\leq \exp\left(\frac{t}{2\left(\sum_j w_j\sqrt{\sum_{X_k\in\mathcal{X}_j}C_k^2}\right)^2}\left(\frac{t}{2}-t\right)\right) = \exp\left(-\frac{t^2}{4\left(\sum_j w_j\sqrt{\sum_{X_k\in\mathcal{X}_j}C_k^2}\right)^2}\right).$$

We can get a simpler, though less tight, bound using the Cauchy-Schwarz inequality:

$$\left(\sum_j w_j\sqrt{\sum_{X_k\in\mathcal{X}_j}C_k^2}\right)^2 \leq \left(\sum_j w_j\right)\left(\sum_j w_j\sum_{X_k\in\mathcal{X}_j}C_k^2\right) = \sum_j w_j\sum_{k=1}^{N}C_k^2,$$

where the last equality follows by lemma 4.2. This holds for any exact proper fractional cover, so the theorem follows. □

We can also apply this method to prove a generalisation of Bernstein's inequality.

Theorem 5.3 (Bernstein inequality, dependent case)**.** *(Tanoue 2022, theorem 3.1.) Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a set of subexponential random variables with $E[\exp(\lambda(X_k - E[X_k]))] \leq \exp(\lambda^2 C_k^2)$ for $|\lambda| \leq 1/C_k$. (This holds for example with $C_k = 6e\|X_k - E[X_k]\|_{\psi_1}$, see proposition 2.5). Then, for any $t \geq 0$, we have*

$$P\left(\sum_{k=1}^{N}X_k - E\left[\sum_{k=1}^{N}X_k\right] \geq t\right) \leq \exp\left(-\min\left\{\frac{t^2}{4\chi^*(\mathcal{X})\sum_k C_k^2}, \frac{t}{\max_k C_k}\right\}\right).$$

Proof. We can copy-paste the previous proof up to the formula

$$P\left(\sum_{k=1}^{N} X_k \geq t\right) \leq \exp\left(\lambda^2 \left(\sum_j w_j \sqrt{\sum_{X_k \in \mathcal{X}_j} C_k^2}\right)^2 - \lambda t\right),$$

only that this formula holds only for $|\lambda| \leq 1/\max_k C_k$. We have to account for this limitation when minimising with respect to λ. If this value is in the interval, the minimum is again where

$$\lambda = \frac{t}{2\left(\sum_j w_j \sqrt{\sum_{X_k \in \mathcal{X}_j} C_k^2}\right)^2}.$$

Otherwise, the optimal λ is at an end point of the interval. It is easy to see that it has to be the right end point, i.e.

$$\lambda = 1/\max_k C_k.$$

We will proceed by distinguishing the two cases:

- If

  $$\frac{t}{2\left(\sum_j w_j \sqrt{\sum_{X_k \in \mathcal{X}_j} C_k^2}\right)^2} \leq \frac{1}{\max_k C_k},$$

  i.e.

  $$t \leq \frac{2\left(\sum_j w_j \sqrt{\sum_{X_k \in \mathcal{X}_j} C_k^2}\right)^2}{\max_k C_k},$$

  we get the same bound as in the previous theorem, namely

  $$P\left(\sum_{k=1}^{N} X_k \geq t\right) \leq \exp\left(-\frac{t^2}{4\chi^*(\mathcal{X})\sum_k C_k^2}\right).$$

- Otherwise, i.e. for

  $$t > \frac{2\left(\sum_j w_j \sqrt{\sum_{X_k \in \mathcal{X}_j} C_k^2}\right)^2}{\max_k C_k},$$

  we have instead that

  $$\begin{aligned} P\left(\sum_{k=1}^{N} X_k \geq t\right) &\leq \exp\left(\frac{\left(\sum_j w_j \sqrt{\sum_{X_k \in \mathcal{X}_j} C_k^2}\right)^2}{\max_k C_k^2} - \frac{t}{\max_k C_k}\right) \\ &\leq \exp\left(\frac{t}{2\max_k C_k} - \frac{t}{\max_k C_k}\right) = \exp\left(-\frac{t}{2\max_k C_k}\right). \end{aligned}$$

Thus for all $t > 0$ one of the two bounds holds, so the larger one definitely holds. □

Remark 5.4. *Interestingly, the bound for large t is independent of the dependence structure. But the larger $\chi^*(\mathcal{X})$, the larger t must be for this bound to hold.*

Janson 2004 proves a few other concentration inequalities with proper covers (for bounded random variables). These further results involve not only the expectation but also the variance. Their proof is based on yet other estimates of $E[\exp(\lambda(X_k - E[X_k]))]$, but follows the same structure.

As for McDiarmid's inequality, it can *not* be generalised in the same way as in the above theorems. This was shown by Paulin 2014, based on example 4.12. But there are various workarounds: ibid. himself showed a McDiarmid-type inequality based on HD dependence (his theorem 7.3.4). Zhang and Amini 2022 base their various results on more involved properties of dependency graphs, notably tree partitions.

In both of the theorems that we presented here, the tails of the distribution are of the same order as in the independent case: subgaussian or subexponential, respectively. In the following, we will see that this is not the case if the dependency graph is imperfect, i.e. a thresholded dependency graph.

5.2 Concentration inequalities with soft covers

Similar arguments can also be applied in the case of thresholded dependency graphs or, more precisely, soft covers. Lampert et al. 2018 were the first to do this; later Tanoue 2021 provided better bounds. These are the results we will present here. These are two theorems, based on proposition 3.28 and (the original version of) proposition 3.29, respectively. We will of course also incorporate our generalised version of proposition 3.29.

The first result is for bounded random variables:

Theorem 5.5. *(ibid., theorem 3.1.) Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a set of bounded random variables with $a_k \leq X_k \leq b_k$ a.s. Then for any α-threshold $\gamma > 0$, any $t > 0$ and $\lambda < t$:*

$$P\left(\sum_{k=1}^{N} X_k - E\left[\sum_{k=1}^{N} X_k\right] \geq t\right)$$
$$\leq \exp\left(-\frac{2(t-\lambda)^2}{\chi_{\alpha\leq\gamma}(\mathcal{X})\sum_{k=1}^{N}(b_k - a_k)^2}\right) + \frac{2N\chi_{\alpha\leq\gamma}(\mathcal{X})\gamma\max_k(b_k - a_k)}{\lambda}.$$

Proof. This proof is inspired by the previous ones. Again, assume without loss of generality that $E[X_k] = 0$. This time, we consider a minimal soft (but not fractional) cover $\{\mathcal{X}_j\}_{j=1}^{\chi_{\alpha\leq\gamma}(\mathcal{X})}$ with α-threshold γ; again, we can take it to be exact. For each j, define $Y_j = \sum_{X\in\mathcal{X}_j} X$.
We now apply proposition 3.28 to each $\mathcal{X}_j$ separately. It yields random variables $X_1^*, ..., X_N^*$, covered by sets $\mathcal{X}_j^*$ corresponding to $\mathcal{X}_j$.
Now the idea is to do a case distinction between the "good" events where X_k^* is "close enough" to X_k, and the "bad" events where they are "too different". To this effect, we consider the "good" events to be $E_j = \{\sum_{X_k\in\mathcal{X}_j} |X_k - X_k^*| \leq \lambda\}$. We can now bound the

probability that this does not happen:

$$\begin{aligned}
P(E_j^c) &= P\left(\sum_{X_k \in \mathcal{X}_j} |X_k - X_k^*| \geq \lambda\right) \\
&\leq \frac{E\left[\sum_{X_k \in \mathcal{X}_j} |X_k - X_k^*|\right]}{\lambda} \\
&\leq 2\frac{\max_{X_k \in \mathcal{X}_j}(b_k - a_k)\vec{\alpha}(\mathcal{X}_j)(|\mathcal{X}_j| - 1)}{\lambda} \\
&\leq 2\frac{\max_{X_k \in \mathcal{X}_j}(b_k - a_k)\vec{\alpha}(\mathcal{X}_j)|\mathcal{X}_j|}{\lambda}.
\end{aligned}$$

(The first inequality follows from Markov's inequality; the second one from proposition 3.28.)
We can conclude the following:

$$\begin{aligned}
P\left(\sum_{k=1}^N X_k \geq t\right) &= P\left(\left\{\sum_{k=1}^N X_k \geq t\right\} \cap \bigcap_j E_j\right) + P\left(\left\{\sum_{k=1}^N X_k \geq t\right\} \cap \bigcup_j E_j^c\right) \\
&\leq P\left(\left\{\sum_{k=1}^N X_k - \sum_{k=1}^N X_k^* \geq t - \sum_{k=1}^N X_k^*\right\} \cap \bigcap_j E_j\right) + P\left(\bigcup_j E_j^c\right) \\
&\leq P\left(\left\{\lambda \geq t - \sum_{k=1}^N X_k^*\right\} \cap \bigcap_j E_j\right) + \sum_j P\left(E_j^c\right) \\
&\leq P\left(\sum_{k=1}^N X_k^* \geq t - \lambda\right) + 2\sum_j \frac{\max_{X_k \in \mathcal{X}_j}(b_k - a_k)\vec{\alpha}(\mathcal{X}_j)\chi_{\alpha \leq \gamma}(\mathcal{X}_|)}{\lambda} \\
&\leq P\left(\sum_{k=1}^N X_k^* \geq t - \lambda\right) + 2N\chi_{\alpha \leq \gamma}(\mathcal{X})\frac{\max_{X_k \in \mathcal{X}}(b_k - a_k)\gamma}{\lambda} \\
&\leq \exp\left(-2\frac{(t-\lambda)^2}{\chi_{\alpha \leq \gamma}(\mathcal{X})\sum_{k=1}^N (b_k - a_k)^2}\right) + 2N\chi_{\alpha \leq \gamma}(\mathcal{X})\frac{\max_{X_k \in \mathcal{X}}(b_k - a_k)\gamma}{\lambda},
\end{aligned}$$

where in the last step we used theorem 5.1. □

There are analogous results for subgaussian and subexponential random variables:

Theorem 5.6. *1. (Tanoue 2021, theorem 4.1.) Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a set of subgaussian random variables with c_k such that $P(|X_k - E[X_k]| > t) \leq 2\exp(-t^2/c_k^2)$ (for example $c_k = \|X_k - E[X_k]\|_{\psi_2}$, see proposition 2.4). Then for any α-threshold $\gamma > 0$, any $t > 0$ and $\lambda < t$:*

$$\begin{aligned}
&P\left(\sum_{k=1}^N X_k - E\left[\sum_{k=1}^N X_k\right] \geq t\right) \\
&\leq \exp\left(-\frac{(t-\lambda)^2}{4C^2\chi_{\alpha \leq \gamma}(\mathcal{X})\sum_{k=1}^N c_k^2}\right) + \frac{4}{\lambda}\sum_{k=1}^{N-1}\int_0^{\alpha(X_k, \{X_{k+1}, ..., X_N\})} \sqrt{-c_k^2 \ln(u/2)}du
\end{aligned}$$

for a constant $C \leq 6\sqrt{e}$ (independent of the choice of random variables or of t and λ).

2. *Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a set of subexponential random variables with c_k such that $P(|X_k - E[X_k]| > t) \leq 2\exp(-t/c_k)$ (for example $c_k = \|X_k - E[X_k]\|_{\psi_1}$, see proposition 2.5). Then for any α-threshold $\gamma > 0$, any $t > 0$ and $\lambda < t$:*

$$P\left(\sum_{k=1}^{N} X_k - E\left[\sum_{k=1}^{N} X_k\right] \geq t\right) \leq \exp\left(-\min\left\{\frac{t^2}{4C^2\chi_{\alpha\leq\gamma}(\mathcal{X})\sum_k c_k^2}, \frac{t}{C\max_k c_k}\right\}\right) + \frac{4}{\lambda}\sum_{i=1}^{N-1}\int_0^{\alpha(X_k,\{X_{k+1},...,X_N\})} -c_k \ln(u/2)du$$

for a constant $C \leq 6e$ (independent of the choice of random variables or of t and λ).

Proof. The proof is perfectly analogous to the previous one, except that we apply proposition 3.29 instead of proposition 3.28. In the last step, we use theorem 5.1 in the subgaussian case or theorem 5.3 in the subexponential case. □

Unlike classical dependency graphs, in these results the tails are of order $1/t$ in the best case (because $1/\lambda > 1/t$). This is of the same order as Markov's inequality. For this reason, we should be very careful when we only have a thresholded dependency graph instead of a classical one.

5.3 Assessing optimality

The concentration inequalities above guarantee that $P(\sum_k X_k > t)$ is below a certain bound. But it might be close to this bound or much lower. In other words, the concentration inequalities may or may not be optimal.
Actually, in each situation, we may apply several inequalities: Markov's inequality, an inequality based on proper covers (possibly just with the trivial cover, corresponding to the complete dependency graph), and an inequality based on soft covers (possibly with threshold 0, corresponding to proper covers). Which inequality gives the best results in which situation?
These are the two questions we will examine in this section. In a first subsection, we will examine some examples with proper (fractional) covers or classical dependency graphs, and estimate or simulate the true concentration of their sums. In a second subsection, we provide a theoretical comparison of a concentration inequality with soft covers, namely theorem 5.5, with the Markov inequality and a concentration inequality with proper covers.

5.3.1 Proper covers: some examples and their concentration

Are theorem 5.1 and theorem 5.3 optimal? To give a definitive "yes", we would need an example (ideally a non-trivial one) where the bound is actually reached. To give a

definitive "no", we would need a stronger theorem.
In two extreme examples (full independence and full equality), the bounds are actually reached. But for all other cases, we are aware of neither a definitive "yes" nor a definitive "no". Instead, in the following we will present a few examples and show how most of them are actually more concentrated than what the theorems predict. For simplicity, we will always construct the examples so that $E[X_k] = 0$.

In the following, for simplicity, we will say "case 1" for the subgaussian case and "case 2" for the subexponential case. We will mostly concentrate on the subgaussian case, since often much the same arguments also hold for the exponential case.

First of all, what if the random variables are independent? Then $\chi^*(\mathcal{X}) = 1$, and we recover the theorems from section 2.1. So from this side, we can expect no improvement. The real question is of course what happens if the dependency graph is non-empty.

On the other extreme, assume the X_k are all equal (write X instead). Then the only dependency graph is the complete graph and $\chi^*(\mathcal{X}) = N$. For later reference, note that we have $HD(1, N)$ dependence.
Consider the subgaussian case (with $E[\exp(\lambda X)] \leq \exp(\lambda^2 C^2)$). The theorem guarantees that $P(NX \geq t) \leq \exp\left(-\frac{t^2}{4N^2C^2}\right)$. But we can also directly apply the Cramér-Chernoff method to our example:

$$P(NX \geq t) \leq \exp(-\lambda t)E[\exp(\lambda NX)] \leq \exp(\lambda^2 N^2 C^2 - \lambda t)$$

and this is minimal at $\lambda = \frac{t}{2N^2C^2}$, with value

$$\exp\left(\frac{t^2}{4N^2C^2} - \frac{t^2}{2N^2C^2}\right) = \exp\left(-\frac{t^2}{4N^2C^2}\right).$$

Thus the theorem is optimal in this case as well.
The same arguments apply in the subexponential case.

Now we will go through our examples from chapter 4: the three pairwise independent random variables from example 1.2 (continued in example 4.4 and example 4.11); the double sequence from example 4.12; and various examples with a dependency graph deduced from HD dependence.

Example 5.7. *Consider the three pairwise independent random variables from example 1.2. Remember from example 4.4 that its fractional covering number is 3/2. Therefore, theorem 5.1 implies that* $P(Y_1 + Y_2 + Y_3 - 3/2 \geq t) \leq \exp(-4t^2/9)$.
On the other hand, we can compute these probabilities explicitly. For $t \leq 1/2$, *we have*

$$P(Y_1 + Y_2 + Y_3 - 3/2 \geq t) = P(Y_1 + Y_2 + Y_3 \geq 2) = P(X_1 = X_2 = X_3) = 2/8 = 1/4.$$

The same even holds for $t \leq 3/2$. *But for* $t > 3/2$, *the probability becomes 0.*
If we plug in $t = 3/2$ *into the bound* $\exp(-4t^2/9)$, *we get* $\exp(-1) \approx 0.368 > 1/4$. *This is not bad, but not perfect either. If we wanted the perfect Hoeffding-type bound in this situation, it would be* $\exp(-t^2 4\ln(4)/9)$, *with a further factor* $\ln(4)$ *in the exponential.*
For later reference, note that here we have $HD(2, 2)$ *dependence.*

Example 5.8. *In the double sequence example 4.12, we had a weak dependency graph linking every X_k with the corresponding $X_{n+k} = QX_k$. It is easy to see that its colouring number is 2. Therefore, theorem 5.1 implies that $P(\sum X_k \geq t) \leq \exp(-t^2/(2n))$.*
To compute the probabilities explicitly, note that with probability $1/2$, namely when $Q = -1$, $\sum X_k = 0$. Thus,

$$P(\sum_{k=1}^{2n} X_k \geq t) = \frac{1}{2} P(\sum_{k=1}^{n} X_k \geq t/2).$$

By the original Hoeffding inequality (theorem 2.2), this is less or equal to $1/2 \exp(-t^2/(4n)) = \exp(-t^2/(4n) - \ln(2))$. Which bound is sharper depends on t and n. For large t, it is the $\exp(-t^2/(2n))$ bound that is ultimately sharper.
The situation is analogous if we choose $X_1, ..., X_k$ to be standard normal or exponential instead of Rademacher.
For later reference, here we have $HD(2, n)$ dependence.

Now let us move on to more general examples. Assume we are given any specific (non-empty) graph. We would then like to generate a set of subgaussian or subexponential random variables with this graph as a dependency graph. We will use the general idea from section 4.3 and give several ideas of choices for the edge variables Y_e, the node variables Y_k and the functions f_k to put them together. In all of these cases, we will have $HD(\Delta + 1, 2)$ dependence, where Δ is the maximal degree of the given dependency graph. (Actually, in example 5.9, the dependence is even $HD(\Delta, 2)$, because there are no Y_k associated to the nodes.)

Example 5.9. *Perhaps the simplest idea is to make each Y_e an independent standard normal random variable and let f_k be the sum function. But then $\sum X_k$ is the same as $\sum 2Y_e$, a sum of independent normal random variables. This has tails bounded by $\exp(-t^2/(8M))$, where M is the number of edges. The theorem, on the other hand, only yields the bound*

$$\exp\left(-\frac{t^2}{4(\Delta+1)\sum_k \frac{deg(k)}{2}}\right) = \exp\left(-\frac{t^2}{4M(\Delta+1)}\right).$$

This bound is strictly looser as soon as $\Delta > 1$; if $\Delta = 1$, it is equivalent.
The situation is similar if instead of standard normal we take everything to be standard exponential (minus one, to ensure the mean is zero).
Much the same also happens with the arithmetic mean instead of the sum.

Example 5.10. *Another idea is to sum Bernouilli random variables Y_e first and then to multiply each sum by an independent standard normal Y_k. Then at each node, X_k can be bounded (if positive) by $deg(k)Y_k$. This yields the tail bound*

$$\exp\left(-\frac{t^2}{2\sum_k (deg(k))^2}\right).$$

On the other hand, the theorem yields

$$\exp\left(-\frac{t^2}{2(\Delta+1)\sum_k (deg(k))^2}\right),$$

which is again looser.
Again, the same applies to the standard exponential case.

In example 5.9, we could also take the geometric mean instead of the sum. But then the same type of bound will hold; this follows from the inequality of geometric and arithmetic means.[1] What about a type of mean that is *larger* than the arithmetic mean? We may for example think about

$$f_k(\{Y_e\}) = \sqrt{\frac{1}{deg(k)} \sum_e Y_e^2}.$$

But any type of mean is bounded from above by the maximum of the absolute values, which is in turn bounded by the sum of the absolute values. In the following, we present a slightly more refined example based on the sum of absolute values.

Let $Y_e^{(1)}, Y_k^{(1)}$ for edges e and nodes k be independent random variables with distribution $\mathcal{N}(0,1)$. Analogously, let $Y_e^{(2)}, Y_k^{(2)}$ for edges e and nodes k be independent random variables with distribution $\mathcal{E}(1)$. We define

$$\begin{aligned} X_k^{(1)} &= \frac{1}{deg(k)+1} Z_k \sum_{e,k} |Y_e^{(1)}| \\ X_k^{(2)} &= \frac{1}{deg(k)+1} Z_k \sum_{e,k} Y_e^{(2)}, \end{aligned} \tag{5.3.1}$$

where the sum is over k itself and all edges e adjacent to k, and Z_k are independent Rademacher random variables. (The multiplication by Z_k is to ensure the mean is zero.) It turns out that they are indeed subgaussian and subexponential, respectively:

Lemma 5.11. *1. $X_k^{(1)}$ is subgaussian with $E[\exp(\lambda X_k^{(1)})] \leq \exp(\lambda^2/2)$.*

2. $X_k^{(2)}$ is subexponential with

$$\forall \lambda \in \left[-\sqrt{\frac{e-1}{e}}, \sqrt{\frac{e-1}{e}} \right] : E[\exp(\lambda X_k^{(2)})] \leq \exp\left(\frac{e}{e-1} \lambda^2 \right).$$

Proof. We consider only a single node k, so we can write d for its degree and Z for Z_k.

1. We write X instead of $X_k^{(1)}$ and Y_e instead of $Y_e^{(1)}$ to simplify notation. Analogously to the proof of theorem 5.1, we use Jensen's inequality.

$$\begin{aligned} E[\exp(\lambda X)] &= E\left[\exp\left(\sum_{e,k} \frac{1}{d+1} \lambda Z |Y_e| \right) \right] \\ &\leq \sum \frac{1}{d+1} E[\exp(\lambda Z |Y_e|)] = \frac{d+1}{d+1} E[\exp(\lambda Y_e)] = \exp(\lambda^2/2). \end{aligned}$$

2. We write X instead of $X_k^{(2)}$ and Y_e instead of $Y_e^{(2)}$ to simplify notation. The same argument as above shows that $E[\exp(\lambda X)] \leq E[\exp(\lambda Z Y_e)]$. Now (writing Y for

[1] See e.g. Wikipedia: Ungleichung vom arithmetischen und geometrischen Mittel.

Y_e), for $|\lambda| < 1$,

$$\begin{aligned} E[\exp(\lambda ZY)] &= \frac{1}{2}E[\exp(\lambda Y)] + \frac{1}{2}E[\exp(-\lambda Y)] \\ &= \frac{1}{2}\frac{1}{1-\lambda} + \frac{1}{2}\frac{1}{1+\lambda} = \frac{1}{1-\lambda^2}. \end{aligned}$$

This is equal to $\exp\left(\frac{e}{e-1}\lambda^2\right)$ iff $\lambda^2 \in \{0, \frac{e-1}{e}\}$. Since everything is continuous, for $\lambda^2 \leq (e-1)/e$ we either always have $\frac{1}{1-\lambda^2} \leq \exp\left(\frac{e}{e-1}\lambda^2\right)$ or always $\frac{1}{1-\lambda^2} \geq \exp\left(\frac{e}{e-1}\lambda^2\right)$. By looking at an example, it is easy to convince oneself that it has to be the first option.

□

Simulations for case 1 seem to suggest that no substantially smaller constants are possible (see Jupyter notebook test.ipynb in appendix A.2). Indeed, for some values of d the simulations even give slightly larger constants, which must be due to computational imprecision, so we can hardly trust any smaller values either.

Using theorem 5.1 and theorem 5.3 respectively, we can conclude the following:

Corollary 5.12. *Consider the setup described so far, let d_k be the degree of node k and let Δ be the maximum degree of the dependency graph D. Then:*

1. $$P\left(|\sum_{k=0}^{N} X_k^{(1)}| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2N(\Delta+1)}\right).$$

2. $$P\left(|\sum_{k=0}^{N} X_k^{(2)}| \geq t\right) \leq 2\exp\left(-\min\left\{\frac{(e-1)t^2}{4eN(\Delta+1)}, \sqrt{\frac{e-1}{e}}t\right\}\right),$$

Remark 5.13. *The first bound becomes non-trivial (namely, smaller than 1) at $t = \sqrt{2N(\Delta+1)\ln(2)}$.*
The second bound becomes non-trivial at $t = 2\sqrt{eN(\Delta+1)\ln(2)/(e-1)}$. The subexponential bound only begins to hold at $t = 4N(\Delta+1)\sqrt{\frac{e}{e-1}}$; in the simulations this will be far outside the observable range.

There is no obvious way to compute or otherwise estimate the true values explicitly, so we do simulations.

We fix three different dependency graphs of 50 nodes each: the complete graph, a dense random graph $G(50, 0.5)$, and a sparse random graph $G(50, 0.1)$. For each of these graphs, we simulate 100 000 independent copies of the random vectors $(X_1^{(1)}, ..., X_{50}^{(1)}), (X_1^{(2)}, ..., X_{50}^{(2)})$ described above. Denote the value of the s-th copy of $X_k^{(1)}$ by $x_{ks}^{(1)}$. We then provide point estimates for $P(|\sum X_k| \geq t)$, for 50 values of t. (For each dependency graph, we choose a different range of t that provides meaningful

information.) These point estimates are just

$$\frac{1}{100000}\sum_s \widehat{p}_s := \frac{1}{100000}\sum_s \mathbf{1}_{\sum_k x_{ks}>t}.$$

Of course, these point estimates are not exactly the true values. But we can apply Hoeffding's inequality to get intervals where the true value is with at least 95% probability. Concretely, in our case, if p is the true value, then

$$P\left(|\frac{1}{100000}\sum_s \widehat{p}_s - p| \geq \epsilon\right) = P\left(|\sum_s \widehat{p}_s - 100000p| \geq 100000\epsilon\right) \leq 2\exp\left(-200000\epsilon^2\right).$$

Setting the right hand side to be 0.05, we get

$$\epsilon = \sqrt{\frac{\ln(40)}{200000}} = 0.004294694$$

approximately.

The Python code can be found in the files depgraph.py (appendix A.1) and depgraph.ipynb (appendix A.3).

We show the results of these simulations in fig. 5.1. The plots show that the bounds of the theorems are much too loose for this particular example, especially when the dependency graph is dense. We can readily exclude two simple explanations based on the theorems themselves: there are no better subgaussian / subexponential norms, and at least in the complete graph, the fractional chromatic number is really equal to $\Delta + 1$. Therefore it is the theorems themselves that are non-optimal in our case.

If these examples are representative of what we can get with the assumptions of subgaussianity or subexponentiality and dependency graphs (and this is a big *if*!), this suggests that there are significantly better, though yet unknown, concentration inequalities waiting for us.
Consider the different variants of $HD(m,n)$ dependence that we have encountered. It seems that the theorems from section 5.1 are (more or less) optimal when most of the dependence is in the n factor (the maximal number of places a hidden variable appears). An example is the extreme case where all terms are equal, where the dependence is $HD(1,N)$. On the other hand, the concentration apparently does not depend so much on the m factor (the maximal number of hidden variables determining an overt variable). This suggests the following conjecture (or a similar one):

Conjecture 1. *Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a set of mean zero subgaussian random variables with $E[\exp(\lambda X_k)] \leq \exp(\lambda^2 C_k^2)$. Assume that they are $HD(m,n)$ dependent. Then, for any $t \geq 0$, we have*

$$P\left(\sum_{k=1}^N X_k \geq t\right) \leq \exp\left(-\frac{t^2}{4n\sum_k C_k^2}\right).$$

In particular, the tail does not depend on m.

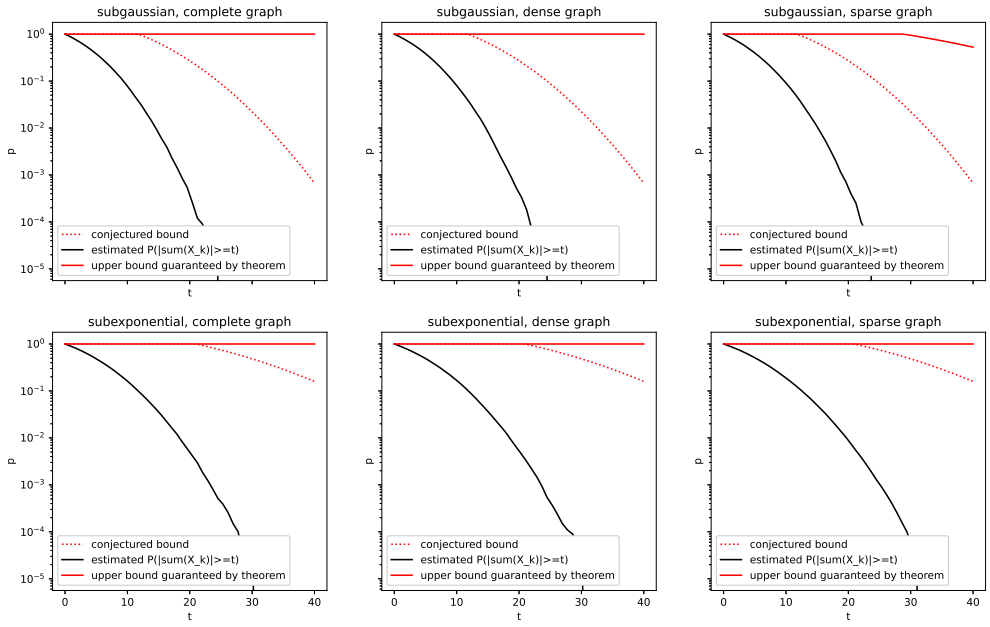


Figure 5.1: Comparison of theorem, conjecture and estimated true concentration, for the example described in eq. (5.3.1). The black line represents the estimated true concentration. The red line represents the bound guaranteed by corollary 5.12; note that in the observed range the bound is mostly trivial and hence replaced by the constant 1. The dotted red line represents conjectures 1 and 2 respectively. The first row corresponds to the subgaussian case (case 1) and the second line to the subexponential case (case 2). The three columns represent the three different dependency graphs. Notice the logarithmic scale in the vertical axis.

However surprising it seems, this conjecture is consistent with all the examples we have encountered. If the bound were correct, it would be stronger than theorem 5.1 if and only if the second coefficient (n) of HD dependence is smaller than the fractional chromatic number.
Going through our examples where $n < \chi^*$, we can confirm that they are consistent with the conjecture. In example 5.9, we know that the tail is actually bounded by $\exp(-t^2/(8M))$, and this is exactly what the conjecture yields as well. In example 5.10, we know that the tail is actually bounded by $\exp(-t^2/(2\sum(deg(k))^2))$, and the conjecture yields the bound $\exp(-t^2/(4\sum(deg(k))^2))$, which is larger. For the simulated examples, fig. 5.1 shows the conjecture as a dotted red line.

Analogously, we also conjecture the following for the subexponential case (inspired by theorem 5.3); the same comments apply:

Conjecture 2. *Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a set of mean zero subexponential random variables with $E[\exp(\lambda X_k)] \leq \exp(\lambda^2 C_k^2)$ for $|\lambda| \leq 1/C_k$. Assume that they are $HD(m, n)$ dependent. Then, for any $t \geq 0$, we have*

$$P\left(\sum_{k=1}^{N} X_k \geq t\right) \leq \exp\left(-\min\left\{\frac{t^2}{4n\sum_k C_k^2}, \frac{t}{\max_k C_k}\right\}\right).$$

In particular, the tail does not depend on m.

As an additional empirical argument in favour of the conjectures, we note that in the simulations the behaviour of the tails is very similar regardless of how dense the graph is.

5.3.2 Soft covers: comparing with simpler inequalities

Now let us proceed to the inequalities with soft covers, from section 5.2. Here the tails are of order $1/t$ at best. So how do they compare to Markov's inequality?

In the following, for simplicity, assume that the X_k have mean zero, and write χ for $\chi_{\alpha\leq\gamma}(\mathcal{X})$. We assume of course that $\gamma \neq 0$, otherwise we recover the inequalities from section 5.1.

For simplicity, we will only consider the version for bounded random variables, theorem 5.5.

On the one hand, this theorem tells us that (for all $\lambda < t$)

$$\begin{aligned} &P\left(|\sum_{k=1}^{N} X_k| \geq t\right) \\ &\leq 2\exp\left(-\frac{2(t-\lambda)^2}{\chi\sum_{k=1}^{N}(b_k-a_k)^2}\right) + \frac{4\chi\gamma N\max_k(b_k-a_k)}{\lambda}. \end{aligned}$$

On the other hand, Markov's inequality tells us that

$$\begin{aligned} P\left(|\sum_{k=1}^{N} X_k| \geq t\right) &\leq \frac{E\left[|\sum X_k|\right]}{t} \\ &\leq \frac{\sum \|X_k\|_\infty}{t} \leq \frac{\sum(b_k-a_k)}{t} \leq \frac{N\max_k(b_k-a_k)}{t}. \end{aligned}$$

Thus theorem 5.5 stands a chance only if

$$\frac{4\chi\gamma}{\lambda} < \frac{1}{t},$$

in other words $\lambda > 4\chi\gamma t$. This requires in particular that $4\chi\gamma < 1$.
We get a very similar result for $P\left(\sum X_k \geq t\right)$ (without the absolute value); to this effect we can use $E[\sum X_k \mathbf{1}_{\sum X_k > 0}]$ instead of $E[\sum |X_k|]$.

On the other hand, we also need to compare this theorem to theorem 5.1. Indeed, we can always apply theorem 5.1 with the complete dependency graph.
It is clear that for very small and very large t, theorem 5.1 is better: for small t, the Markov-type term blows up, and for large t, it is larger than any Hoeffding-type bound. But there may be a range of t's in between, where (for some appropriate $\lambda \in]4\chi\gamma t, t[$)

$$\exp\left(-\frac{2(t-\lambda)^2}{\chi\sum_{k=1}^{N}(b_k-a_k)^2}\right) + \frac{2\chi\gamma N\max_k(b_k-a_k)}{\lambda} < \exp\left(-\frac{2t^2}{N\sum_{k=1}^{N}(b_k-a_k)^2}\right).$$

Putting in numerical values, we can see that there is indeed such a range in certain cases. For example, if $[a_k, b_k] = [0, 1], \vec{\alpha}(\{X_k\}) = 1/16$, and $N \geq 32$, then we can take $\gamma < 1/16$, $\chi = 1$, $\lambda = 5t/8$; then a WolframAlpha computation shows that

$$\exp\left(-\frac{2(t-\lambda)^2}{\chi \sum_{k=1}^{N}(b_k - a_k)^2}\right) + \frac{2\chi\gamma N \max_k(b_k - a_k)}{\lambda} = \exp\left(-9(t/N)^2\right) + \frac{1}{5}(t/N)^{-1}$$
$$< \exp\left(-\frac{2t^2}{N^2}\right) = \exp\left(-2(t/N)^2\right)$$

for t/N between 0.406169 and 0.847915.

These theorems are therefore useful only to estimate the probabilities of a narrow range medium-sized deviations. For small and large deviations, it is better to use the theorems from section 5.1.

In this context, we should also point out that it is often difficult to compute or estimate α-separation coefficients. This makes these theorems even less helpful in practice.

Still, it is worth investigating further if we can improve the theorems of section 5.1 when the dependencies are weak. This is the purpose of the following subsection.

5.3.3 Examples with weak dependencies and their concentration

Let us adapt the simulated example from section 5.3.1. This time, we give a special status to the random variables Y_k associated to the nodes, as opposed to the Y_e associated to the edges. More precisely, instead of the arithmetic mean we consider a weighted arithmetic mean, where Y_k gets half of the weight on its own. It is difficult to explicitly compute α-separation coefficients in this setup, but intuitively, it is clear that our adaptation makes the dependencies between the X_k weaker, particularly when the corresponding nodes have high degrees. Indeed, the dependencies arise because of the Y_e and not because of the Y_k.

Concretely, define, for this section,

$$\begin{aligned} X_k^{(1)*} &= \frac{1}{2} Z_k \left(|Y_k^{(1)}| + \frac{1}{deg(k)} \sum_e |Y_e^{(1)}| \right), \\ X_k^{(2)*} &= \frac{1}{2} Z_k \left(|Y_k^{(1)}| + \frac{1}{deg(k)} \sum_e Y_e^{(2)} \right). \end{aligned} \tag{5.3.2}$$

Of course, if k has no edges, we just define $X_k^* = Z_k |Y_k|$.

Their subgaussian or subexponential norms, respectively, are the following:

Lemma 5.14. *1. $X_k^{(1)*}$ is subgaussian with $E[\exp(\lambda X_k^{(1)*})] \leq \exp(\lambda^2/2)$.*

2. $X_k^{(2)*}$ is subexponential with

$$\forall \lambda \in \left[-\sqrt{\frac{e-1}{e}}, \sqrt{\frac{e-1}{e}} \right] : E[\exp(\lambda X_k^{(2)})] \leq \exp\left(\frac{e}{e-1} \lambda^2 \right).$$

Proof. We consider only a single node k, so we can write d for its degree and Z for Z_k. We assume that $d > 0$; otherwise the result is trivial. We proceed analogously to the other lemma.

1. We write X instead of $X_k^{(1)}$ and Y_e instead of $Y_e^{(1)}$ to simplify notation.
   We use Jensen's inequality:

$$E[\exp(\lambda X)] = E\left[\exp\left(\frac{1}{2}Z|Y_k| + \sum_e \frac{1}{2d}\lambda Z|Y_e|\right)\right]$$
$$\leq \frac{1}{2}E[\exp(\lambda Z|Y_k|)] + \frac{1}{2d}\sum E[\exp(\lambda Z|Y_e|)] = E[\exp(\lambda Y_e)] = \exp(\lambda^2/2).$$

2. We write X instead of $X_k^{(2)*}$ and Y_e instead of $Y_e^{(2)*}$ to simplify notation. The same argument as above shows that $E[\exp(\lambda X)] = E[\exp(\lambda Z Y_e)]$. The result follows as in the previous lemma.

□

Intuitively, we expect that in this example the behaviour will be even closer to independence, i.e. the tails will decay faster.

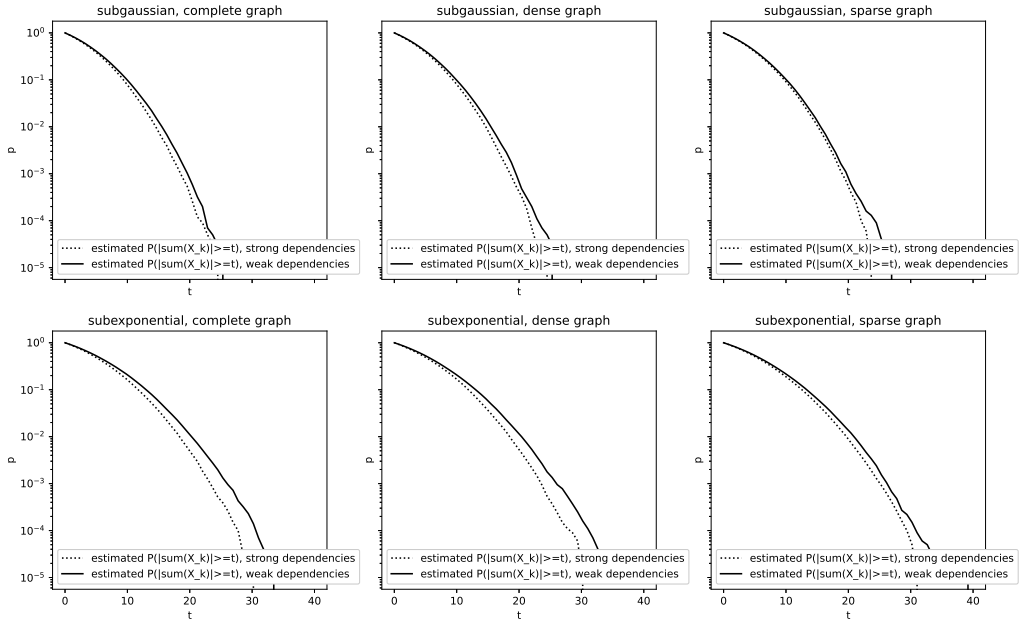


Figure 5.2: Comparison of estimated true concentrations between the examples from eqs. (5.3.1) and (5.3.2) (the second one having weaker dependencies). The dotted lines represent the first example (stronger dependencies, same as the solid black lines in fig. 5.1), and the solid ones represent the second example (weaker dependencies). The first row corresponds to the subgaussian case (case 1) and the second line to the subexponential case (case 2). The three columns represent the three different dependency graphs. Notice the logarithmic scale in the vertical axis.

However, this expectation is contradicted by the results of the simulation. fig. 5.2 compares these simulations to the ones from fig. 5.1, and it turns out there is no big difference. Indeed in some cases the tails even seem to decay (slightly) slower than with strong dependencies. But keeping in mind the "confidence interval" of roughly 4×10^{-3} on each side of the curves, there is really no difference at all.

If this example is representative of weak dependencies, it must warn us again not to use independence assumptions too carelessly.

5.4 Applications to statistical learning

As already described in section 2.1.5, a typical application of concentration inequalities is in statistical learning. In textbook examples, the model is trained on i.i.d. samples and the goal is to approximate their common distribution. Here we will show examples of how samples can be graph-dependent, and how the concentration inequalities described above help us in this case. We have adapted these examples from Zhang and Amini 2022. These authors (and those that they cite) prove rather involved generalisation bounds, involving properties of the sample and of the hypothesis class. This is beyond our scope, so we will stick to simpler results, such as validation error bounds.

5.4.1 Bipartite ranking

In bipartite ranking, our original datapoints are again of the form (x_k, y_k), where $y_k \in \{\pm 1\}$. But now, we do not just want to classify new datapoints, but to rank them according to how "suspicious" they are. Concretely, we want a function $h : \mathcal{X} \to \mathbb{R}$ such that higher values of $h(x)$ make it more likely that $y = 1$.
Such a ranking may for example be useful for a bank, to help it discover fraud in electronic transactions. A ranking function would yield a list of today's transaction according to how suspicious they seem, and then human employees could check the n first ones in more detail.

One can examine this problem from several angles and define the loss function in several ways (Shalev-Shwartz and Ben-David 2014, section 17.5). Now, Zhang and Amini 2022 consider yet another framework:
The original sample T contains independent datapoints of the type (x_k, y_k). From it we construct another sample S with datapoints of the form (x^+, x^-). Here x^+ runs through all x_k such that $y_k = 1$, and x^- runs through all x_k such that $y_k = -1$. If there are N^+ x^+ and N^- x^-, the sample size of S is N^+N^-. Translating the problem to this framework, we want a function $f(x_k, x_j) = h(x_k) - h(x_j)$ such that, ideally, $f(x_k, x_j) > 0$ iff $y_k = 1, y_j = -1$. (This condition already implies that $f(x_j, x_k) < 0$ in this case, and that $f(x_k, x_j) = 0$ whenever $y_k = y_j$.) Therefore, we can define a loss function $L((y_k, y_j), (y'_k, y'_j)) = \mathbf{1}_{(y_k - y_j)(y'_k - y'_j) \leq 0}$ which yields $L((y_k, y_j), f(x_k, x_j)) = \mathbf{1}_{(y_k - y_j) f(x_k, x_j) \leq 0}$. The empirical risk is then

$$R_S(f) = \frac{1}{N^+N^-} \sum_{(x^+, x^-) \in S} \mathbf{1}_{f(x^+, x^-) \leq 0},$$

while the true risk is $R(f) = E[R_S(f)]$.

Suppose we have trained an algorithm in this framework and found a hypothesis h, or equivalently, f. Now we want to estimate its true risk using validation. We can sample N_V^+ new datapoints from the distribution restricted to $y = +1$, and N_V^- new negative datapoints. (For the following argument it is necessary to fix N_V^+ and N_V^- deterministically, i.e. before sampling. One possible way to do it in practice is to continue sampling until we have enough datapoints of both categories, and then keep the N_V^+ first positive ones and the N_V^- first negative ones.)
From this we can construct a validation sample S_V with datapoints of the type (x^+, x^-), like the training sample S.

Now the $N_V = N_V^+ N_V^-$ datapoints (or rather, the underlying random variables of the type (X^+, X^-)) admit a dependency graph D, in which (X_k^+, X_k^-) is linked to (X_j^+, X_j^-) whenever $X_k^+ = X_j^+$ or $X_k^- = X_j^-$. The maximal degree of D is $\max(N_V^+, N_V^-) - 1$.
There can be no fractional colouring with weight sum less than $\max(N_V^+, N_V^-)$. Indeed, any subset $\mathcal{X}_j$ of independent random variables has at most $\min(N_V^+, N_V^-)$ elements. Therefore, by lemma 4.2, for any proper fractional cover (with weights w_j) we have $N_V \leq \sum_j w_j \min(N_V^+, N_V^-)$ and hence $\sum_j w_j \geq \max(N_V^+, N_V^-)$.
Therefore, the fractional colouring number of D is $\chi^*(D) = \max(N_V^+, N_V^-)$.
Alternatively, we can describe the variables (X_k^+, X_k^-) as being $HD(2, \max(N_V^+, N_V^-))$ dependent. This means that conjecture 1 wouldn't help us in this case, even if it turned out to be true.

Therefore, by theorem 5.1, we can conclude that

$$\begin{aligned} P(|R_{S_V}(f) - R(f)| \geq \epsilon) &= P(|N_V R_{S_V}(f) - N_V R(f)| \geq N_V \epsilon) \\ &\leq 2\exp\left(-\frac{2N_V^2\epsilon^2}{N_V \max(N_V^+, N_V^-)}\right) = 2\exp\left(-2\min(N_V^+, N_V^-)\epsilon^2\right). \end{aligned}$$

Equivalently, with probability at least $1-\delta$ over the choice of the validation set, we have

$$|R_{S_V}(f) - R(f)| < \sqrt{\frac{\ln(2) - \ln(\delta)}{2\min(N_V^+, N_V^-)}}.$$

Yet another equivalent formulation: In order to have the bound $|R_{S_V}(f) - R(f)| < \epsilon$ with probability at least $1-\delta$, it is enough to choose N_V^+, N_V^- such that

$$\min(N_V^+, N_V^-) \geq \frac{\ln(2) - \ln(\delta)}{2\epsilon^2}.$$

A somewhat surprising result is that once we have a certain large enough number of positive instances and the same number of negative instances, it is useless to sample more instances from one of the two classes (since it does not change the minimum). This can help us avoid unnecessary computations.

In a similar way, we can get a generalisation bound for finite hypothesis classes. To this effect, we can use theorem 5.1 and the union bound.

5.4.2 Data assumed to be m-dependent

In many real-world statistical learning situations, we cannot assume that even the original data are independent. This occurs for example if they are distributed in time and/or

space, and datapoints close to each other tend to have similar values. The example mentioned in Zhang and Amini 2022 is a sequence of houses along a street (the target variable being their price). Furrer 2023 mentions a lot of similar examples of temporally or spatially dependent data, such as the concentration of CO2 in the atmosphere at various points in time.

A common assumption in such cases is that these datapoints are m-dependent. Despite the doubts we have expressed about the suitability of this assumption, let us follow it for now. We will show a bound on the validation error for binary classification (with the 0-1 loss as usual).

Say our data is an m-dependent sequence $\{(x_k, y_k)\}_{k=1}^{N}$, taken from a much longer or even infinite m-dependent sequence (where k may go well beyond N). After training, we find a hypothesis h. To find a validation set, we can for example sample the N_V following points of the sequence.

The minimal dependency graph D of this validation set will link X_k with X_j if and only if $|k - j| \leq m$. Thus each node has at most m neighbours. Assuming $N_V \geq m$, the covering number is also m, because the m first elements must be in different independent subsets.

Therefore, by theorem 5.1, we can conclude that

$$\begin{aligned} P(|R_{S_V}(h) - R(h)| \geq \epsilon) &= P(|N_V R_{S_V}(h) - N_V R(h)| \geq N_V \epsilon) \\ &\leq 2\exp\left(-\frac{2N_V^2\epsilon^2}{N_V m}\right) = 2\exp\left(-\frac{2N_V\epsilon^2}{m}\right). \end{aligned}$$

Equivalently, with probability at least $1 - \delta$ over the choice of the validation set, we have

$$|R_{S_V}(h) - R(h)| < \sqrt{\frac{m(\ln(2) - \ln(\delta))}{2N_V}}.$$

Yet another equivalent formulation: In order to have the bound $|R_{S_V}(h) - R(h)| < \epsilon$ with probability at least $1 - \delta$, it is enough to choose N_V such that

$$N_V \geq \frac{m(\ln(2) - \ln(\delta))}{2\epsilon^2}.$$

In particular, we need a validation set of size proportional to m (with a constant determined by ϵ and δ as in the formula).

But what if we want to be perfectly rigorous and take into account that there may be, however weak, dependencies even beyond a distance of m?

With the knowledge that we have (compare section 5.3.2), the best thing we can do is probably to use theorem 5.1 with the complete dependency graph (fractional chromatic number N_V instead of m). This guarantees the bound

$$P(|R_{S_V}(h) - R(h)| \geq \epsilon) \leq 2\exp\left(-2\epsilon^2\right),$$

which does not improve with N_V at all! The same holds for the Markov bound we derived in section 5.3.2.

There may be room for substantially improving concentration inequalities for thresholded dependency graphs. This must involve introducing a narrower definition in the first place. Such improved inequalities would hopefully yield usable bounds for learning from spatially or temporally dependent data, even without the relatively simplistic assumption of m-dependence.

Chapter 6

Generalised Central Limit Theorems

In this section we present some central limit theorems in the dependency graph setting. They are comparable to the classical central limit theorem because they also state that some S'_n, defined as a properly normalised sum of random variables, will converge in distribution to the standard normal distribution. We will begin with the simplest version, following Janson 1988.

6.1 Janson's Central Limit Theorem

These results typically rely on the method of cumulants (corollary 3.9). Concretely, we need to show that $\kappa_j(S'_n) \to_n 0$ for $j > 2$. This is possible for example if we can prove an upper bound to these cumulants. For example, ibid. showed the following:

Lemma 6.1. *(ibid., lemma 4.) Let $X_1, ..., X_N$ be bounded random variables bounded in absolute value by A. Let D be a dependency graph for $\{X_k\}$ and Δ its maximal degree. Then*

$$|\kappa_j \left(\sum X_i\right)| \leq C_j N(\Delta + 1)^{j-1} A^j.$$

for some C_j depending on j only.

Proof. We expand the cumulant into joint cumulants:

$$\kappa_j \left(\sum X_i\right) = \kappa \left(\sum X_i, ..., \sum X_i\right) = \sum_{i_1} ... \sum_{i_j} \kappa(X_{i_1}, ..., X_{i_j}).$$

Now, whenever the $X_{i_1}, ..., X_{i_j}$ are *not* a connected subgraph in D, we can split them into two sets independent of each other and therefore their joint cumulant is zero. It remains to consider those tuples $(X_{i_1}, ..., X_{i_j})$ that do form a connected subgraph.
How many of these tuples are there? Suppose $(X_{i_1}, ..., X_{i_j})$ form a connected subgraph. We may create any such tuple (up to ordering) by taking first X_{i_1} (N possibilities), then a neighbour X_{i_2} of X_{i_1} or X_{i_1} itself (at most $\Delta + 1$ possibilities), then a X_{i_3} connected or equal to X_{i_1} or X_{i_2} (at most $2\Delta + 2$ possibilities), and so on. Therefore, up to ordering, there are at most $N \cdot (\Delta + 1) \cdot (2\Delta + 2) \cdot ... \cdot ((j-1)\Delta + j - 1) = (j-1)! N(\Delta + 1)^{j-1}$ such tuples. If we take ordering into account, this gets multiplied by at most $j!$.

It remains to bound $\kappa(X_{i_1}, ..., X_{i_j})$. We write X_k instead of X_{i_k} for simplicity. Using eq. (3.1.1) and the generalised Hölder inequality, we have

$$\begin{aligned}
|\kappa(X_{i_1}, ..., X_{i_j})| &= |\sum_{I_1,...,I_n} (-1)^{n-1}(n-1)! \prod_{r=1}^{n} E\left[\prod_{k\in I_r} X_k\right]| \\
&\leq \sum_{I_1,...,I_n} (n-1)! \prod_{r=1}^{n} E\left[\prod_{k\in I_r} |X_k|\right] \\
&\leq \sum_{I_1,...,I_n} (n-1)! \prod_{r=1}^{n} \prod_{k\in I_r} \|X_k\|_{|I_r|} \\
&\leq \sum_{I_1,...,I_n} (n-1)! \prod_{r=1}^{n} \prod_{k\in I_r} \|X_k\|_j \\
&= \sum_{I_1,...,I_n} (n-1)! \prod_{k=1}^{j} \|X_k\|_j \\
&\leq C'_j \prod_{k=1}^{j} \|X_k\|_j \\
&\leq C'_j A^j
\end{aligned}$$

for some C'_j depending only on j.
Putting it all together, we get the result. □

This serves to prove the following central limit theorem:

Theorem 6.2 (Janson's Central Limit Theorem)**.** *(Janson 1988) For each n, let $\{X_{ni}\}_{i=1}^{N_n}$ be a family of bounded random variables (bounded in absolute value by a constant A_n which may depend on n). Let D_n be a dependency graph for $\{X_{ni}\}_i$. Let Δ_n be the maximal degree of D_n (or 1 if D_n is the empty graph). Let $S_n = \sum_k X_{nk}$.*
Additionally, assume that

$$(N_n/\Delta_n)^{1/3}\Delta_n A_n/\sqrt{\mathrm{Var}(S_n)} \to_{n\to\infty} 0.$$

Then $(S_n - E[S_n])/\sqrt{\mathrm{Var}(S_n)}$ converges in distribution to $\mathcal{N}(0,1)$.

Proof. We assume without loss of generality that $E[X_{nk}] = 0$ and $\mathrm{Var}(X_{nk}) = 1$. Otherwise we can replace X_{nk} by $(X_{nk} - E[X_{nk}])/\sqrt{\mathrm{Var}(X_{nk})}$; this is bounded in absolute value by $2A_n/\sqrt{\mathrm{Var}(X_{nk})}$ instead of A_n.
By corollary 3.9, it is enough to show that $\kappa_j(S_n) \to_n 0$ for $j \geq 3$.
By the above lemma, we have:

$$\begin{aligned}
|\kappa_j(S_n)| &\leq C_j N_n(\Delta_n + 1)^{j-1} A_n^j = C_j(N_n^{1/j}(\Delta_n+1)^{1-1/j}A_n)^j \\
&\leq C_j 2^{j-1}(N_n^{1/j}\Delta_n^{1-1/j}A_n)^j = C_j 2^{j-1}((N_n/\Delta_n)^{1/j}\Delta_n A_n)^j \\
&\leq C_j 2^{j-1}((N_n/\Delta_n)^{1/3}\Delta_n A_n)^j \to_n 0,
\end{aligned}$$

where in the last line we used that $\Delta_n < N_n$. □

Remark 6.3. *Janson 1988, theorem 2 actually shows that this theorem is true even if we replace the 3 in the exponent by any $j \geq 3$. Because $\Delta_n < N_n$, this is a weaker condition and hence easier to check. It follows from remark 3.10.*

A few years before Janson's article, Petrovskaya and Leontovich 1983 already proved a very similar theorem (their theorem 2). They directly used the count of connected subgraphs instead of estimating it by $C_j N(\Delta+1)^{j-1}$. It was partly for this reason that they not see that it suffices to check a single exponent $j \geq 3$. Also, they were unaware of the method of cumulants, so their proof was more involved. These two points were therefore Janson 1988's main contributions (but he was apparently unaware of the previous work).

6.2 Extensions

There are at least two different ways of extending this result to unbounded random variables. ibid., remark 3 himself proposes a truncation method. Here we will present another idea of our own (though we may be not the first to have it).

Our method relies on the fact that the proof of lemma 6.1 only requires a bound on the *moments* of X_k. Thus we can state the following more general lemma:

Lemma 6.4. *Let $X_1, ..., X_N$ be subexponential random variables with $A = \max_k \|X_k\|_{\psi_1}$. Let D be a dependency graph for $\{X_k\}$ and Δ its maximal degree. Then*

$$|\kappa_j \left(\sum X_i \right)| \leq C_j N (\Delta+1)^{j-1} A^j.$$

for some C_j depending on j only.

Proof. We follow the proof of lemma 6.1. Instead of the last inequality, we recall that by property 3 of proposition 2.5, $\|X_k\|_j \leq Cj\|X_k\|_{\psi_1} \leq CjA$ for a universal constant C. We conclude that

$$C'_j \prod_{k=1}^{j} \|X_k\|_j \leq C'_j C^j j^j A^j,$$

and the constant $C'_j C^j j^j$ also depends on j only. □

Analogously to theorem 6.2 (and remark 6.3), we can conclude the following:

Theorem 6.5. *For each n, let $\{X_{nk}\}_{k=1}^{N_n}$ be a family of subexponential random variables with subexponential norms bounded by A_n (which may depend on n). Let D_n be a dependency graph for $\{X_{ni}\}_i$. Let Δ_n be the maximal degree of D_n (or 1 if D_n is the empty graph). Let $S_n = \sum_k X_{nk}$.*
Additionally, assume that, for some $j \geq 3$,

$$(N_n/\Delta_n)^{1/j} \Delta_n A_n / \sqrt{\mathrm{Var}(S_n)} \to_{n\to\infty} 0.$$

Then $(S_n - E[S_n])/\sqrt{\mathrm{Var}(S_n)}$ converges in distribution to $\mathcal{N}(0,1)$.

We could clearly extend this even further with the appropriate definitions. For example, for any fixed $\alpha > 1$ we could consider those distributions that satisfy the bound $\forall p \geq 1 : \|X\|_p \leq Cp^\alpha A$. So-called Orlicz spaces (see e.g. Vershynin 2018, section 2.7.1) are likely to be a useful notion in this context. The essential thing is that all X_{nk} must belong to the same such class.

There have been other extensions in different directions: Mikhailov 1992 has extended Janson's argument to more general quantities than N_n and Δ_n. Féray 2016 has extended Mikhailov's theorem to his own invention of weighted dependency graphs.

6.3 Applications

Generalised central limit theorems can help to understand the approximate behaviour of large real-world systems. In this section we will present the example that originally motivated Petrovskaya and Leontovich 1983 to prove their central limit theorem with dependency graphs.

Here the system that we want to describe is a group of cells descending from a common ancestor. Roughly speaking, we want to show that the total time lag between cells of a given generation is approximately normally distributed.

More precisely, we consider the following model. We start with a single cell (generation zero) that divides into cells numbered 0 and 1 (generation 1). We give each cell a bit string (which we may interpret as a binary number): the cell numbered $k_1...k_n$ divides into $k_1...k_n0$ and $k_1...k_n1$ (no typo).
Each cell $k = (k_1...k_n)$ has an i.i.d. random lifetime $T_{n,k}$ until it divides. Thus it divides at time $U_{n,k} = U_{(k_1,...,k_n)} := T_{1,k_1} + ... + T_{n,(k_1,...,k_n)}$ (where time 0 is when the first cell divides). We can then define the lag between the division times of two cells of the same generation: $X^{(d)}_{n,k} := (U_{n,k} - U_{n,k-d})^2$. Finally, a sum of such lags in generation n is

$$S^{(d)}_n = \sum_{k=0^n}^{1^n-d} X^{(d)}_{n,k+d}.$$

It is helpful to think of the process as a tree, where each cell $k_1...k_n$ is a node and is linked to its two descendants $k_1...k_n0$ and $k_1...k_n1$.

In the following, we consider the asymptotic behaviour of $S^{(1)}_n$, which we will denote S_n for simplicity. We will show that, if the T_k are (say) subexponential, it satisfies the conditions of a generalised central limit theorem, in our case theorem 6.5. (Of course, ibid. had to use their own theorem.)

To avoid too big a clutter of abstract formulas, assume $T_k \sim \mathcal{E}(1)$, which is a reasonable assumption for random times. The same arguments hold more generally.

First of all, we have to compute the expectation and variance of S_n. ibid. show that

$$E[S_n] = \mathrm{Var}(T)\Sigma_n = \Sigma_n;$$

here T stands for any of the identically distributed T_k, and Σ_n is twice the sum over all k of the number of generations to go back to find the last common ancestor of k and $k+1$.

They also show that

$$\mathrm{Var}(S_n) = \sigma^{n,1} E[(T - E[T])^4] + (2\sigma^{n,2} - 3\sigma^{n,1}) Var(T)^2 = 6\sigma^{n,1} + 2\sigma^{n,2};$$

here $\sigma^{n,1}$ is the sum, over all pairs j_1, j_2 of cells in generation n, of the number of common edges in the paths in the tree leading from j_1 to $j_1 + 1$ and from j_2 to $j_2 + 1$; and $\sigma^{n,2}$ is the same but squaring each of the terms in the sum.
Finally they show that as n tends to infinity, $\mathrm{Var}(S_n)$ behaves as 2^n.

The $U_{n,k}$ are sums of independent exponentials, so they are exponential with mean n. The $X_{n,k}$ therefore behave like squared exponential random variables; for any p we have

$$\begin{aligned}\|X_{n,k}\|_p &\leq \|U_{n,k}^2\|_p + \|U_{n,k-1}^2\|_p + 2\|U_{n,k}U_{n,k-1}\|_p \\ &\leq 2\|U_{n,k}\|_{2p}^2 + 2\|U_{n,k}^p\|_2^{1/p}\|U_{n,k-1}^p\|_2^{1/p} = 4\|U_{n,k}\|_{2p}^2 \\ &\leq 4C^2(2p)^2 n^2.\end{aligned}$$

We can thus set $A_n = n^2$.

When are $X_{n,k}$ and $X_{n,j}$ independent of each other? It is when the paths in the tree leading from k to $k-1$ and from j to $j-1$ have no edge in common. Equivalently, the last common ancestors of the pair $(k, k-1)$ and of the pair $(j, j-1)$ must not belong to the same line (one must not be a descendant of the other).

How then does a dependency graph D_n of the $\{X_{n,k}\}_{k=0^{n-1}1,\ldots,1^n}$ look like?
Each $X_{n,k}$ must be linked to its neighbour $X_{n,k-1}$.
Furthermore, if the last digit of k is 0, then the path from k to $k-1$ goes one level higher in the tree. Therefore we must also link $X_{n,k}$ to all those other nodes that begin with the same digits $k_1 ... k_{n-2}$ and also end with 0.
If even the two last digits are 00, the path goes even higher. We must link $X_{n,k}$ to all those other nodes that begin with the same digits $k_1 ... k_{n-3}$ and also end with 00.
Continuing this way, we get a dependency graph. In all of this, we of course exclude the cell 0^n, because there is no $X_{n,0}$.
The maximum degree of this dependency graph is the degree of $X_{n,2^{n-1}}$ (because 2^{n-1} corresponds to the binary 10...0). This node has got an additional edge at each step in the procedure, and there are n steps. Therefore $\Delta_n = n$.

Putting it all together, we can verify the condition from theorem 6.5:

$$\lim_n (N_n/\Delta_n)^{1/3} \Delta_n A_n / \sqrt{\mathrm{Var}(S_n)} \leq \lim_n (2^n/n)^{1/3} n^3 / 2^{n/2} = \lim_n 2^{-n/6} n^{8/3} = 0.$$

We can conclude the central limit theorem:

Corollary 6.6. *With the definitions of this section,* $(S_n - \Sigma_n)/\sqrt{6\sigma^{n,1} + 2\sigma^{n,2}}$ *converges in distribution to* $\mathcal{N}(0,1)$.

In other, less formal, words, after many generations, the total time lag between the cells of this generation is approximately distributed as $\mathcal{N}(\Sigma_n, 6\sigma^{n,1} + 2\sigma^{n,2})$.

Chapter 7

Conclusion and outlook

In this Master's thesis, we have presented the theory of dependency graphs and related subjects, with a focus on concentration inequalities and central limit theorems. As it stands today, the theory already has important applications: the concentration inequalities are useful for statistical learning, for example in bipartite ranking and in learning from m-dependent data; the central limit theorems help us to understand the behaviour of large complex systems.

However, it has also become clear that we need more research, in at least two directions.

First, we have seen that in many cases the concentration inequalities of section 5.1 are not optimal, and we have conjectured other inequalities which are sharper in precisely these cases. Further research will have to prove or disprove these conjectures. Either way, this will improve our understanding of systems of partly dependent random variables, and presumably of statistical learning.

Another shortcoming of the current theory concerns the case where the dependency graph is only an approximation, that is, when it is really a thresholded dependency graph. We have seen that then the Hoeffding- and Bernstein-type inequalities for dependency graphs lose most of their power. We need further research to see if there are better concentration inequalities for thresholded dependency graphs. This may require alternative definitions of thresholded dependency graphs, for example with other dependence quantifiers.
It would also be interesting to look for central limit theorems in such contexts.

Bibliography

Berbee, H. C. P. (1979). *Random walks with stationary increments and renewal theory.*

Bollobás, B. (1990). *Graph theory : an introductory course.* eng. [Corr. 3rd print.] Graduate texts in mathematics 63. New York: Springer. ISBN: 0387903992.

Bradley, R. C. (1983). "Approximation theorems for strongly mixing random variables." In: *Michigan Mathematical Journal* 30, pp. 69–81.

– (2005). "Basic properties of strong mixing conditions. A survey and some open questions". In: *Probability Surveys* 2, pp. 107–144.

Dedecker, J. and C. Prieur (2005). "New dependence coefficients. Examples and applications to statistics". In: *Probability Theory and Related Fields* 132, pp. 203–236.

Erdös, P. and L. Lovász (1975). "Problems and Results on 3-chromatic Hypergraphs and Some Related Questions". In: *Colloquia Mathematica Societatis János Bolyai.*

Féray, V. (2016). "Weighted dependency graphs". In: *arXiv: Probability.*

Féray, V., P.-L. Méliot, and A. Nikeghbali (2016). "Dependency graphs and mod-Gaussian convergence". In: *Mod-Phi Convergence: Normality Zones and Precise Deviations.* Cham: Springer International Publishing, pp. 95–110. ISBN: 978-3-319-46822-8.

Filmus, Y. (2010). *Two Proofs of the Central Limit Theorem.*

Furrer, R. (2023). *Modeling Dependent Data: An Excursion.* Lecture notes. Zurich.

Hoeffding, W. (1963). *Probability inequalities for sums of bounded random variables.*

Isaev, M. et al. (2021). *Extremal independence in discrete random systems.*

Jacod, J. and P. Protter (2004). *Probability Essentials.* eng. 2nd ed. 2004. Universitext. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 3-642-55682-5.

Janson, S. (1988). "Normal Convergence by Higher Semiinvariants with Applications to Sums of Dependent Random Variables and Random Graphs". In: *Annals of Probability* 16, pp. 305–312.

– (2004). "Large deviations for sums of partly dependent random variables". In: *Random Structures & Algorithms* 24.

Janson, S., T. Łuczak, and A. Ruciński (2000). *Random graphs.* eng. Wiley-Interscience Series in Discrete Mathematics and Optimization. New York, New York: John Wiley & Sons, Inc. ISBN: 1-118-03271-3.

Klenke, A. (2020). *Wahrscheinlichkeitstheorie.* ger. 4., überarbeitete und ergänzte Auflage. Masterclass. Berlin: Springer Spektrum. ISBN: 9783662620885.

Lampert, C. H., L. Ralaivola, and A. Zimin (2018). "Dependency-dependent Bounds for Sums of Dependent Random Variables". In: *arXiv: Probability.*

Leonov, V. P. and A. N. Shiryaev (1959). "On a Method of Calculation of Semi-Invariants". In: *Theory of Probability and Its Applications* 4, pp. 319–329.

Mikhailov, V. G. (1992). "On a Theorem of Janson". In: *Theory of Probability and Its Applications* 36, pp. 173–176.

Paulin, D. (2014). *Concentration Inequalities for Dependent Random Variables*. PhD Thesis.

Peligrad, M. (2002). "Some remarks on coupling of dependent random variables". In: *Statistics & Probability Letters* 60, pp. 201–209.

Petersen, L. C. (1982). "On the relation between the multidimensional moment problem and the one-dimensional moment problem." In: *Mathematica Scandinavica* 51, pp. 361–366.

Petrovskaya, M. B. and A. M. Leontovich (1983). "The Central Limit Theorem for a Sequence of Random Variables with a Slowly Growing Number of Dependences". In: *Theory of Probability and its Applications*.

Rio, E. (2013). *Inequalities and limit theorems for weakly dependent sequences*.

Shalev-Shwartz, S. and S. Ben-David (2014). *Understanding machine learning : from theory to algorithms*. eng. Cambridge, United Kingdom: Cambridge University Press. ISBN: 1107057132.

Tanoue, Y. (2021). "Improved Hoeffding inequality for dependent bounded or sub-Gaussian random variables". In: *Probability, Uncertainty and Quantitative Risk*.

– (2022). "Concentration inequality of sums of dependent subexponential random variables and application to bounds for value-at-risk". In: *Communications in Statistics - Theory and Methods*.

Vershynin, R. (2018). *High-Dimensional Probability*.

Zhang, R. and M.-R. Amini (2022). "Generalization bounds for learning under graph-dependence: A survey". In: *ArXiv* abs/2203.13534.

Appendix A

Python code and Jupyter notebooks

A.1 Useful functions: depgraph.py

```
import numpy as np
import networkx as nx
import scipy.optimize as optimize
import matplotlib.pyplot as plt

# The following two functions will be used in test.ipynb to
    estimate subgaussian/subexponential norms.
def base(d, sample_size, rng, subgaussian=True):
    '''Generate the appropriate random variables
    Input:
    d: integer. Number of terms in the sum
    sample_size: integer, size of the sample
    rng: random number generator.
    subgaussian: boolean, default True.
    If True, the terms follow a standard normal distribution so
        that the result is subgaussian;
    if false, the terms follow a standard exponential
        distribution so that the result is subexponential.
    Output:
    1-dimensional numpy array of size sample_size.
    '''
    if subgaussian:
        base = np.abs(rng.standard_normal((d, sample_size)))
    else:
        base = rng.standard_exponential((d, sample_size))
    X = np.sum(base, axis=0) / d
    return X*rng.choice([-1,1], sample_size)

def mgf(X, t):
    '''Estimate the moment generating function of a sample.
    Input:
```

```
        X: 1-dimensional numpy array, representing a sample from a
            probability distribution.
        t: 1-dimensional numpy array. Range of t's for which to
            estimate E[exp(tX)]
        Output:
        1-dimensional numpy array of the same size as t,
            representing the empirical mgf of X at the points t.
        '''
    e = np.exp(np.outer(X, t))
    return np.sum(e, axis=0)/X.size

# The following functions will be used in depgraph.ipynb to
    assess optimality of concentration inequalities in the
    dependency graph setting.
def generate(G, sample_size, rng, subgaussian=True, weak=False,
    verbose=False):
    '''Generate subexponential or subgaussian random variables
        with a given dependency graph G.
    Input:
    G: networkx graph. G must have its nodes numbered from 0 to
        n-1.
    rng: random number generator.
    subgaussian: boolean, default True. If True, generate a
        subgaussian distribution; if False, generate a
        subexponential distribution.
    Output:
    numpy array
    '''
    n = G.number_of_nodes()
    m = G.number_of_edges()

    #m+n independent standard normal/exponential random
        variables, one for each edge and each node.
    #take their absolute value (not necessary in the exponential
        case)
    if subgaussian:
        base = np.abs(rng.standard_normal((m, sample_size)))
        X = np.abs(rng.standard_normal((n, sample_size)))
    else:
        base = rng.standard_exponential((m, sample_size))
        X = rng.standard_exponential((n, sample_size))
    #associate the base random variables to the edges
    index=0
    for (u,v) in G.edges():
        G[u][v]['base'] = base[index]
        index +=1

    if weak:
```

```
            # For each node, we compute a weighted sum of the base
                variables corresponding to its edges and the one
                corresponding to the node itself.
            for node in G.nodes():
                d = G.degree[node]
                if d >= 1:
                    X2 = 0
                    for (u,v) in G.edges(node):
                        X2 += G[u][v]['base']
                    X2 /= 2*d
                    X[node] = 0.5*X[node] + X2
        else:
            # For each node, we compute the sum of the base
                variables corresponding to its edges and the one
                corresponding to the node itself.
            for node in G.nodes():
                for (u,v) in G.edges(node):
                    X[node] += G[u][v]['base']
                #normalise
                X[node] = X[node]/(G.degree[node]+1)

        if verbose:
            print(X)

        #To ensure the expectation in zero, we multiply by
            independent Rademacher random variables
        X *= rng.choice([-1,1], (n, sample_size))
        if verbose:
            print(X)
        return X

    def bound(G, t, subgaussian=True):
        '''Compute the upper bound for P(|sum(X)|>t) guaranteed by
            the theorems,
         in the situation described in the text and implemented in
            the function generate().
         Input:
         G: networkx graph. The dependency graph.
         t: 1-dimensional numpy array. Range of t's for which to
            compute the upper bound. Typically an output from np.
            linspace() or similar.
         subgaussian: boolean, default True. If True, compute the
            bound for the subgaussian case (dependent Hoeffding). If
             False, compute the bound for the subexponential case (
            dependent Bernstein).
         Output:
         bound: 1-dimensional numpy array of the same size as t.
            Upper bound guaranteed by the theorem, with dependency
```

```
        graph G and for the specified t. If the bound is larger
        than 1, it is replaced by 1.
     constant1: negative float. The constant in the exponential:
         bound(t) = 2*exp(constant1*t**2). In the subexponential
         case, constant in the subgaussian regime (small t).
     '''
    degrees = np.array(list(nx.degree(G)))[:,1]
    Delta = np.max(degrees)
    if subgaussian:
        constant1 = -1/(2*(Delta+1)*G.number_of_nodes())
        bounds = 2*np.exp(constant1*t**2)
    else:
        constant1 = -(np.e-1)/(4*np.e*(Delta+1)*G.
            number_of_nodes())
        bounds1 = constant1*t**2
        bounds2 = -t*np.sqrt((np.e-1)/np.e)
        bounds = 2*np.exp(np.maximum(bounds1, bounds2))
    return np.minimum(bounds, 1), constant1

def estimate(X, t, credint=True, precision = 0.05, verbose=False
    ):
    '''Estimate P(|sum(X)|>t) for a random vector X and t in the
        specified array.
     Input:
     X: numpy array of shape (n, sample_size). Typically output
        from generate().
     t: numpy array of dimension 1. Array  of values x for which
         we will compute P(sum(X)>x)
     Output:
     p: estimated probabilities
     '''
    tsize = t.size

    S = np.abs(np.sum(X, axis=0).flatten()) #should be of shape
        sample_size
    sample_size = S.size

    #point estimate
    p_unnorm = np.empty(tsize)
    for index in range(tsize):
        p_unnorm[index] = np.count_nonzero(S>=t[index])
    p = p_unnorm / sample_size
    if verbose:
        print(p_unnorm)
        print("p:\n", p)

    if credint:
        epsilon = np.sqrt(np.log(2/precision)/(2*sample_size))
```

```
        p_low = np.maximum(0, p-epsilon)
        p_high = np.minimum(1, p+epsilon)
        return p, p_low, p_high
    else:
        return p

def draw(t, p, bounds, credint=False, p_low=None, p_high=None,
    conjecture=np.asarray(None), name='', log=False):
    fig, ax= plt.subplots(figsize=(16,10))
    ax.set_xlabel("t")
    ax.set_ylabel("p")
    if log:
        ax.set_yscale('log')
    if credint:
        ax.fill_between(t, p_low, p_high, color="grey", label="
            credible interval assuming U[0,1] prior")
    ax.plot(t, p, color="black", label="estimated P(|sum(X_k)|>=
        t)")
    ax.plot(t, bounds, color="red", label="upper bound
        guaranteed by theorem")
    if (conjecture!=None).any():
        ax.plot(t, conjecture, color="red", linestyle="dotted",
            label="conjectured bound")
    ax.legend()
    if name:
        ax.set_title(name)
        fig.savefig(f'{name}.eps')
    plt.show()
    return

def f(t, a, b, c):
    return a*t**2 + b*t + c

def conjecture(t, a, b, c):
    return np.exp(f(t,a,b,c))

def fit(t, p):
    t = t[p>0]
    p = p[p>0]
    if t.size <= 3:
        raise Exception("Too few values for a meaningful
            quadratic curve fit")
    return optimize.curve_fit(f, t, np.log(p))
```

A.2 Testing subgaussian and subexponential norms: test.ipynb

[1]:
```
import depgraph
```

[2]:
```
import numpy as np
```

[3]:
```
t = np.linspace(0.01, 2)
```

[4]:
```
rng = np.random.default_rng(2512801)
```

Subgaussian case

[5]:
```
for d in range(6):
    print('======')
    print(d+1)
    X = depgraph.base(d+1, 1000000, rng)
    estimate = depgraph.mgf(X, t)
    #print(estimate)
    rate = np.log(estimate)/t**2
    print(rate)
    bound = np.max(rate)
    print(bound)
```

```
======
1
[0.48745126 0.49769816 0.49885516 0.4993219  0.49958553 0.49976186
 0.49989255 0.49999632 0.50008285 0.50015763 0.50022406 0.50028433
 0.50033994 0.50039194 0.5004411  0.500488   0.50053307 0.50057666
 0.50061903 0.50066041 0.50070096 0.50074083 0.50078012 0.50081891
 0.5008573  0.50089532 0.50093301 0.50097041 0.50100754 0.50104441
 0.50108101 0.50111734 0.5011534  0.50118916 0.50122461 0.50125973
 0.50129449 0.50132886 0.50136283 0.50139636 0.50142944 0.50146205
 0.50149416 0.50152578 0.50155689 0.5015875  0.50161762 0.50164728
 0.5016765  0.50170533]
0.5017053298998431
======
2
[0.61283499 0.44891182 0.43077518 0.42361864 0.41962579 0.41694071
 0.41489691 0.41319676 0.411688   0.4102852  0.40893726 0.40761221
 0.40628947 0.40495556 0.40360169 0.4022222  0.40081363 0.39937409
 0.39790283 0.39639994 0.39486612 0.39330254 0.39171072 0.39009244
 0.38844963 0.3867844  0.38509889 0.38339532 0.38167592 0.3799429
 0.37819847 0.37644477 0.37468392 0.37291796 0.37114884 0.36937847
 0.36760865 0.36584111 0.36407749 0.36231934 0.36056812 0.3588252
```

```
 0.35709188 0.35536934 0.35365871 0.35196102 0.35027722 0.3486082
 0.34695474 0.3453176 ]
0.6128349935082357
======
3
[0.42878038 0.38942618 0.38493633 0.38299897 0.3817413  0.38071945
 0.37977489 0.37883774 0.3778732  0.37686267 0.37579599 0.37466773
 0.37347536 0.37221819 0.37089676 0.36951247 0.36806732 0.36656373
 0.36500447 0.3633925  0.36173095 0.36002306 0.35827213 0.35648147
 0.3546544  0.35279423 0.35090418 0.34898746 0.34704716 0.34508629
 0.34310778 0.34111443 0.33910896 0.33709393 0.33507183 0.33304499
 0.33101564 0.32898589 0.32695771 0.32493298 0.32291344 0.32090073
 0.31889637 0.31690178 0.31491827 0.31294705 0.31098925 0.30904589
 0.30711791 0.30520616]
0.42878037568102717
======
4
[0.33859551 0.35846828 0.36047598 0.36103239 0.3610902  0.36087676
 0.36047497 0.35992315 0.35924222 0.35844529 0.35754153 0.35653806
 0.35544081 0.35425507 0.35298571 0.35163735 0.35021449 0.3487215
 0.34716269 0.34554229 0.34386448 0.34213337 0.34035301 0.33852736
 0.33666029 0.33475559 0.33281693 0.33084787 0.32885187 0.32683225
 0.3247922  0.32273478 0.32066293 0.31857943 0.31648693 0.31438796
 0.3122849  0.31017998 0.30807534 0.30597294 0.30387466 0.30178222
 0.29969723 0.29762121 0.29555554 0.2935015  0.29146028 0.28943297
 0.28742054 0.28542391]
0.36109019576289497
======
5
[0.38506534 0.36060155 0.35773666 0.35641162 0.35546418 0.35461936
 0.35377943 0.35290186 0.35196596 0.35096129 0.34988269 0.3487281
 0.34749728 0.34619127 0.34481188 0.34336156 0.34184312 0.34025973
 0.33861474 0.33691164 0.33515406 0.33334566 0.33149013 0.32959115
 0.32765236 0.32567738 0.32366973 0.32163286 0.31957012 0.31748475
 0.31537988 0.31325852 0.31112354 0.30897771 0.30682364 0.30466382
 0.3025006  0.3003362  0.29817272 0.29601212 0.29385623 0.29170678
 0.28956536 0.28743346 0.28531244 0.28320358 0.28110804 0.27902689
 0.2769611  0.27491157]
0.38506534450982594
======
6
[0.28736493 0.33656847 0.34178511 0.34356567 0.34426256 0.34443567
 0.34428754 0.34391013 0.34335205 0.34264214 0.34179926 0.34083676
 0.33976487 0.33859189 0.33732489 0.33597021 0.3345336  0.33302049
 0.33143601 0.3297851  0.3280725  0.32630284 0.32448057 0.32261004
 0.32069543 0.31874083 0.31675017 0.31472724 0.3126757  0.31059905
 0.30850066 0.30638374 0.30425135 0.30210641 0.29995169 0.2977898
 0.29562322 0.29345426 0.29128512 0.28911786 0.28695437 0.28479646
```

```
 0.28264578 0.28050387 0.27837217 0.27625197 0.2741445  0.27205087
 0.26997207 0.26790905]
0.34443567448635515
```

Subexponential case

[6]:
```
c2 = np.e / (np.e-1)
print(c2)
```

```
1.5819767068693265
```

[7]:
```
t_2 = np.linspace(0.01, 1/np.sqrt(c2))
```

[8]:
```
for d in range(6):
    print('======')
    print(d+1)
    X = depgraph.base(d+1, 1000000, rng, subgaussian=False)
    estimate = depgraph.mgf(X, t_2)
    #print(estimate)
    rate = np.log(estimate)/t_2**2
    print(rate)
    bound = np.max(rate)
    print(bound)
```

```
======
1
[0.96592554 0.98649684 0.99192307 0.99495653 0.99732009 0.99951312
 1.00173545 1.00408418 1.00661365 1.00935796 1.01234087 1.01558056
 1.01909214 1.02288914 1.02698437 1.03139046 1.0361203  1.0411873
 1.04660564 1.05239048 1.05855813 1.06512627 1.07211408 1.07954253
 1.08743449 1.09581505 1.10471175 1.11415484 1.12417766 1.13481695
 1.14611325 1.15811137 1.17086084 1.18441644 1.19883877 1.21419488
 1.23055884 1.24801246 1.2666459  1.28655827 1.30785818 1.33066417
 1.35510483 1.38131882 1.40945433 1.43966809 1.47212376 1.50698941
 1.54443422 1.58462392]
1.5846239204393118
======
2
[1.04128041 0.86267925 0.82022647 0.80123486 0.79049428 0.78361236
 0.77884916 0.77537767 0.77275455 0.77072082 0.76911509 0.7678316
 0.76679821 0.76596405 0.76529226 0.76475555 0.76433325 0.76400954
 0.76377205 0.76361108 0.76351889 0.7634893  0.76351731 0.76359889
 0.76373077 0.76391029 0.76413531 0.7644041  0.7647153  0.76506782
 0.76546085 0.76589376 0.76636614 0.76687772 0.76742839 0.76801814
 0.7686471  0.76931548 0.77002358 0.77077179 0.77156057 0.77239045
 0.77326201 0.7741759  0.77513282 0.7761335  0.77717876 0.7782694
 0.77940631 0.7805904 ]
```

```
1.041280406088624
======
3
[0.65580898 0.66216153 0.66363944 0.6642619  0.66457146 0.66472484
 0.66478424 0.66477926 0.66472587 0.66463348 0.66450805 0.66435355
 0.66417279 0.66396784 0.66374029 0.66349141 0.66322224 0.66293367
 0.66262649 0.6623014  0.66195908 0.66160012 0.66122511 0.66083464
 0.66042924 0.66000947 0.65957587 0.65912898 0.65866933 0.65819746
 0.65771391 0.65721923 0.65671395 0.65619862 0.65567379 0.65514001
 0.65459783 0.65404782 0.65349052 0.65292651 0.65235633 0.65178056
 0.65119976 0.65061448 0.65002531 0.6494328  0.64883751 0.64824001
 0.64764086 0.64704063]
0.6647842362394143
======
4
[0.54563534 0.59477583 0.60640615 0.61155153 0.61439976 0.61616028
 0.6173123  0.61808371 0.6185968  0.61892313 0.61910742 0.61917911
 0.61915841 0.61905972 0.61889357 0.61866793 0.61838895 0.61806147
 0.61768942 0.61727601 0.61682392 0.61633547 0.61581262 0.61525713
 0.61467056 0.6140543  0.61340965 0.61273778 0.61203982 0.6113168
 0.61056972 0.60979951 0.6090071  0.60819336 0.60735913 0.60650526
 0.60563254 0.60474177 0.60383372 0.60290916 0.60196883 0.60101348
 0.60004382 0.59906058 0.59806445 0.59705614 0.59603633 0.59500569
 0.5939649  0.59291462]
0.6191791097989585
======
5
[0.3693761  0.51083046 0.54441531 0.55939401 0.56781518 0.57315833
 0.57680195 0.57940141 0.58130824 0.58272795 0.58378879 0.58457499
 0.58514421 0.58553729 0.58578402 0.58590665 0.58592223 0.58584401
 0.58568254 0.5854463  0.58514226 0.58477619 0.58435294 0.58387667
 0.58335093 0.58277885 0.58216316 0.58150631 0.58081048 0.58007767
 0.5793097  0.57850824 0.57767485 0.57681097 0.57591797 0.57499714
 0.5740497  0.57307682 0.57207961 0.57105914 0.57001645 0.56895252
 0.56786833 0.5667648  0.56564285 0.56450334 0.56334715 0.56217511
 0.56098803 0.55978671]
0.5859222287303083
======
6
[0.47886134 0.54360774 0.55894194 0.56573542 0.56950498 0.57184387
 0.57338337 0.57442366 0.5751258  0.57558405 0.57585727 0.57598421
 0.57599144 0.57589788 0.57571736 0.57546033 0.5751348  0.57474713
 0.57430239 0.57380476 0.57325772 0.5726642  0.57202676 0.5713476
 0.5706287  0.56987181 0.56907854 0.56825037 0.56738866 0.56649469
 0.56556966 0.56461471 0.56363095 0.56261942 0.56158114 0.56051708
 0.55942819 0.55831541 0.55717964 0.55602177 0.55484266 0.55364315
 0.5524241  0.55118631 0.54993058 0.54865772 0.54736849 0.54606366
 0.54474398 0.54341019]
```

```
0.5759914429796903
```

A.3 Simulating concentration inequalities: depgraph.ipynb

Simulations

[1]:
```
import numpy as np
import networkx as nx
```

[2]:
```
import depgraph
```

[3]:
```
rng = np.random.default_rng(2512800)
```

[4]:
```
t = np.linspace(0, 40)
```

Complete dependency graph

[5]:
```
G_complete = nx.complete_graph(50)
```

[6]:
```
X_subgaussian_complete = depgraph.generate(G_complete, 100000, rng)
```

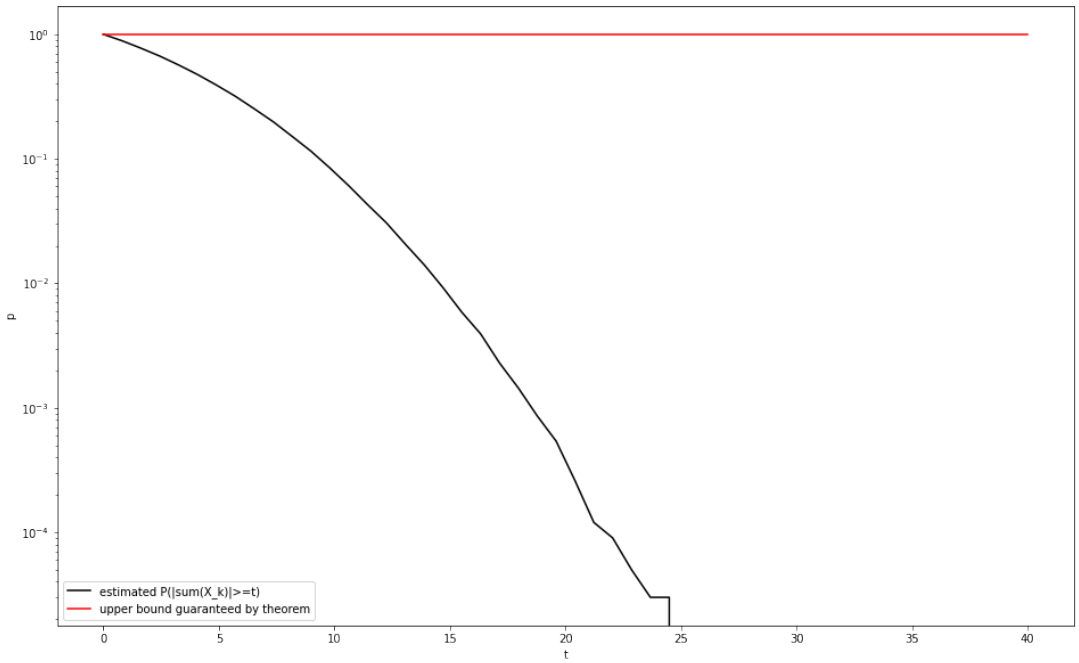
[7]:
```
p_subgaussian_complete = depgraph.estimate(X_subgaussian_complete, t,␣
 ↪credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subgaussian_complete)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 8.8634e-01 7.7275e-01 6.6749e-01 5.6599e-01 4.7453e-01
 3.9051e-01 3.1757e-01 2.5205e-01 1.9861e-01 1.5161e-01 1.1530e-01
 8.4550e-02 6.0900e-02 4.2920e-02 3.0610e-02 2.0680e-02 1.4120e-02
 9.2400e-03 5.8600e-03 3.9200e-03 2.2900e-03 1.4400e-03 8.6000e-04
 5.4000e-04 2.6000e-04 1.2000e-04 9.0000e-05 5.0000e-05 3.0000e-05
 3.0000e-05 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[8]:
```
bounds_subgaussian_complete, constant_subgaussian_complete = depgraph.
 ↪bound(G_complete, t)
print('bounds guaranteed by theorem:')
print(bounds_subgaussian_complete)
print('constant in the exponential:', constant_subgaussian_complete)
```

```
bounds guaranteed by theorem:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1.]
constant in the exponential: -0.0002
```

[9]:
```
depgraph.draw(t, p_subgaussian_complete, bounds_subgaussian_complete,␣
 ↪log=True)
```



[10]:
```
X_subexp_complete = depgraph.generate(G_complete, 100000, rng,␣
 ↪subgaussian=False)
```

[11]:
```
p_subexp_complete = depgraph.estimate(X_subexp_complete, t,␣
 ↪credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subexp_complete)
```

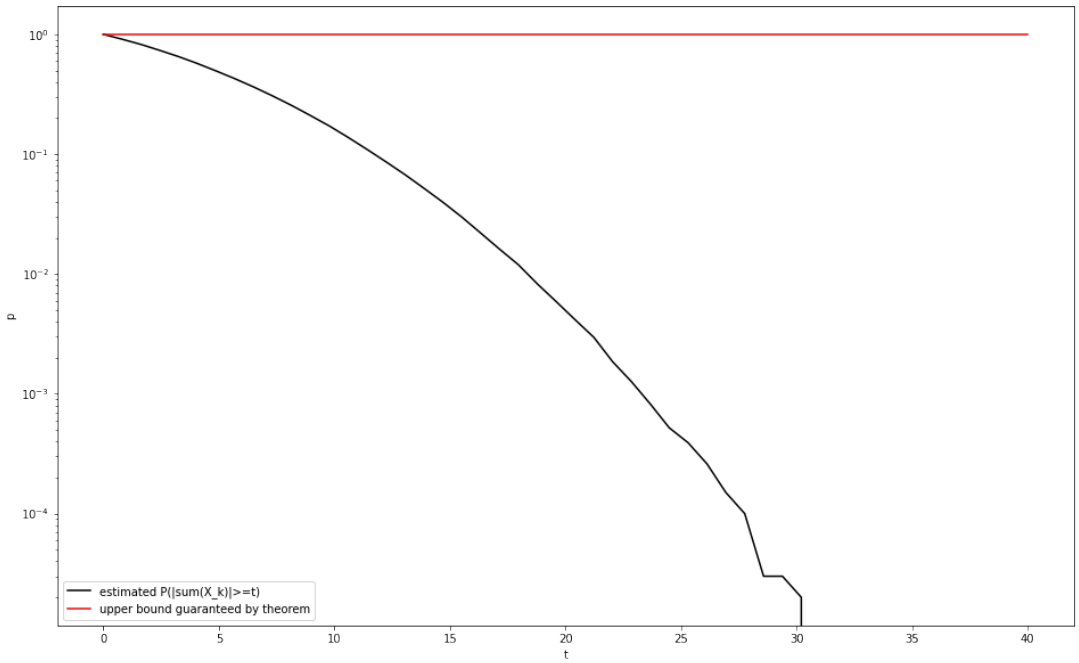
```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 9.1003e-01 8.2299e-01 7.3381e-01 6.5054e-01 5.6940e-01
 4.9300e-01 4.2434e-01 3.6159e-01 3.0458e-01 2.5405e-01 2.0906e-01
 1.7108e-01 1.3724e-01 1.0896e-01 8.5910e-02 6.7280e-02 5.1810e-02
 3.9690e-02 2.9860e-02 2.1960e-02 1.6120e-02 1.1940e-02 8.2800e-03
 5.8900e-03 4.1600e-03 2.9600e-03 1.8500e-03 1.2600e-03 8.2000e-04
 5.2000e-04 3.9000e-04 2.6000e-04 1.5000e-04 1.0000e-04 3.0000e-05
 3.0000e-05 2.0000e-05 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
```

```
 0.0000e+00 0.0000e+00]
```

[12]:
```
bounds_exp_complete, constant1_exp_complete = depgraph.
 ↪bound(G_complete, t, subgaussian=False)
print('bounds guaranteed by theorem:')
print(bounds_exp_complete)
print('constant in the squared exponential:', constant1_exp_complete)
```

```
bounds guaranteed by theorem:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1.]
constant in the squared exponential: -6.321205588285576e-05
```

[13]:
```
depgraph.draw(t, p_subexp_complete, bounds_exp_complete, log=True)
```



A dense dependency graph

[14]:
```
G_dense = nx.erdos_renyi_graph(50, 0.5, seed = rng)
Delta = max(G_dense.degree())[1]
print(Delta)
```

```
24
```

[15]:
```
X_subgaussian_dense = depgraph.generate(G_dense, 100000, rng)
```

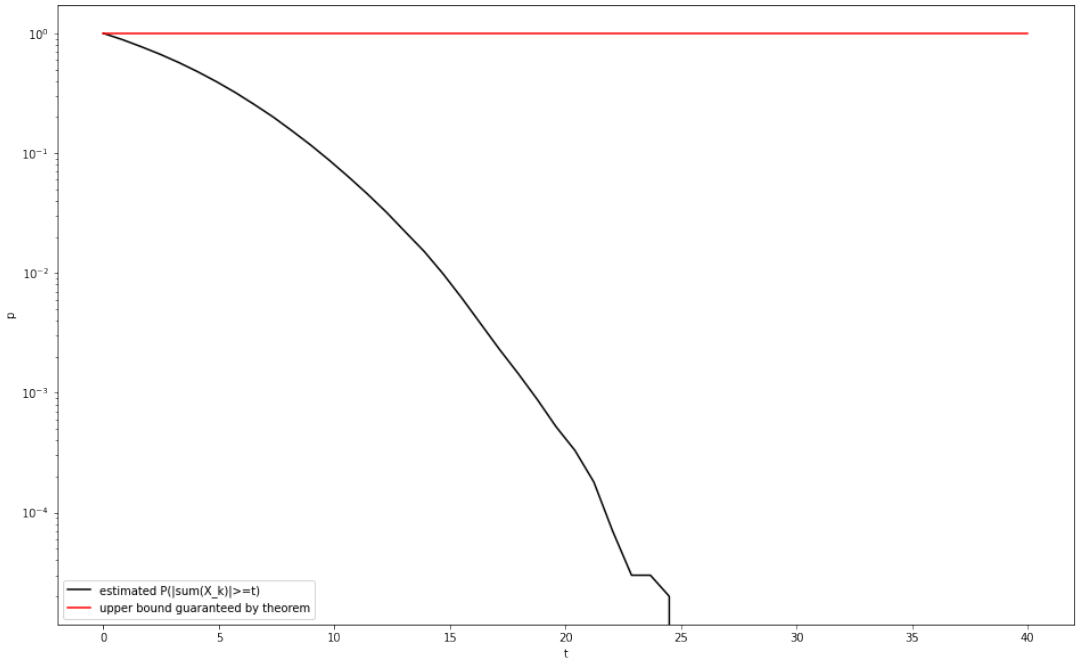
[16]:
```
p_subgaussian_dense = depgraph.estimate(X_subgaussian_dense, t,␣
 ↪credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subgaussian_dense)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 8.8704e-01 7.7408e-01 6.6956e-01 5.7032e-01 4.7817e-01
 3.9449e-01 3.2009e-01 2.5467e-01 2.0030e-01 1.5359e-01 1.1651e-01
 8.6570e-02 6.3280e-02 4.5560e-02 3.2180e-02 2.2050e-02 1.5170e-02
 9.8700e-03 6.1800e-03 3.7700e-03 2.3000e-03 1.4400e-03 8.8000e-04
 5.2000e-04 3.3000e-04 1.8000e-04 7.0000e-05 3.0000e-05 3.0000e-05
 2.0000e-05 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[17]:
```
bounds_subgaussian_dense, constant_subgaussian_dense = depgraph.
 ↪bound(G_dense, t)
print('bounds guaranteed by theorem:')
print(bounds_subgaussian_dense)
print('constant in the exponential:', constant_subgaussian_dense)
```

```
bounds guaranteed by theorem:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1.]
constant in the exponential: -0.0003333333333333333
```

[18]:
```
depgraph.draw(t, p_subgaussian_dense, bounds_subgaussian_dense,␣
 ↪log=True)
```



[19]:
```
X_subexp_dense = depgraph.generate(G_dense, 100000, rng,␣
↪subgaussian=False)
```

[20]:
```
p_subexp_dense = depgraph.estimate(X_subexp_dense, t, credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subexp_dense)
```

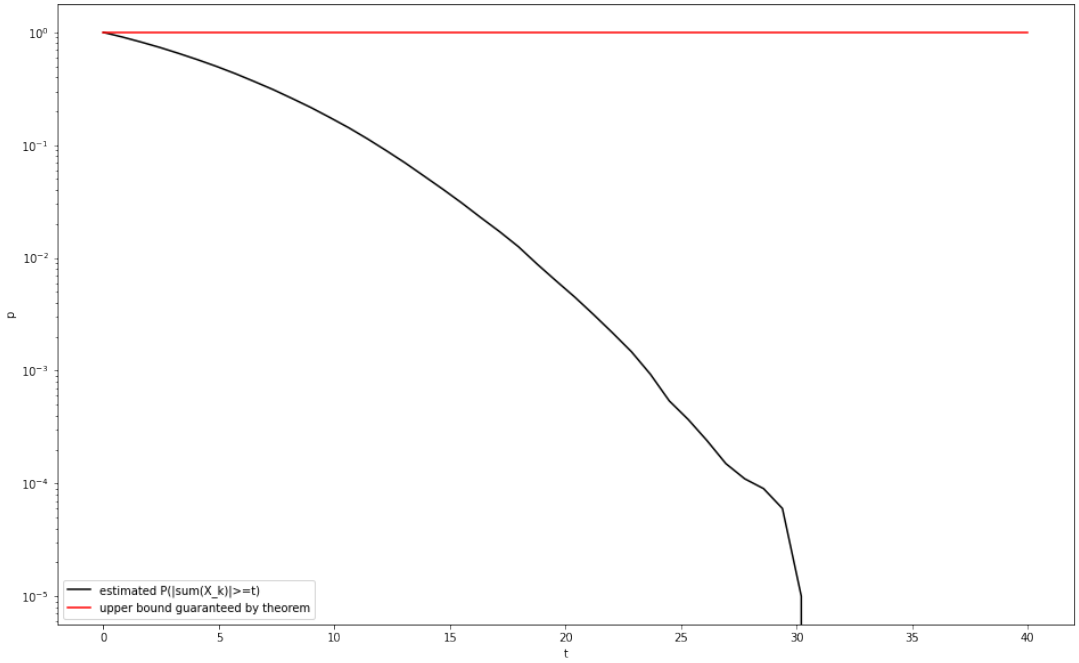
```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 9.1034e-01 8.2071e-01 7.3475e-01 6.5052e-01 5.7213e-01
 4.9905e-01 4.3036e-01 3.6670e-01 3.1092e-01 2.5965e-01 2.1568e-01
 1.7614e-01 1.4289e-01 1.1378e-01 8.9340e-02 6.9530e-02 5.3190e-02
 4.0630e-02 3.0700e-02 2.2840e-02 1.7130e-02 1.2590e-02 8.8000e-03
 6.2600e-03 4.4800e-03 3.1300e-03 2.1600e-03 1.4700e-03 9.3000e-04
 5.4000e-04 3.7000e-04 2.4000e-04 1.5000e-04 1.1000e-04 9.0000e-05
 6.0000e-05 1.0000e-05 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[21]:
```
bounds_exp_dense, constant1_exp_dense = depgraph.bound(G_dense, t,␣
↪subgaussian=False)
print('bounds guaranteed by theorem:')
print(bounds_exp_dense)
print('constant in the (squared) exponential:', constant1_exp_dense)
```

```
bounds guaranteed by theorem:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
```

```
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1.]
constant in the (squared) exponential: -0.00010535342647142628
```

[22]:
```
depgraph.draw(t, p_subexp_dense, bounds_exp_dense, log=True)
```



A sparse dependency graph

[23]:
```
G_sparse = nx.erdos_renyi_graph(50, 0.1, seed = rng)
Delta= max(G_sparse.degree())[1]
print(Delta)
```

```
5
```

[24]:
```
X_subgaussian_sparse = depgraph.generate(G_sparse, 100000, rng)
```

[25]:
```
p_subgaussian_sparse = depgraph.estimate(X_subgaussian_sparse, t,␣
 ↪credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subgaussian_sparse)
```

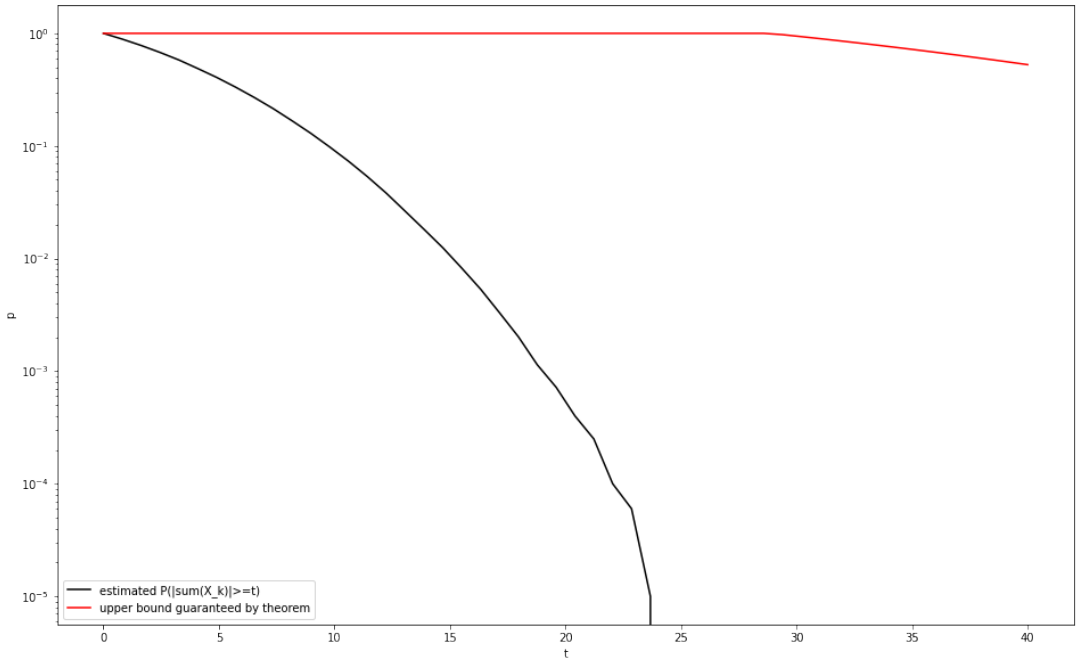
```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 8.8830e-01 7.8043e-01 6.7687e-01 5.8022e-01 4.8709e-01
 4.0648e-01 3.3332e-01 2.6936e-01 2.1452e-01 1.6731e-01 1.2926e-01
 9.7820e-02 7.3040e-02 5.3310e-02 3.8010e-02 2.6300e-02 1.8150e-02
```

```
 1.2440e-02 8.2100e-03 5.3400e-03 3.3000e-03 2.0200e-03 1.1400e-03
 7.2000e-04 4.0000e-04 2.5000e-04 1.0000e-04 6.0000e-05 1.0000e-05
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[26]:
```
bounds_subgaussian_sparse, constant_subgaussian_sparse = depgraph.
 ↪bound(G_sparse, t)
print('bounds guaranteed by theorem:')
print(bounds_subgaussian_sparse)
print('constant in the exponential:', constant_subgaussian_sparse)
```

```
bounds guaranteed by theorem:
[1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 0.97379649 0.93510953 0.89696277 0.85941712 0.82252904 0.78635043
 0.75092864 0.71630644 0.68252207 0.64960924 0.61759723 0.58651097
 0.55637113 0.52719428]
constant in the exponential: -0.0008333333333333334
```

[27]:
```
depgraph.draw(t, p_subgaussian_sparse, bounds_subgaussian_sparse,␣
 ↪log=True)
```



[28]:
```
X_subexp_sparse = depgraph.generate(G_sparse, 100000, rng,␣
↪subgaussian=False)
```

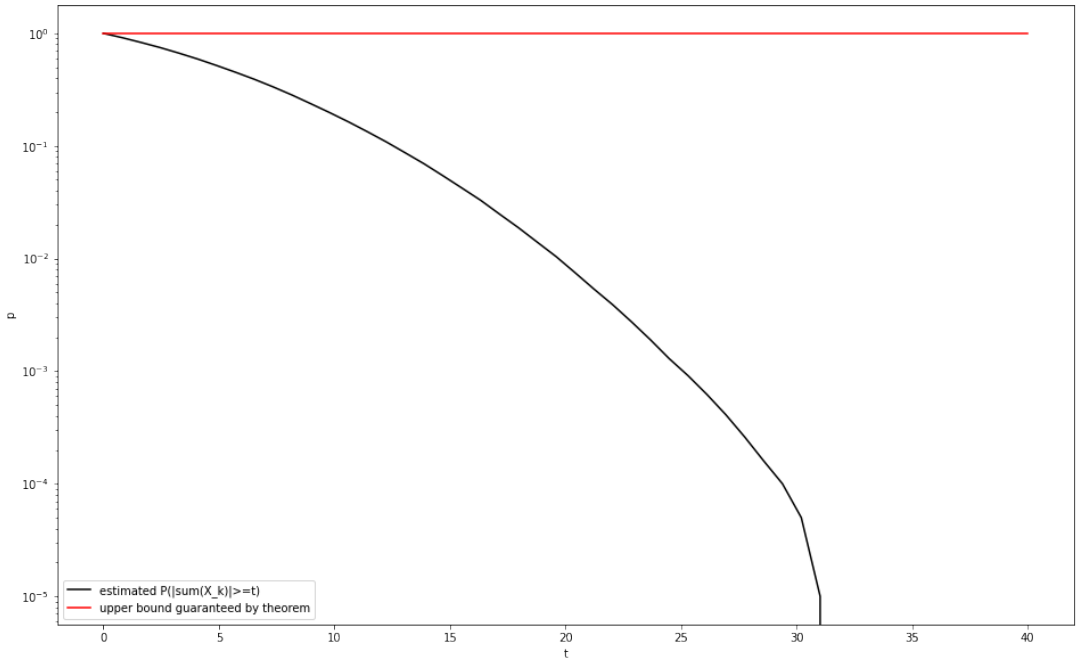
[29]:
```
p_subexp_sparse = depgraph.estimate(X_subexp_sparse, t, credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subexp_sparse)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 9.1504e-01 8.2959e-01 7.4819e-01 6.6738e-01 5.9124e-01
 5.1808e-01 4.5156e-01 3.9094e-01 3.3444e-01 2.8317e-01 2.3688e-01
 1.9754e-01 1.6337e-01 1.3362e-01 1.0854e-01 8.6920e-02 6.9420e-02
 5.4360e-02 4.2350e-02 3.2910e-02 2.4820e-02 1.8810e-02 1.3980e-02
 1.0410e-02 7.5000e-03 5.3700e-03 3.8900e-03 2.7400e-03 1.9000e-03
 1.2900e-03 9.1000e-04 6.2000e-04 4.1000e-04 2.6000e-04 1.6000e-04
 1.0000e-04 5.0000e-05 1.0000e-05 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[30]:
```
bounds_exp_sparse, constant1_exp_sparse = depgraph.bound(G_sparse, t,␣
↪subgaussian=False)
print('bounds guaranteed by theorem:')
print(bounds_exp_sparse)
print('constant in the (squared) exponential:', constant1_exp_sparse)
```

```
bounds guaranteed by theorem:
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1.]
constant in the (squared) exponential: -0.0002633835661785657
```

[31]:
```
depgraph.draw(t, p_subexp_sparse, bounds_exp_sparse, log=True)
```



How do the tails actually behave?

[32]:
```
fit_subgaussian_complete, cov_subgaussian_complete = depgraph.fit(t,␣
 ↪p_subgaussian_complete)
print('approximate tail in the subgaussian case with complete␣
 ↪dependency graph:')
print(f'exp({fit_subgaussian_complete[0]} t^2 +␣
 ↪{fit_subgaussian_complete[1]} t + {fit_subgaussian_complete[2]})')
```

```
approximate tail in the subgaussian case with complete dependency graph:
exp(-0.013001295483448905 t^2 + -0.13147416015634486 t + 0.
 ↪021113432547213)
```

[33]:
```
ratio_subgaussian_complete = fit_subgaussian_complete[0] /␣
 ↪constant_subgaussian_complete
print(ratio_subgaussian_complete)
```

```
65.00647741724453
```

[34]:
```
fit_subexp_complete, cov_subexp_complete = depgraph.fit(t,␣
 ↪p_subexp_complete)
print('approximate tail in the subexponential case with complete␣
 ↪dependency graph:')
print(f'exp({fit_subexp_complete[0]} t^2 + {fit_subexp_complete[1]} t +␣
 ↪{fit_subexp_complete[2]})')
```

```
approximate tail in the subexponential case with complete dependency␣
 ↪graph:
exp(-0.009248033948269898 t^2 + -0.07824519000805567 t + -0.
 ↪07672376204039545)
```

[35]:
```
ratio_subexp_complete = fit_subexp_complete[0] / constant1_exp_complete
print(ratio_subexp_complete)
```

```
146.3017429049975
```

[36]:
```
fit_subgaussian_dense, cov_subgaussian_dense = depgraph.fit(t,␣
 ↪p_subgaussian_dense)
print('approximate tail in the subgaussian case with dense dependency␣
 ↪graph:')
print(f'exp({fit_subgaussian_dense[0]} t^2 + {fit_subgaussian_dense[1]}␣
 ↪t + {fit_subgaussian_dense[2]})')
```

```
approximate tail in the subgaussian case with dense dependency graph:
exp(-0.01423987204212375 t^2 + -0.10579626722843102 t + -0.
 ↪04367846025891317)
```

[37]:
```
ratio_subgaussian_dense = fit_subgaussian_dense[0] /␣
 ↪constant_subgaussian_dense
print(ratio_subgaussian_dense)
```

```
42.71961612637125
```

[38]:
```
fit_subexp_dense, cov_subexp_dense = depgraph.fit(t, p_subexp_dense)
print('approximate tail in the subexponential case with dense␣
 ↪dependency graph:')
print(f'exp({fit_subexp_dense[0]} t^2 + {fit_subexp_dense[1]} t +␣
 ↪{fit_subexp_dense[2]})')
```

```
approximate tail in the subexponential case with dense dependency graph:
exp(-0.009025759283523163 t^2 + -0.07915956876697998 t + -0.
 ↪07040671482131548)
```

[39]:
```
ratio_subexp_dense = fit_subexp_dense[0] / constant1_exp_dense
print(ratio_subexp_dense)
```

```
85.67124569005935
```

[40]:
```
fit_subgaussian_sparse, cov_subgaussian_sparse = depgraph.fit(t,␣
 ↪p_subgaussian_sparse)
```

```
print('approximate tail in the subgaussian case with sparse dependency␣
 ↪graph:')
print(f'exp({fit_subgaussian_sparse[0]} t^2 +␣
 ↪{fit_subgaussian_sparse[1]} t + {fit_subgaussian_sparse[2]})')
```

```
approximate tail in the subgaussian case with sparse dependency graph:
exp(-0.016703909924465875 t^2 + -0.042065797497196446 t + -0.
 ↪20813577916736184)
```

[41]:
```
ratio_subgaussian_sparse = fit_subgaussian_sparse[0] /␣
 ↪constant_subgaussian_sparse
print(ratio_subgaussian_sparse)
```

```
20.04469190935905
```

[42]:
```
fit_subexp_sparse, cov_subexp_sparse = depgraph.fit(t, p_subexp_sparse)
print('approximate tail in the subexponential case with sparse␣
 ↪dependency graph:')
print(f'exp({fit_subexp_sparse[0]} t^2 + {fit_subexp_sparse[1]} t +␣
 ↪{fit_subexp_sparse[2]})')
```

```
approximate tail in the subexponential case with sparse dependency graph:
exp(-0.00902642469097084 t^2 + -0.04895164472412505 t + -0.
 ↪17280584246389313)
```

[43]:
```
ratio_subexp_sparse = fit_subexp_sparse[0] / constant1_exp_sparse
print(ratio_subexp_sparse)
```

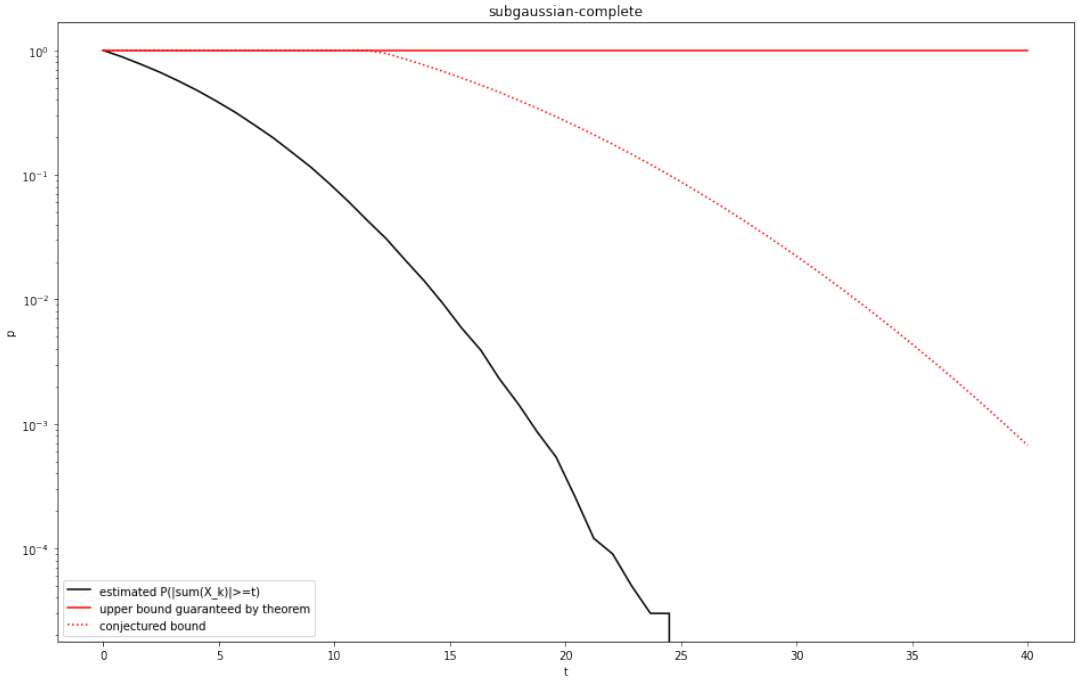
```
34.271024657822466
```

Conjectures

[44]:
```
conjecture_subgaussian = np.minimum(1, 2*np.exp(-t**2/200))
```

[45]:
```
depgraph.draw(t, p_subgaussian_complete, bounds_subgaussian_complete,␣
 ↪conjecture=conjecture_subgaussian, log=True,␣
 ↪name='subgaussian-complete')
```

```
The PostScript backend does not support transparency; partially␣
 ↪transparent
artists will be rendered opaque.
```



Overview plot

[46]:
```
import matplotlib.pyplot as plt
```

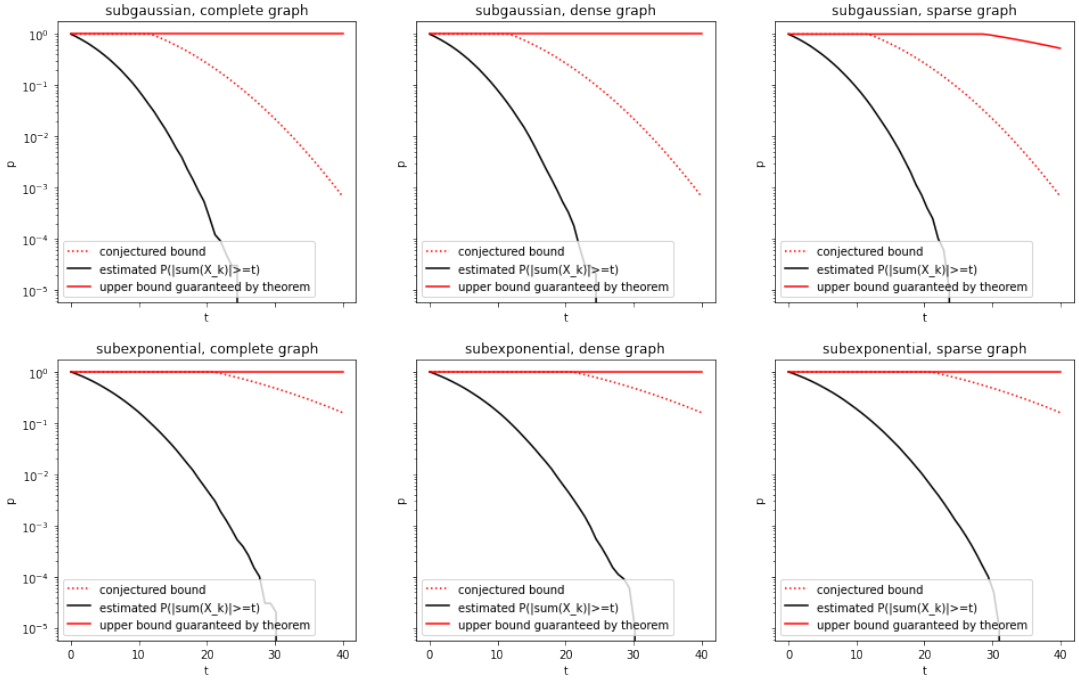
[47]:
```
conjecture_subexp = np.minimum(1, 2*np.exp(-np.minimum(t**2*(np.e-1)/
 ↪(400*np.e), t*np.sqrt((np.e-1)/np.e))))
```

[48]:
```
fig, axs = plt.subplots(2, 3, sharex=True, sharey=True, figsize=(16,10))


for index in [(0,0), (0,1), (0,2), (1,0), (1,1), (1,2)]:
    axs[index].set_xlabel("t")
    axs[index].set_ylabel("p")
    axs[index].set_yscale("log")
    if index[0]==0:
        axs[index].plot(t, conjecture_subgaussian, color="red",␣
 ↪linestyle="dotted", label="conjectured bound")
    else:
        axs[index].plot(t, conjecture_subexp, color="red",␣
 ↪linestyle="dotted", label="conjectured bound")

axs[0,0].set_title("subgaussian, complete graph")
axs[0,0].plot(t, p_subgaussian_complete, color="black",␣
 ↪label="estimated P(|sum(X_k)|>=t)")
```

```
axs[0,0].plot(t, bounds_subgaussian_complete, color="red", label="upper␣
 ↪bound guaranteed by theorem")

axs[0,1].set_title("subgaussian, dense graph")
axs[0,1].plot(t, p_subgaussian_dense, color="black", label="estimated␣
 ↪P(|sum(X_k)|>=t)")
axs[0,1].plot(t, bounds_subgaussian_dense, color="red", label="upper␣
 ↪bound guaranteed by theorem")

axs[0,2].set_title("subgaussian, sparse graph")
axs[0,2].plot(t, p_subgaussian_sparse, color="black", label="estimated␣
 ↪P(|sum(X_k)|>=t)")
axs[0,2].plot(t, bounds_subgaussian_sparse, color="red", label="upper␣
 ↪bound guaranteed by theorem")

axs[1,0].set_title("subexponential, complete graph")
axs[1,0].plot(t, p_subexp_complete, color="black", label="estimated␣
 ↪P(|sum(X_k)|>=t)")
axs[1,0].plot(t, bounds_exp_complete, color="red", label="upper bound␣
 ↪guaranteed by theorem")

axs[1,1].set_title("subexponential, dense graph")
axs[1,1].plot(t, p_subexp_dense, color="black", label="estimated␣
 ↪P(|sum(X_k)|>=t)")
axs[1,1].plot(t, bounds_exp_dense, color="red", label="upper bound␣
 ↪guaranteed by theorem")

axs[1,2].set_title("subexponential, sparse graph")
axs[1,2].plot(t, p_subexp_sparse, color="black", label="estimated␣
 ↪P(|sum(X_k)|>=t)")
axs[1,2].plot(t, bounds_exp_sparse, color="red", label="upper bound␣
 ↪guaranteed by theorem")

for index in [(0,0), (0,1), (0,2), (1,0), (1,1), (1,2)]:
    axs[index].legend()

fig.savefig('overview.eps')
fig.show()
```

```
The PostScript backend does not support transparency; partially␣
 ↪transparent
artists will be rendered opaque.
/tmp/ipykernel_8461/3329640641.py:41: UserWarning: Matplotlib is␣
 ↪currently using
module://matplotlib_inline.backend_inline, which is a non-GUI backend,␣
 ↪so cannot
show the figure.
```

```
fig.show()
```



What if the dependencies are weaker?

[49]:
```
X_subgaussian_complete_weak = depgraph.generate(G_complete, 100000,␣
↪rng, weak=True)
```

[50]:
```
p_subgaussian_complete_weak = depgraph.
↪estimate(X_subgaussian_complete_weak, t, credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subgaussian_complete_weak)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 8.9280e-01 7.8728e-01 6.8573e-01 5.9001e-01 4.9872e-01
 4.1755e-01 3.4369e-01 2.8043e-01 2.2404e-01 1.7648e-01 1.3759e-01
 1.0540e-01 7.9440e-02 5.8660e-02 4.2720e-02 3.1010e-02 2.2010e-02
 1.4850e-02 1.0160e-02 6.7300e-03 4.2600e-03 2.7600e-03 1.6800e-03
 1.0300e-03 5.9000e-04 3.2000e-04 2.0000e-04 7.0000e-05 5.0000e-05
 3.0000e-05 2.0000e-05 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[51]:
```
X_subexp_complete_weak = depgraph.generate(G_complete, 100000, rng,␣
↪subgaussian=False, weak=True)
```

[52]:
```
p_subexp_complete_weak = depgraph.estimate(X_subexp_complete_weak, t,␣
↪credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subexp_complete_weak)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 9.1921e-01 8.3723e-01 7.5768e-01 6.7978e-01 6.0638e-01
 5.3608e-01 4.6964e-01 4.0886e-01 3.5332e-01 3.0300e-01 2.5689e-01
 2.1717e-01 1.8075e-01 1.4863e-01 1.2169e-01 9.8370e-02 7.9160e-02
 6.2680e-02 4.8910e-02 3.7930e-02 2.9590e-02 2.2820e-02 1.7170e-02
 1.2810e-02 9.5800e-03 7.1500e-03 5.2100e-03 3.7800e-03 2.7500e-03
 1.9700e-03 1.3200e-03 9.5000e-04 7.2000e-04 4.3000e-04 3.2000e-04
 2.3000e-04 1.4000e-04 7.0000e-05 4.0000e-05 3.0000e-05 1.0000e-05
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[53]:
```
X_subgaussian_dense_weak = depgraph.generate(G_dense, 100000, rng,␣
↪weak=True)
```

[54]:
```
p_subgaussian_dense_weak = depgraph.estimate(X_subgaussian_dense_weak,␣
↪t, credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subgaussian_dense_weak)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 8.9379e-01 7.8820e-01 6.8750e-01 5.9028e-01 4.9981e-01
 4.1838e-01 3.4592e-01 2.8098e-01 2.2420e-01 1.7738e-01 1.3766e-01
 1.0489e-01 7.9790e-02 5.9510e-02 4.3300e-02 3.1010e-02 2.1720e-02
 1.4900e-02 9.8700e-03 6.5600e-03 4.2600e-03 2.8500e-03 1.7300e-03
 9.3000e-04 4.8000e-04 3.1000e-04 2.0000e-04 1.1000e-04 7.0000e-05
 5.0000e-05 3.0000e-05 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

[55]:
```
X_subexp_dense_weak = depgraph.generate(G_dense, 100000, rng,␣
↪subgaussian=False, weak=True)
```

[56]:
```
p_subexp_dense_weak = depgraph.estimate(X_subexp_dense_weak, t,␣
↪credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subexp_dense_weak)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 9.1915e-01 8.3701e-01 7.5806e-01 6.8073e-01 6.0744e-01
 5.3715e-01 4.7030e-01 4.0888e-01 3.5239e-01 3.0092e-01 2.5578e-01
```

```
 2.1580e-01 1.8031e-01 1.4919e-01 1.2221e-01 9.9430e-02 8.0360e-02
 6.3940e-02 4.9730e-02 3.9170e-02 3.0050e-02 2.2730e-02 1.7390e-02
 1.3180e-02 9.9900e-03 7.3400e-03 5.3400e-03 3.7500e-03 2.6500e-03
 1.7800e-03 1.3500e-03 9.5000e-04 7.8000e-04 5.4000e-04 3.7000e-04
 2.5000e-04 1.6000e-04 1.1000e-04 7.0000e-05 4.0000e-05 1.0000e-05
 1.0000e-05 1.0000e-05 1.0000e-05 1.0000e-05 1.0000e-05 1.0000e-05
 1.0000e-05 1.0000e-05]
```

[57]:
```
X_subgaussian_sparse_weak = depgraph.generate(G_sparse, 100000, rng,␣
 ↪weak=True)
```

[58]:
```
p_subgaussian_sparse_weak = depgraph.
 ↪estimate(X_subgaussian_sparse_weak, t, credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subgaussian_sparse_weak)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 8.9409e-01 7.8795e-01 6.8814e-01 5.9224e-01 5.0532e-01
 4.2355e-01 3.5022e-01 2.8496e-01 2.2966e-01 1.8175e-01 1.4194e-01
 1.0934e-01 8.3410e-02 6.2070e-02 4.5090e-02 3.1980e-02 2.2390e-02
 1.5620e-02 1.0250e-02 6.9000e-03 4.3300e-03 2.8100e-03 1.6500e-03
 1.1000e-03 6.0000e-04 3.8000e-04 2.6000e-04 1.6000e-04 1.3000e-04
 9.0000e-05 4.0000e-05 3.0000e-05 2.0000e-05 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00]
```

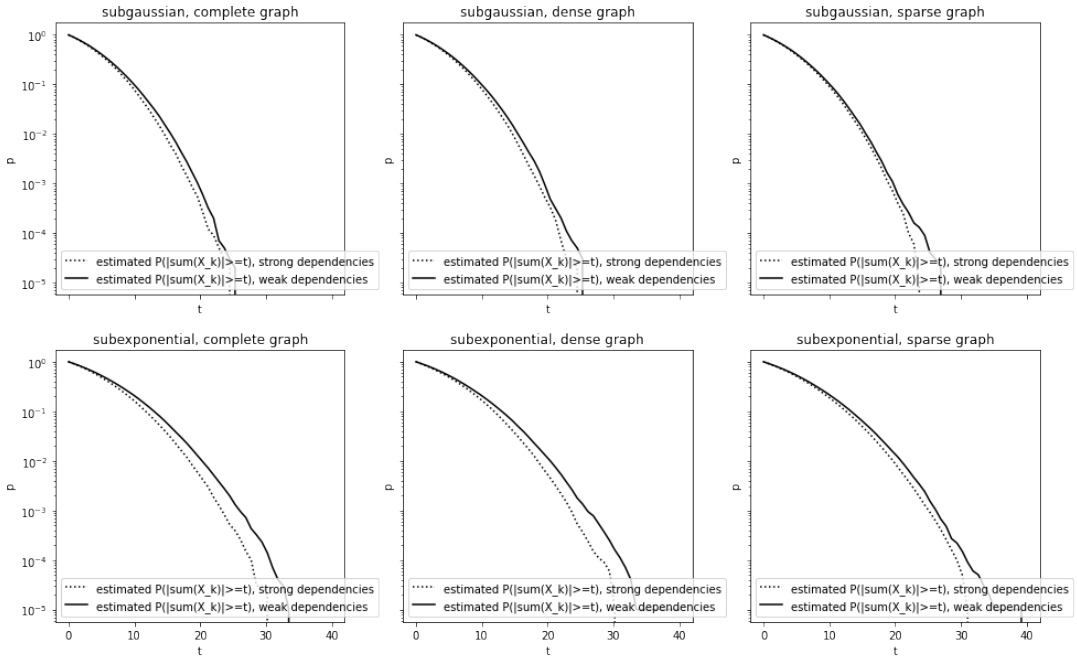
[59]:
```
X_subexp_sparse_weak = depgraph.generate(G_sparse, 100000, rng,␣
 ↪subgaussian=False, weak=True)
```

[60]:
```
p_subexp_sparse_weak = depgraph.estimate(X_subexp_sparse_weak, t,␣
 ↪credint=False)
print('estimated P(|sum(X_i)|>=t):')
print(p_subexp_sparse_weak)
```

```
estimated P(|sum(X_i)|>=t):
[1.0000e+00 9.1946e-01 8.3974e-01 7.6138e-01 6.8598e-01 6.1403e-01
 5.4489e-01 4.7893e-01 4.1836e-01 3.6273e-01 3.1163e-01 2.6608e-01
 2.2564e-01 1.8950e-01 1.5815e-01 1.3043e-01 1.0701e-01 8.6290e-02
 6.9480e-02 5.5470e-02 4.4110e-02 3.4540e-02 2.6540e-02 2.0440e-02
 1.5600e-02 1.2010e-02 8.8800e-03 6.4600e-03 4.6100e-03 3.3800e-03
 2.4100e-03 1.5300e-03 1.0500e-03 6.7000e-04 4.8000e-04 2.7000e-04
 2.2000e-04 1.5000e-04 9.0000e-05 6.0000e-05 5.0000e-05 3.0000e-05
 2.0000e-05 1.0000e-05 1.0000e-05 1.0000e-05 1.0000e-05 1.0000e-05
 1.0000e-05 0.0000e+00]
```

Overview of the examples with weak dependencies

[61]:
```
fig, axs = plt.subplots(2, 3, sharex=True, sharey=True, figsize=(16,10))


for index in [(0,0), (0,1), (0,2), (1,0), (1,1), (1,2)]:
    axs[index].set_xlabel("t")
    axs[index].set_ylabel("p")
    axs[index].set_yscale("log")

axs[0,0].set_title("subgaussian, complete graph")
axs[0,0].plot(t, p_subgaussian_complete, color="black",␣
 ↪linestyle="dotted", label="estimated P(|sum(X_k)|>=t), strong␣
 ↪dependencies")
axs[0,0].plot(t, p_subgaussian_complete_weak, color="black",␣
 ↪label="estimated P(|sum(X_k)|>=t), weak dependencies")

axs[0,1].set_title("subgaussian, dense graph")
axs[0,1].plot(t, p_subgaussian_dense, color="black",␣
 ↪linestyle="dotted", label="estimated P(|sum(X_k)|>=t), strong␣
 ↪dependencies")
axs[0,1].plot(t, p_subgaussian_dense_weak, color="black",␣
 ↪label="estimated P(|sum(X_k)|>=t), weak dependencies")

axs[0,2].set_title("subgaussian, sparse graph")
axs[0,2].plot(t, p_subgaussian_sparse, color="black",␣
 ↪linestyle="dotted", label="estimated P(|sum(X_k)|>=t), strong␣
 ↪dependencies")
axs[0,2].plot(t, p_subgaussian_sparse_weak, color="black",␣
 ↪label="estimated P(|sum(X_k)|>=t), weak dependencies")

axs[1,0].set_title("subexponential, complete graph")
axs[1,0].plot(t, p_subexp_complete, color="black", linestyle="dotted",␣
 ↪label="estimated P(|sum(X_k)|>=t), strong dependencies")
axs[1,0].plot(t, p_subexp_complete_weak, color="black",␣
 ↪label="estimated P(|sum(X_k)|>=t), weak dependencies")

axs[1,1].set_title("subexponential, dense graph")
axs[1,1].plot(t, p_subexp_dense, color="black", linestyle="dotted",␣
 ↪label="estimated P(|sum(X_k)|>=t), strong dependencies")
axs[1,1].plot(t, p_subexp_dense_weak, color="black", label="estimated␣
 ↪P(|sum(X_k)|>=t), weak dependencies")

axs[1,2].set_title("subexponential, sparse graph")
axs[1,2].plot(t, p_subexp_sparse, color="black", linestyle="dotted",␣
 ↪label="estimated P(|sum(X_k)|>=t), strong dependencies")
```

```
axs[1,2].plot(t, p_subexp_sparse_weak, color="black", label="estimated␣
 ↪P(|sum(X_k)|>=t), weak dependencies")

for index in [(0,0), (0,1), (0,2), (1,0), (1,1), (1,2)]:
    axs[index].legend()

fig.savefig('overview2.eps')
fig.show()
```

```
The PostScript backend does not support transparency; partially␣
 ↪transparent
artists will be rendered opaque.
/tmp/ipykernel_8461/1012412642.py:37: UserWarning: Matplotlib is␣
 ↪currently using
module://matplotlib_inline.backend_inline, which is a non-GUI backend,␣
 ↪so cannot
show the figure.
  fig.show()
```