Розділ 1: Статистична постановка задач розпізнавання
Домашнє завдання

Перш ніж почати

Вимоги

- якомога більше дій повинно бути описано словами: це дозволяє перевірити розуміння матеріалу та перевіряти роботи на унікальність;
- для кожної змінної, константи і функції повинно бути вказано множину (або область значень і визначення), якій вона належить: це дозволяє запобігати багатьох помилок та покращує зрозумілість роботи;
- письмові роботи виконуються індивідуально.

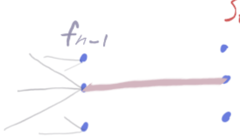
Рекомендації

- ретельно перевіряйте свою роботу перед тим, як відправляти її викладачеві;
- якщо завдання виконується на аркуші паперу і фотографується, намагайтеся задіяти достатньо освітлення і фотографувати якомога рівніше, щоб усі частини сторінки з розв'язком було добре видно.

Ілюстрації

Рекомендації до виконання письмових завдань на прикладах і картинках

Звідки такий висновок?



$\Rightarrow d_{n-1} \in \arg\min_{d} f_{n-1}(d) = F_{n-1}(d)$ ← колізія імені d

d ← в якій множині шукати d?

Малюнок не є доведенням

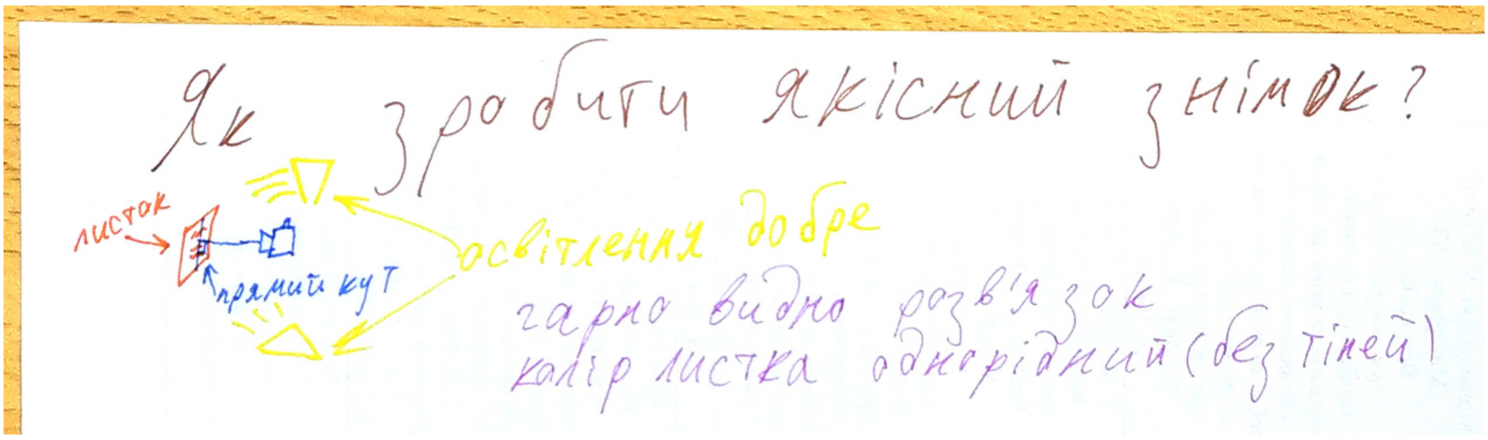
$\min_{d \in D} \left[f_{n-1}(d) + \lambda(d_n) \right] =$

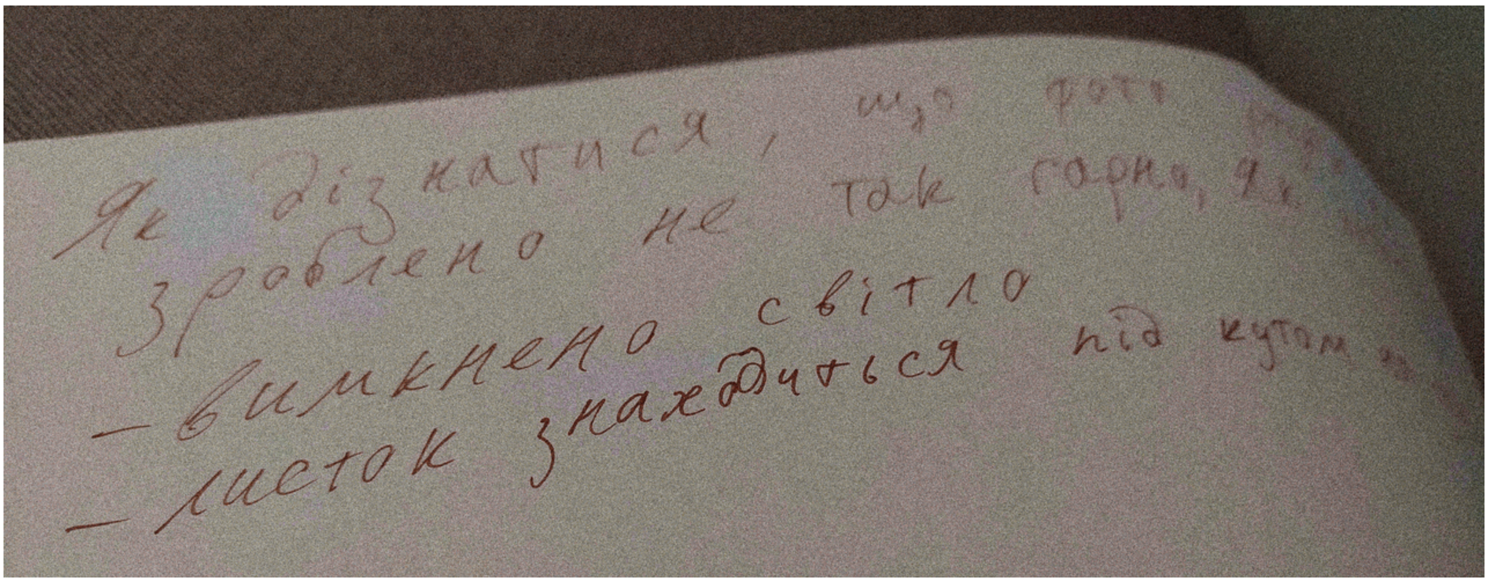
Оскільки $\lambda(d_n)$ не залежить від d, його можна винести як константу за межі мінімізації

детальні коментарі

$= \left[\min_{d \in D} f_{n-1}(d) \right] + \lambda(d_n), \ \forall d_n \in D, \forall n = N..1.$

вказано множини   визначено всі змінні





1 Основне (2 бали)

1.1 Мінімізація часткового ризику

Показати, що для таких стратегій q^*, що

$$q^* \in \underset{q:X\to D}{\operatorname{argmin}} \sum_{\substack{k\in K\\ x\in X}} w\left(q\left(x\right), k\right) \cdot p\left(x, k\right),$$

виконується

$$\forall x \in X : q^*\left(x\right) \in \underset{d\in D}{\operatorname{argmin}} \sum_{k\in K} w\left(d, k\right) \cdot p\left(x, k\right).$$

Навіщо цей факт потрібен на практиці?

1.2 Розпізнавання тривимірних об'єктів

Розв'язати задачу 2.1 (додаткова задача цього розділу) з дещо іншими умовами:

- $K = \mathbb{R}^{100}$ — множина параметрів моделі;
- $X = \mathbb{R}^{10^6\times 3}$ — множина тривимірних моделей з мільйоном вершин;
- $\otimes = +$ — операція додавання матриць $10^6 \times 3$;
- розподіл шуму та параметрів береться за варіантами, що призначає викладач

Варіант	$p(k)$	$p(N)$
1	$\mathcal{N}$	$\mathcal{N}$
2	$\mathcal{N}$	$\mathcal{L}$
3	$\mathcal{L}$	$\mathcal{N}$
4	$\mathcal{L}$	$\mathcal{L}$

$\mathcal{N}$ — компоненти незалежні та розподілені за стандартним нормальним законом, $\mathcal{L}$ — компоненти незалежні та розподілені за законом Лапласа з масштабом 1 та зсувом 0.

Для допитливих

За посиланням `https://flame.is.tue.mpg.de/interactive-model-viewer` можна побавитися з параметрами k, щоб побачити $g(k)$. За посиланням `https://flame.is.tue.mpg.de` ви можете дізнатися

- що собою може являти g;
- звідки береться розподіл параметрів k;
- як це дозволяє розпізнавати тривимірні об'єкти.

2 Додаткове (3 бали)

2.1 Розпізнавання за допомогою породжуючої моделі

Дано

- групу $\langle X, \otimes \rangle$, де X — скінченна множина сигналів;
- скінченну множину станів K;
- множину можливих розв'язків $D = K$, що співпадає з множиною станів;
- штрафну функцію $w(d, k) = t \cdot [\![d \neq k]\!] + c$, де t — додатнє дійсне число, c — невід'ємне дійсне число, а $[\![\cdot]\!]$ — дужки Айверсона.

Статистичну модель $p(x, k)$ задано опосередковано: відомо, що $x = g(k) \otimes N$, де $N : \Omega \to X$ — шум з відомим розподілом, а $g : K \to X$ — відома бієктивна функція, що для кожного стану генерує певний "ідеальний" образ; апріорні ймовірності $p(k)$ теж відомі для всіх $k \in K$.

Знайти оптимальну стратегію q^* (якомога краще спростити вираз)

$$q^* \in \operatorname*{argmin}_{q:X \to D} \sum_{\substack{k \in K \\ x \in X}} w(q(x), k) \cdot p(x, k) .$$

- Чи всі властивості групи є необхідними для використання отриманої стратегії?
- Чи є достатнім для функції g бути сюр'єктивною?
- Чи стане простішою задача, якщо $\forall k \in K : p(k) = \frac{1}{|K|}$?

3 Комп'ютерне (4 бали)

Перш ніж почати

- спочатку розв'язується теорія, а вже потім на її основі пишеться код: це дозволяє не починати роботу з реалізації алгоритму, що є невірнім;
- бажано перевірити алгоритм на папірці: отримані дані також можна буде використовувати в автотестах;
- мова програмування довільна;
- дозволяється написання коду в бригадах розміром до двох людей; за commit messages буде визначатися внесок кожного учаснику, і на основі цього робитися висновок щодо того, чи було роботу виконано бригадою чи однією людиною;
- бажано використовувати технології, що спрощують збірку й запуск коду: CMake для C, C++ і Fortran, setuptools для Python, npm для NodeJS, Maven/Ant для Java/Scala і так далі.

Критерії оцінювання

- 2 бали (обов'язкова умова): коректно працююча програма з кодом у доступному викладачеві репозитарії (наприклад, GitHub, GitLab, BitBucket) та наявність теоретичного розв'язку поставленої задачі;
- 1 бал: наявність автотестів (наприклад, doctest у Python, Boost.Test у C++, JUnit у Java);
- 1 бал: наявність змістовних commit messages та CI (наприклад, TravisCI, Jenkins, GitLab Runner).

Якщо не знаєте, що обрати для тестування або CI, або як писати тести чи прив'язати до свого репозитарію CI, зверніться до викладача.

3.1 Бінарна штрафна функція

Задача

На вхід програмі через WebSocket передаються бінарні зашумлені зображення відомих еталонів. Шум — набір незалежних однаково розподілених випадкових величин з розподілом Бернуллі і параметром $0 \leq p \leq 1$, який визначається користувачем. Шум застосовується до кожного пікселя за допомогою виключної диз'юнкції (XOR). Потрібно мінімізувати ризик байєсової стратегії для штрафної функції $w(k,d) = [\![k \neq d]\!]$.

Мета

Закріпити навички максимізації апостеріорної ймовірності.

Завдання

Завдання First з https://sprs.herokuapp.com. Програма має працювати за умов

1. рівень шуму 0 (без шуму всі відповіді мають бути вірними);
2. рівень шуму 1 (при повній інверсії всі відповіді мають бути вірними);
3. рівень шуму 0.4 і менше (а також 0.6 і більше) й масштабі 20 на 20 і більше, адже, коли картинка має розмір 100 на 60, ймовірність "перетворення" однієї цифри на іншу після накладання такого рівня шуму надто мала, тому відповідь майже завжди повинна бути вірною.

3.2 Штрафна функція L_1

Задача

На вхід програмі через WebSocket передаються ненормовані значення $p(k \mid x)$. Мінімізувати ризик байєсової стратегії для штрафної функції $w(k,d) = |k-d|$. Потужність множини $K = \{0, 1, \ldots, |K|-1\}$ визначає користувач.

Мета

Засвоїти на практиці, що максимізація апостеріорної ймовірності не у всіх випадках є оптимальною стратегією.

Завдання

Завдання Second з https://sprs.herokuapp.com.

Розділ 2: Навчання з учителем
Домашнє завдання

Вважається, що на вхід алгоритмам подаються коректні дані — такі, для яких існує розділяюча гіперплощина. Завдання виконуються за варіантами, які призначає викладач.

1 Основне (2 бали)

1.1 Варіант 1

Довести, що алгоритм Козинця для пошуку гіперплощини, що розділяє дві множини $X_1 \subset \mathbb{R}^n$ і $X_2 \subset \mathbb{R}^n$, збігається за скінченну кількість корекцій. Навести та довести якомога кращу оцінку кількості корекцій.

1.2 Варіант 2

Довести, що алгоритм навчання персептрону для пошуку гіперплощини, що розділяє дві множини $X_1 \subset \mathbb{R}^n$ і $X_2 \subset \mathbb{R}^n$, збігається за скінченну кількість корекцій. Навести та довести якомога кращу оцінку кількості корекцій.

2 Додаткове (3 бали)

Побудувати алгоритм навчання та використання класифікатора реалізацій ґауссового вектору з невідомими параметрами. Тобто, знайти такий вектор $\boldsymbol{\mu} \in \mathbb{R}^n$, додатно визначену матрицю $\Sigma \in \mathbb{R}^{n \times n}$ і $\theta \in \mathbb{R}$ для вирішального правила $p\left(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma\right) \overset{1}{\underset{2}{\gtrless}} \theta$.

Варіанти:

1. Варіант 1: взяти за основу алгоритм Козинця.
2. Варіант 2: взяти за основу алгоритм навчання персептрону.

3 Комп'ютерне (4 бали)

Якщо в бригаді з написання програми є представники двох різних варіантів, студенти вільні обрати будь-який з них.

3.1 Класифікація ґауссових векторів

Задача

Програма на вхід приймає три скінченні послідовності двовимірних дійснозначних векторів. Перша послідовність містить значення, що з більшою ймовірністю належать певному (невідомому програмі) ґауссовому розподілу, ніж значення з другої послідовності (за аналогією з завданням 2). Ці дві послідовності програма має використовувати для навчання. Третя послідовність містить довільні вектори, і їх треба класифікувати. На вихід програма виводить індекси тих елементів третьої послідовності, що належать першому класу.

Варіанти:

1. Варіант 1: взяти за основу алгоритм навчання персертрону.
2. Варіант 2: взяти за основу алгоритм Козинця.

Мета

Закріпити навички з використання ядрових методів та лінійних класифікаторів для розпізнавання реалізації випадкових величин з експоненційного сімейства розподілів.

Розділ 3: Навчання без учителя
Домашнє завдання

1 Основне (2 бали)

Дано вибірку $\overline{x}$ з відомої множини X, множину класів $K = \{1, \ldots, m\}$, апостеріорні ймовірності $p(k \mid x), \forall x \in X, \forall k \in K$ і сімейство породжуючих ймовірностей $P_{X|K} \subset [0;1]^{X \times K}$. Показати, що справедлива рівність

$$\max_{p_{X|K} \in P_{X|K}} \sum_{k \in K} \sum_{x \in \overline{x}} p(k \mid x) \cdot \ln p_{X|K}(x \mid k) = \sum_{k \in K} \max_{p_{X|K} \in P_{X|K}} \sum_{x \in \overline{x}} p(k \mid x) \cdot \ln p_{X|K}(x \mid k).$$

Чому не можна винести за $\max$ знак суми по x?

$$\max_{p_{X|K} \in P_{X|K}} \sum_{k \in K} \sum_{x \in \overline{x}} p(k \mid x) \cdot \ln p_{X|K}(x \mid k) \neq \sum_{x \in \overline{x}} \max_{p_{X|K} \in P_{X|K}} \sum_{k \in K} p(k \mid x) \cdot \ln p_{X|K}(x \mid k).$$

2 Додаткове (3 бали)

Вивести спосіб застосування ЕМ-алгоритму до задачі розділення вибірки $\overline{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ на два класи $K = \{1, 2\}$. Відомо, що елементи вибірки є двовимірними дійснозначними векторами $\boldsymbol{x}_i \in \mathbb{R}^2, i = \overline{1, n}$, що є реалізаціями незалежних випадкових величин з нормальним розподілом з невідомими математичними сподіваннями і коваріаційними матрицями $P_{X|K} \sim \{\mathcal{N}(\boldsymbol{\mu}, \Sigma) : \boldsymbol{\mu} \in \mathbb{R}^2, \Sigma \in \mathbb{R}^{2\times 2}\}$.

3 Комп'ютерне (4 бали)

Задача

Припустимо, що бінарні зображення моделюються матрицею назалежних випадкових величин, що мають розподіл Бернуллі з відомими (та в загальному випадку різними) параметрами p_{xy} для кожного пікселя $\langle x, y \rangle$. Побудуйте та реалізуйте ЕМ-алгоритм для кластеризації двох класів зображень з MNIST або EMNIST.

Детальний розбір задачі: `http://blog.manfredas.com/expectation-maximization-tutorial`.

Мета

Закріпити на практиці використання ЕМ-алгоритму самонавчання.

Завдання

1. Обрати два візуально різні класи зображень (наприклад, цифри 0 і 1 або букви V и S).
2. Реалізувати ЕМ-алгоритм для розв'язку поставленої задачі.
3. Запустити свою програму для самонавчання на навчальній вибірці обраних класів.
4. Перевірити отримані розподіли на тестовій вибірці.

Розділ 4: Задоволення обмежень
Домашнє завдання

1 Основне (2 бали)

1.1 Задоволення обмежень — одна з простіших задач розмітки

Узагальнена задача розмітки на напівкільці $\langle R, \oplus, \otimes \rangle$ для відомого неорієнтованого графу $\langle T, \tau \subset T^2 \rangle$, відомої скінченної множини міток K і відомої функції $g : \tau \to R$ полягає у підрахунку значення

$$r = \bigoplus_{k:T \to K} \bigotimes_{tt' \in \tau} g_{tt'} \left(k_t, k_{t'} \right)$$

і (якщо має сенс) пошуку такої розмітки $k^* \in K^T$ що

$$\bigotimes_{tt' \in \tau} g_{tt'} \left(k_t^*, k_{t'}^* \right) = r.$$

Припустимо, що ми не маємо спеціалізованих алгоритмів розв'язку задач розмітки на напівкільці $\langle \{0, 1\}, \vee, \wedge \rangle$. Уявімо, що вміємо розв'язувати будь-які задачі розмітки на тих напівкільцях, що наведено нижче. Побудуйте та обґрунтуйте правило зведення забачі перевірки незаперечності обмежень до задачі розмітки на напівкільці, що відповідає вашому варіанту, а також оберіть носій R.

Варіант	$\bigoplus$	$\bigotimes$
1	$+$	$\times$
2	min	$+$
3	min	$\times$
4	min	max
5	max	$+$
6	max	$\times$
7	max	min
8	$\cup$	$\cap$

Підказка

Основна складність завдання полягає у пошуку ізоморфізму між напівкільцем $\langle \{0, 1\}, \vee, \wedge \rangle$ та тим, що вказане у варіанті. Ізоморфізмом двох напівкілець $\langle R, \oplus, \otimes \rangle$ і $\left\langle \widetilde{R}, \widetilde{\oplus}, \widetilde{\otimes} \right\rangle$ називається таке бієктивне відображення $f : R \to \widetilde{R}$, що

$$\forall x, y \in R : \begin{cases} f(x \oplus y) = f(x) \widetilde{\oplus} f(y), \\ f(x \otimes y) = f(x) \widetilde{\otimes} f(y). \end{cases}$$

Оскільки носія $\widetilde{R}$ в умові не задано, його також треба знайти.

Для зручності сприйняття замість функцій max і min можна використовувати оператори $\uparrow$ та $\downarrow$ відповідно:

$$x \uparrow y \equiv \max(x, y),$$
$$x \downarrow y \equiv \min(x, y).$$

2 Додаткове (3 бали)

2.1 Задоволення обмежень на ланцюзі

Дано скінченні множини K та $I = \{0, \ldots, n\}$ і функцію $g : \{1, \ldots, n\} \times K^2 \to \{0, 1\}$. Навести та обґрунтувати алгоритм, що має поліноміальну складність та знаходить таке $k^* \in K^I$, що

$$\bigwedge_{i=1}^{n} g_i \left(k_{i-1}^*, k_i^* \right) = \bigvee_{k:I \to K} \bigwedge_{i=1}^{n} g_i \left(k_{i-1}, k_i \right).$$

3 Комп'ютерне (4 бали)

3.1 Судоку

Задача

Сформулювати умови головоломки судоку як задачу пошуку розмітки на напівкільці $\langle \{0, 1\}, \vee, \wedge \rangle$ та реалізувати алгоритм її розв'язку.

Мета

Закріпити навички розв'язку задачі перевірки обмежень на несуперечливість.

Завдання

1. На вхід програма приймає табличку з цифрами від 1 до 9 в заповнених клітинках та 0 в якості пустих клітинок.

2. На вихід програма виводить або розв'язок задачі (табличку з цифрами), або повідомлення про те, що розв'язок не було знайдено.

3. Необов'язковою корисною можливістю буде вивід на екран проміжних результатів: табличок, що містять цифри там, де вже обрано відповідь, та пробіли там, де ще є кілька варіантів можливих цифр.

Примітки

Обраний метод розв'язку повинен розв'язувати якнайменш ті задачі, для яких иснують поліморфізми, що є операторами напіврешіток.

Студенти, що використовують готові рішення для розв'язку задачі задоволення обмежень або судоку, повинні чітко описати алгоритми, що використовуються, а також формально описати випадки, у яких вони дають вірне рішення.

З прикладом розв'язку можна ознайомитись за посиланням `https://www.metalevel.at/sudoku`.

Розділ 5: Вступ до графових моделей у розпізнаванні образів
Домашнє завдання

1 Основне (2 бали)

1.1 Оцінка максимальної правдоподібності для ланцюга Маркова

Дано скінченну множину K, число $n \in \mathbb{N}_+$, функцію $g : \{1, \ldots, n\} \times K^2 \to [0; 1]$, невідомий нормуючий множник $c \in \mathbb{R}_+$ та ланцюг, ймовірність станів якого рахується за формулою

$$p\left(k\right) = c \cdot \exp\left\{\sum_{i=1}^{n} g_i\left(k_{i-1}, k_i\right)\right\}, \quad k \in K^{n+1}.$$

Навести ефективний алгоритм пошуку значення (вважаючи, що множити та складати числа ми **вміємо** з довільною точністю)

$$p^* = \max_{k \in K^{n+1}} p\left(k\right).$$

Навести ефективний алгоритм пошуку найбільш ймовірного стану (враховуючи, що точність множення та складання чисел **обмежена**)

$$k^* = \operatorname*{argmax}_{k \in K^{n+1}} p\left(k\right).$$

Розрахувати складність цих алгоритмів.

2 Додаткове (3 бали)

2.1 Задача розмітки на довільному напівкільці та ациклічній структурі

Для довільного напівкільця $\langle R, \oplus, \otimes \rangle$, відомого орієнтованого дерева $\langle T, \tau \rangle$, такого що

$$\begin{aligned} &tt' \in \tau \implies t't \notin \tau \wedge t \neq t', \\ &\{tt', tt''\} \subseteq \tau \implies t' = t'', \end{aligned}$$

відомої скінченної множини K і функції $g : \tau \times K^2 \to R$ вивести ефективний алгоритм пошуку значення

$$G = \bigoplus_{k \in K^T} \bigotimes_{tt' \in \tau} g_{tt'}\left(k_t, k_{t'}\right)$$

та розрахувати його складність.

3 Комп'ютерне (4 бали)

3.1 Основи розпізнавання зашумленого тексту

Задача

Дано англійську абетку (з пробілом) K та бінарні еталонні зображення букв $\gamma : K \to \{0, 1\}^{h \times w}$. Зображення тексту $k \in K^n$ генерується шляхом конкатенації відповідних еталонів по горизонталі та незалежною інверсією кожного пікселя з ймовірністю p, яку вказує користувач.

$$\begin{aligned} &G : K^n \to \{0, 1\}^{h \times (w \cdot n)}, \\ &G\left(k\right) = \gamma\left(k_1\right) \circ \gamma\left(k_2\right) \circ \cdots \circ \gamma\left(k_n\right), \\ &\xi_{ij} \sim Bern\left(p\right), \quad i = \overline{1, h}, \quad j = \overline{1, w \cdot n}, \\ &x_{ij} = G\left(k\right)_{ij} \oplus \xi_{ij}, \quad i = \overline{1, h}, \quad j = \overline{1, w \cdot n}. \end{aligned}$$

Між сусідніми буквами є статистичний зв'язок, що описується умовними ймовірностями біграм $p\left(k_i \mid k_{i+1}\right)$

$$p\left(k\right) = p\left(k_1\right) \cdot \prod_{i=1}^{n-1} p\left(k_{i+1} \mid k_i\right), \quad k \in K^n.$$

Сумісні ймовірності біграм задано в неявному виді через частоти $n : K^2 \to \mathbb{N}$. Ймовірність букви, що знаходиться на першому місці, вважати рівною ймовірності того, що букві передує пробіл. Для отримання більш точних результатів можна також вважати, що ймовірність певної букви бути останньою в тексті дорівнює ймовірності того, що за нею йде пробіл. Отримали

$$p\left(k\right) = p\left(\smile \mid k_n\right) \cdot \left\{ \prod_{i=1}^{n-1} p\left(k_{i+1} \mid k_i\right) \right\} \cdot p\left(k_1 \mid \smile\right), \quad k \in K^n,$$

де $\smile$ означає символ пробілу.

Зауважте, що умовні ймовірності біграм записуються "навпаки": на першому місці йде той символ, що в тексті йде наступним.

Мета

Закріпити навички розпізнавання станів прихованих моделей Маркова.

Завдання

На вхід програмі подається

- шлях до папки, що містить еталони букв (еталони мають назви a.png, b.png, ..., z.png та space.png для пробілу);
- рівень шуму p від 0 до 1;
- зашумлене зображення з однією строкою тексту.

Програма виводить розпізнаний текст.

Ймовірності надає викладач. Також програма повинна містити режим розпізнавання за рівномірних ймовірностей переходів (незалежне розпізнавання кожного символу).