Занятия 6.2. Метод главных компонент

Гирдюк Дмитрий Викторович

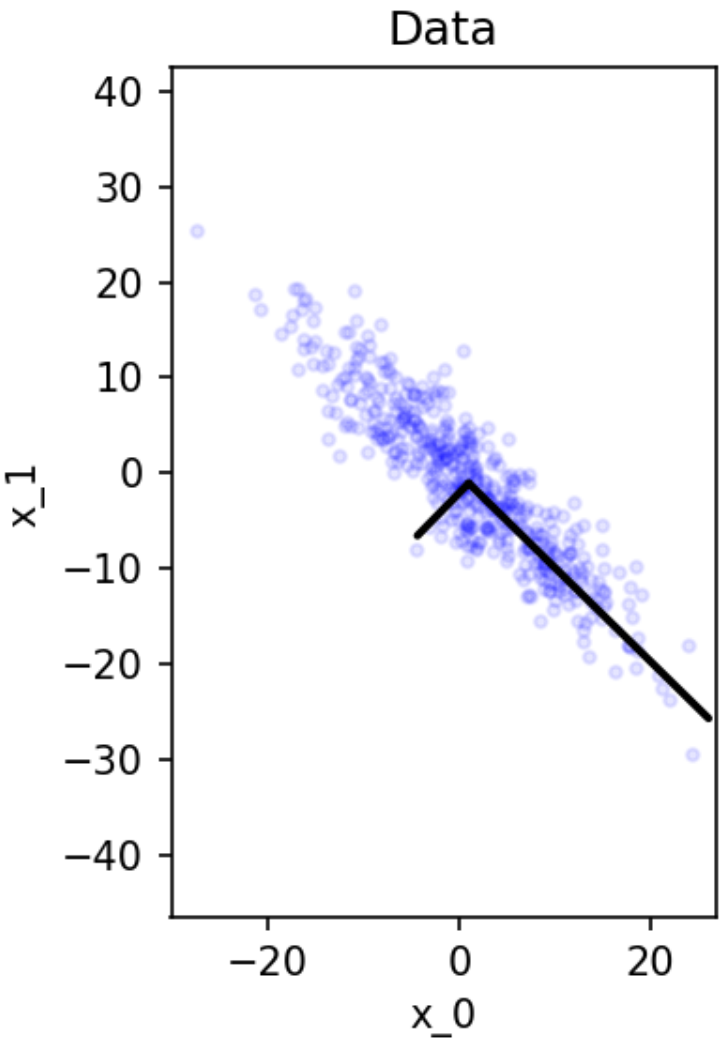
21 октября 2023 г.

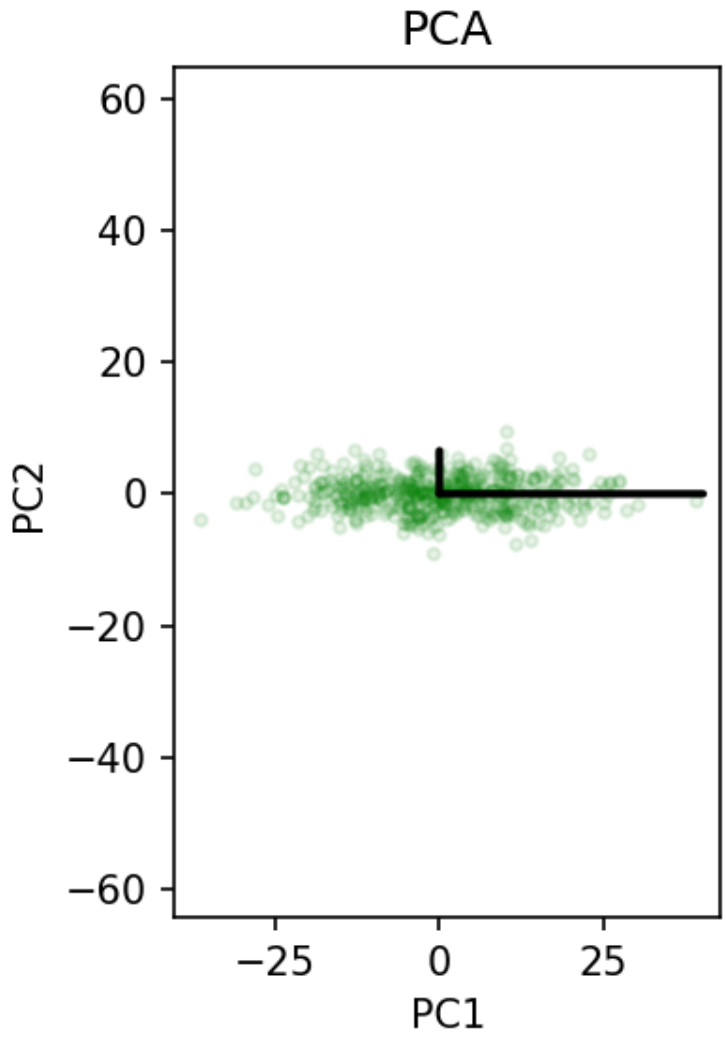
СПбГУ, ПМ–ПУ, ДФС

Метод главных компонент

- Главными компонентами некоторого набора данных $\boldsymbol{X}_{[m\times n]}$ является последовательность из n векторов, каждый из которых наилучшим образом (в смысле минимизации средних квадратов расстояний между наблюдениями и текущим вектором) подгоняется под данные, при этом каждый i-ый вектор ортогонален предыдущим $i-1$ векторам.
- Новый ортогональный базис? Новый ортогональный базис!
- Метод главных компонент (Principal component analysis, PCA) – метод снижения размерности, основанный на построении набора из первых k таких ортогональных векторов.

PCA: визуализация





Общая теория 1

- Хотим получить такой нормированный ортогональный набор векторов $\boldsymbol{v}_j \in R^n, j = 1, 2, \ldots, d$, которыми можно будет саппроксимировать исходные векторы $\boldsymbol{x}_i \in R^n, i = 1, 2, \ldots, m$

$$\boldsymbol{x}_i \approx \sum_{j=1}^{d} c_{ij} \boldsymbol{v}_j, \quad i = 1, 2, \ldots, m$$

- Нормируем (опционально шкалируем) данные!
- От простого к сложному: целевая функция (минимизация дисперсии/разброса проекции точек на главную компоненту) в случае $d = 1$

$$\frac{1}{m} \sum_{i=1}^{m} ||\boldsymbol{x}_i \perp \boldsymbol{v}||_2^2 \longrightarrow \min_{||\boldsymbol{v}||_2 = 1}$$

Общая теория 2

- Теорема Пифагора позволяет преобразовать целевую цункцию

$$||\boldsymbol{x}_i \perp \boldsymbol{v}||_2^2 + \langle \boldsymbol{x}_i, \boldsymbol{v} \rangle^2 = ||\boldsymbol{x}_i||_2^2 \implies$$

$$\implies \frac{1}{m} \sum_{i=1}^{m} \langle \boldsymbol{x}_i, \boldsymbol{v} \rangle^2 = \frac{1}{m} (\boldsymbol{X}\boldsymbol{v})^T (\boldsymbol{X}\boldsymbol{v}) = \frac{1}{m} \boldsymbol{v}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{v} =$$

$$= \frac{1}{m} \boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} \longrightarrow \max_{||\boldsymbol{v}||_2 = 1} \quad (1)$$

- Симметричная матрица $\boldsymbol{A} = \boldsymbol{X}^T \boldsymbol{X}$ есть ничто иное как ковариационная (учитывая шкалирование, еще и корреляционная) матрица. Важное свойство: собственные числа такой матрицы неотрицательны.
- Целевая функция для случай $d > 1$, проекция $\boldsymbol{x}_i$ на векторное подпространство $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$

$$\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{d} \langle \boldsymbol{x}_i, \boldsymbol{v}_j \rangle^2 \longrightarrow \max_{\boldsymbol{V}: ||\boldsymbol{v}_j||_2 = 1}$$

Общая теория 3

- Теперь рассмотрим разложение матрицы $\boldsymbol{A}$

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$$

где матрица $\boldsymbol{V}$ есть ортогональная матрица, а $\boldsymbol{D}$ – диагональная.

- Заметим, что если матрица $\boldsymbol{A} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ (пусть еще $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n \geqslant 0$), то максимум для (1) достигается для $\boldsymbol{v} = \boldsymbol{e}_1$

$$\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v} = \sum_{j=1}^{n} \lambda_j \boldsymbol{v}_j^2$$

- Отсюда следует, что для произвольной матрицы $\boldsymbol{A}$, положив $\boldsymbol{v}_1 = \boldsymbol{V}\boldsymbol{e}_1$, получим

$$\boldsymbol{v}_1^T\boldsymbol{A}\boldsymbol{v}_1 = \boldsymbol{v}_1^T\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T\boldsymbol{v}_1 = \boldsymbol{e}_1^T\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{e}_1 = \boldsymbol{e}_1^T\boldsymbol{D}\boldsymbol{e}_1 = \lambda_1$$

Общая теория 4

- Более того, для любого другого вектора $\hat{\boldsymbol{v}}$, $\hat{\boldsymbol{v}}^T \boldsymbol{A}\boldsymbol{v} \leqslant \lambda_1$.
- Для случая $d > 1$ все выводится аналогичным образом. Оптимальное решение – первые d столбцов матрицы $\boldsymbol{V}$.
- Вопрос: как эту матрицу $\boldsymbol{V}$ искать?
- На самом деле матрица $\boldsymbol{V}$ в разложении $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$ есть ничто иное как матрица, столбцы которой являются собственными векторами матрица $\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X}$

$$\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{A}\boldsymbol{V}\boldsymbol{e}_i = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{e}_i = \boldsymbol{V}\boldsymbol{D}\boldsymbol{e}_i = \lambda_i\boldsymbol{V}\boldsymbol{e}_i = \lambda_i\boldsymbol{v}_i$$

- Последнее, что тут стоит отметить, это уникальность решения: если все собственные числа уникальны, то и разложение уникально. Если же встречаются кратные, то образуется целое подпространство собственных векторов, решающих задачу.

Алгоритмы поиска главных компонент

- Основанный на сингулярном разложении (Singular Value Decomposition, SVD).
- Итерационный алгоритм (Power Iteration).

Сингулярное разложение (SVD)

- Сингулярное разложение

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$$

где $\boldsymbol{U}_{[m\times m]}$ и $\boldsymbol{V}_{[n\times n]}$ ортогональные матрицы, а $\boldsymbol{S}$ – диагональная матрица размерности $m \times n$, значения на диагонали которой отсортированы в убывающем порядке.

- Столбцы матриц $\boldsymbol{U}$ и $\boldsymbol{V}$ называются левыми и правыми сингулярными векторами матрицы $\boldsymbol{X}$ соответственно.

Сингулярное разложение (SVD)

- Левые сингулярные векторы матрицы $\boldsymbol{X}$ есть ничто иное как собственные векторы матрицы $\boldsymbol{X}\boldsymbol{X}^T$. Аналогично, правые сингулярные векторы – собственные векторы матрицы $\boldsymbol{X}^T\boldsymbol{X}$.
- В самом деле

$$\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T = \boldsymbol{U}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{U}^T = \boldsymbol{U}\boldsymbol{D}_1\boldsymbol{U}^T \implies$$
$$\implies \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{D}_1$$

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{D}_2\boldsymbol{V}^T \implies$$
$$\implies \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{D}_2$$

- Отсюда следует алгоритм построения разложения: с помощью библиотек линейной алгебры построить спектральное разложения матриц $\boldsymbol{X}\boldsymbol{X}^T$ и $\boldsymbol{X}^T\boldsymbol{X}$.

Итерационный алгоритм

Algorithm 1: Итерационный алгоритм поиска главной компоненты

input : Матрица $\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X}$

Выбираем произвольный нормированный вектор $\boldsymbol{u}_0$;

for $i = 1, 2, \ldots$ **do**

  $\boldsymbol{u}_i = \boldsymbol{A}^i\boldsymbol{u}_0$;

  if $\boldsymbol{u}_i/||\boldsymbol{u}_i||_2 \approx \boldsymbol{u}_{i-1}/||\boldsymbol{u}_{i-1}||_2$ **then**

    Возвращаем $\boldsymbol{u}_i$;

- Важно отметить (доказывается по индукции): $\boldsymbol{A}^{i+1} = \boldsymbol{A}^i\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}^i\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{D}^{i+1}\boldsymbol{V}^T$
- Как только нашли первую компоненту, проектируем данные ортогонально найденной главной компоненте, т.е. полагаем $\boldsymbol{x}_i := \boldsymbol{x}_i - \langle \boldsymbol{x}_i, \boldsymbol{v}_1\rangle\boldsymbol{v}_1$ и запускаем итерационный алгоритм заново.

Итерационный алгоритм: обсуждение

Почему последовательное домножение на вектор $\boldsymbol{u}_0$ матрицы $\boldsymbol{A}$ приводит нас в конечном итоге к главной компоненте?

- $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$
- $\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{D}$
- Учитывая то, что по столбцам у матрицы $\boldsymbol{V}$ стоят собственные векторы, формирующие ортогональный базис, то любой вектор в этом базисе может быть расписан как $\boldsymbol{u}_0 = \boldsymbol{c}_1\boldsymbol{v}_1 + \ldots + \boldsymbol{c}_n\boldsymbol{v}_n$.
- Тогда получаем

$$\boldsymbol{A}^i\boldsymbol{u}_0 = c_1\boldsymbol{A}^i\boldsymbol{v}_1 + \ldots + c_n\boldsymbol{A}^i\boldsymbol{v}_n = c_1\lambda_1^d\boldsymbol{v}_1 + \ldots + c_n\lambda_n^d\boldsymbol{v}_n =$$

$$= \lambda_1^d(c_1\boldsymbol{v}_1 + c_2\frac{\lambda_2^d}{\lambda_1^d}\boldsymbol{v_2} + \ldots + c_n\frac{\lambda_n^d}{\lambda_1^d}\boldsymbol{v}_n)$$

- Откуда следует, что при $i \to \infty$, учитывая $\lambda_1 \geqslant \ldots \geqslant \lambda_n$, $\boldsymbol{A}^i\boldsymbol{u}_0 \to C(i)\boldsymbol{v}_1$

Обсуждение i

- Простая интерпретация метода главных компонент: компоненты последовательно выбираются так, чтобы дисперсия проекции данных на нее была максимальной. Следует это из центрированности данных и вида целевой функции (1).
- Выбираем число главных компонент на основе объясненной дисперсии, равной кумулятивной сумме отнормированных собственных чисел, соответствующих собственным векторам (главным компонентам). Или используем правило Кайзера:

$$\lambda_i > \frac{1}{n}\text{trace}\boldsymbol{A}$$

- Еще раз, нормализуем (и шкалируем) данные!
- Интерпретация главных компонент зачастую затруднительна.

Обсуждение ii

- Нелинейная структура в данных – используйте другой метод (рассмотрим позже в курсе).
- Например, Kernel PCA! Все отличие лишь в том, что находим собственные векторы не ковариационной матрицы $\boldsymbol{X}^T\boldsymbol{X}$, а ядровой матрицы $\boldsymbol{K}$: $k_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
- Самое главное: $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{V}$. Но мы можем взять лишь первые d компонент для аппроксимации: $\boldsymbol{Y}_d = \boldsymbol{X}\boldsymbol{V}_d$.