ЛЕКЦИЯ 8. KERNEL TRICK

Сергей Николенко

1. Вступление

Пусть есть набор данных в дискретном пространстве. Рассмотрим следующий метод, позволяющий понизить размерность пространства без существенных потерь информации.

Выбирается направление, вдоль которого данные имеют наибольшую дисперсию. Или, другими словами, линейное направление вдоль которого данные наиболее информативны (см. рис. 1).

После первого направления выбираются второе, третье и т.д. и из них составляется базис таким образом, что если из этого базиса оставить только несколько первых векторов и спроецировать все данные на полученное подпространство - потери будут минимальными.

Метод берет данные в виде вектора из x_i и строит матрицу ковариации, диагонализует ее и находит собственные числа и вектора, при этом с.в. матрицы и окажутся теми векторами, вдоль которых максимизируется дисперсия. Сами же собственные вектора выразятся как некоторые линейные комбинации x_i.

2. Выделение главных компонент

На предыдущей лекции мы научились смещать матрицу так, чтобы среднее было равно нулю. Поэтому теперь данные будем всегда считать центрированными (если это не так, то мы всегда можем привести их к центрированному виду).

На рисунке 2 точками отмечены данные, а тонкими линиями - их проекции на вектор в линейном случае.

Пусть данные, которые мы проецируем, образуют не линейную а хорошую нелинейную структуру. В таком случае нам необходимо выделить главные компоненты.

Для этого рассмотрим $\Phi : \mathbb{R}^N \to F$, где Φ - данные. Это отображение переводит вектор $\vec{x} \in \mathbb{R}^N$ в вектор $\vec{X} \in F$.

Произведём анализ главных компонент справа, а не слева:

$$\widetilde{c} = \frac{1}{M}\sum_{j=1}^{M}\Phi(\vec{x_j})\Phi(\vec{x_j})^T$$

$$\lambda\vec{V} = \tilde{c}\vec{V} = \frac{1}{M}\sum_{j=1}^{M}\vec{V}\Phi(\vec{x_j})\Phi(\vec{x_j}) \tag{1}$$

Законспектировали Васильева Екатерина, Смирнов Егор.

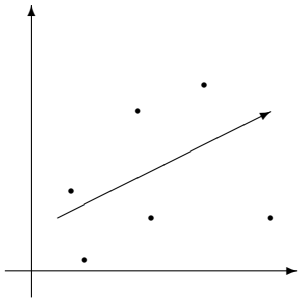

Рис. 1. Данные и направление наибольшей информативности.

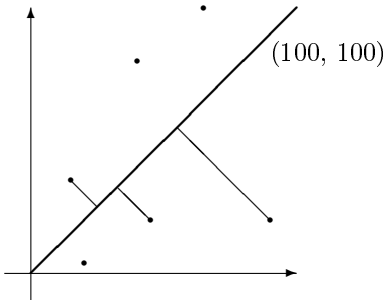

Рис. 2. Собственный вектор матрицы ковариации.

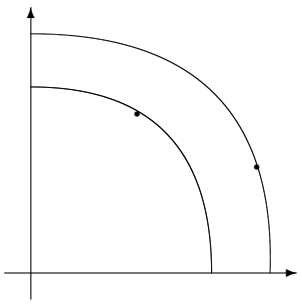

Рис. 3. Нелинейный случай.

$\langle \Phi(\vec{x_1}), ..., \Phi(\vec{x_M}) \rangle$ - подпространство. То что собственный вектор лежит в собственном подпространстве имеет принципиальное значение. Из этого следует:

(1) Можно заменить вектор $\vec{V}$, который может быть в этом пространстве довольно большим, на не более чем M коэффициентов: $\vec{V} = \sum_{i=1}^{M} \alpha_i \Phi(\widetilde{x_i})$.
(2) Выражение (1) сопоставимо с:

$$\forall k \in 1, ...M :$$

$$\lambda \sum_{i=1}^{M} \alpha_i (\Phi(x_i)\Phi(x_k)) = \frac{1}{M} \sum_{i=1}^{M} (\Phi(\widetilde{x}_k)) = \sum_{j=1}^{M} \Phi(x_j)(\Phi(x_i)\Phi(x_k)) \qquad (2)$$

Попытаемся теперь выразить (1) через α_i, то есть

$$\lambda \sum_{i} \alpha_i \Phi(x_i) = \frac{1}{M} \sum_{j} ((\sum \alpha_i \Phi(x_i))\Phi(x_j))\Phi(x_j)$$

Это равенство является векторным. Возьмем и заменим его на k скалярных равенств. Умножая скалярно правую и левую части каждого равенства на $\Phi(x_1)$, $\Phi(x_2)$ и т.д. из выражения (2) можно найти α_i.

Упростим запись:

Пусть есть матрица $K = (\Phi(x_i)\Phi(x_j))_{ij}$

Тогда

$$\lambda(K\vec{\alpha}) = \frac{1}{M} K^2 \vec{\lambda} \qquad (3).$$

Замечание: K - симметрическая и положительно определенная матрица, ее собственные вектора порождают все пространство. Поэтому если сократить на K, то мы ничего не потеряем. Таким образом из (3) получаем $(M\lambda)\vec{\alpha} = K\vec{\alpha}$, где α - элементы собственных векторов для K, а M_α - соответствующие собственные числа.

Рассмотрим матрицу K. Диагонализуем ее и найдем собственные числа и собственные вектора. Получив их, сможем получить α и λ. Напомним, что α - компоненты собственного вектора матрицы $\widetilde{C}$ в базисе $\langle \Phi(\widetilde{x}_1), ..., \Phi(\widetilde{x}_M) \rangle$, а λ - собственный числа матрицы $\widetilde{C}$.

Теперь нам необходимо найти компоненты проекции нового вектора на подпространство.

$$(\Phi(x)\vec{V}^k) = \sum_{j=1}^{M} \alpha_j^k (\Phi(x)\Phi(x_j)),$$

где $\vec{V}^k$ - с.в.

Это верно, так как $\overline{V}^k = \sum_{j=1}^{m} \alpha_j^k \Phi(x_j)$

Основная идея здесь - что все рассуждения велись в терминах скалярного произведения и больше нигде не использовалось пространство F. Оно используется только в $\Phi(\vec{x})\Phi(\vec{y})$. Из этого следует что нужно только уметь вычислять скалярное произведение в этом пространстве:

$\Phi(\vec{x})\Phi(\vec{y}) = K(\vec{x}, \vec{y})$, где K называется *ядром*.

Если ядро $K(\vec{x}, \vec{y})$ можно быстро вычислить, то и$\Phi : \mathbb{R}^N \to F$ можно быстро вычислить.

3. Примеры ядер

Пример 1

Рассмотрим, какому пространству F будет соответствовать такое K:

$$K(\vec{x}, \vec{y}) = (\vec{x}, \vec{y})^2.$$

Преобразуем:

$$K(\vec{x},\vec{y}) = (x_1y_1 + x_2y_2)^2 = x_1^2y_1^2 + 2x_1y_1x_2y_2 + x_2^2y_2^2 = \\ = (x_1^2, x_2^2, \sqrt{2}x_1x_2)\left(y_1^2, y_2^2, \sqrt{2}x_1x_2\right).$$

Получается, что такой выбор ядра отвечает переходу в трёхмерное пространство, оси которого являются мономами второго порядка. Стоит обратить внимание на рост размерности пространства с двух до трёх.

Рассмотрим в общем виде:

$$K(\vec{x},\vec{y}) = (\vec{x},\vec{y})^d = C_d(\vec{x})C_d(\vec{y})$$

$$C_d(x_1..x_N)^d = x_1^{d-1} + x_1^{d-1}x_2 + ... + x_N^d$$

Это все мономы степени d. Размерность растет как N^d. Такой быстрый рост размерности сильно увеличивает сложность вычислений. Например, для обработки образа размером 16×16 с использованием мономов пятого порядка, размерность вектора на входе 256, а на выходе - $2^{5\times8}$.

Мы работаем в подпространстве, для которого размерность $d \leq$ количеству наших векторов. Однако все выделенные нами компоненты происходят из большого пространства.

Пример 2

Пусть мы хотим использовать все степени от 1 до d, а не только саму d, то есть чтобы новый вектор выглядел не как

$$(x_1^2, x_2^2, x_1x_2),$$

а как

$$(x_1^2, x_2^2, x_1x_2, x_1, x_2, 1).$$

Для этого ядро нужно модифицировать одним из следующих образов:

(1) $K(\vec{x},\vec{y}) = \frac{(\vec{x},\vec{y})^{d+1}-1}{(\vec{x},\vec{y})-1} = (\vec{x},\vec{y})^d + ... + 1.$

(2) $K(\vec{x},\vec{y}) = ((\vec{x},\vec{y}) + c)^d.$

У данного метода могут возникнуть проблемы в случае, если одна размерность сильно больше другой (отличие на несколько порядков).

4. Задача на сообразительность:

Как использовать нейронную сеть чтобы найти главные компоненты?

Ответ: Обучаем нейронную сеть на тождественной функции (см. рис. 4). Сеть выполняет два преобразования: прямое и обратное, получая на выходе снова N элементов. При этом результат должен как можно меньше отличаться от входных данных. При таком подходе в сети получается внутренний скрытый уровень с K элементами.

Замечание: Обучение нейронной сети не гарантировано. Можно попасть в локальный минимум и получить не те компоненты, которые мы искали.

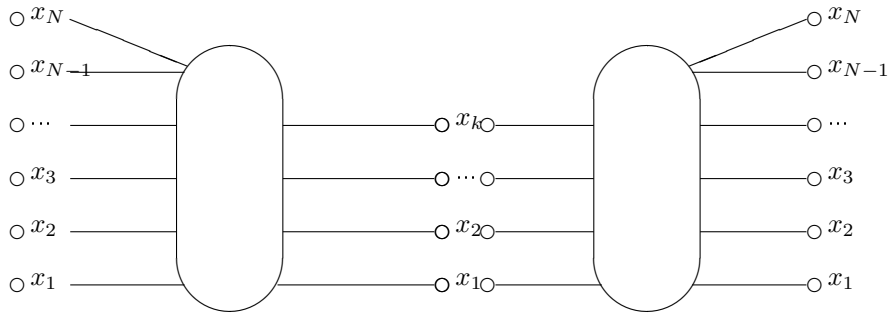

Рис. 4. Нейронная сеть для обучения ($K < N$).

5. Кластеризация

Попытаемся применить kernel trick к задаче кластеризации (разбиения набора данных на несколько групп - *кластеров*). Рассмотрим кластеризацию методом k-средних с заранее известным количеством кластеров K. Начнем с линейного метода:

(1) инициируем $\overline{m}_\alpha = 1..K$ - случайно выбранные точки-центры

(2) для каждого $\vec{x}_i : M_{i,\alpha} = \begin{cases} 1, \text{ если } ||\overline{x}_i - \vec{m}_\alpha||^2 \le ||\vec{x}_i - \vec{m}_\beta||^2, \\ 0 \text{ в остальных случаях } \forall \beta. \end{cases}$

(3) $\forall \alpha$: $\vec{m}_\alpha = \sum_{i:M_{i,\alpha}=1} \vec{x}_i$ - сдвинули центры

(4) снова начали со 2 шага

Процесс останавливается, когда центры перестают изменяться.

Сделаем теперь метод нелинейным:

$$\begin{cases} \vec{x}_1 = \vec{m}_1 \\ \vec{x}_2 = \vec{m}_2 \\ \qquad ... \\ \vec{x}_k = \vec{m}_k \end{cases}$$

$\vec{x}_{t+1} \to M_{t+1,\alpha}(\vec{x}_{t+1} - \vec{m}^t_\alpha)$, где m_α - среднее всех точек которые к нему принадлежат.

Мы начинаем не со случайных точек. Процесс также является итеративным.

$$\vec{m}_\alpha = \sum_{j=1}^{M} \gamma_{\alpha,j} \Phi(\vec{x}_j)$$

Мы хотим, чтобы они находились в пространстве F

$\vec{m}_\alpha \in \langle \Phi(\vec{x}_1), ... \Phi(\vec{x}_M) \rangle$

Пусть это не так. Тогда спроецируем их на линейную оболочку и расстояние до каждой точки уменьшится. Отсюда и вытекает принадлежность л.о.

Выразим расстояние через K:

$$||\Phi(\vec{x}) - \sum_{j=1}^{M} \gamma_{\alpha j} \Phi(\vec{x}_j)||^2 = K(\vec{x}, \vec{x}) - 2 \sum_{i,j=1}^{M} \gamma_{\alpha j} \gamma_{\alpha i} (\vec{x}_j, \vec{x}_i)$$

Далее мы можем воспользоваться той же процедурой, что и в линейном случае, используя полученную формулу для расстояния. Для того, чтобы сделать апдейт кластера, необходимо пересчитать среднее, то есть:

$$\vec{m}_\alpha^{t+1} = \vec{m}_\alpha^t + \xi(\Phi(\vec{x}_{t+1} - \vec{m}_\alpha^t),$$

$$\text{где } \xi = \frac{M_{t+1,\alpha}}{\sum_{i=1}^{t+1} M_{i,\alpha}} \Rightarrow$$

$$\Rightarrow \sum \gamma_{\alpha_j}^{t+1} \Phi(x_j) = \sum_j \gamma_{\alpha_j}^t \Phi(x_j) + \xi(...)$$

$$\gamma_{\alpha_j}^{t+1} = \begin{cases} \xi, j = t+1 \\ \gamma_{\alpha_j}^t (1-\xi), j \neq t+1 \end{cases}$$

Таким образом мы научились представлять нелинейные зависимости посредством линейного метода через скалярное произведение.