Лекция 13
Преобразования признакового пространства

Е. А. Соколов
ФКН ВШЭ

21 декабря 2016 г.

Нам уже известно некоторое количество способов преобразования признаков: можно добавлять признаки более высоких порядков, логарифмировать их или применять другие нелинейные преобразования, отбирать с помощью L_1-регуляризации. Или, например, можно порождать новые признаки с помощью решающих деревьев. В данной лекции мы обсудим два более продвинутых подхода к изменению признакового пространства: ядра, которые позволяют повышать размерность пространства без вычислительных трудностей, и метод главных компонент, который находить оптимальное линейное подпространство.

1 Ядровые методы

§1.1 Восстановление нелинейных зависимостей линейными методами

Линейные методы классификации и регрессии являются хорошо изученными и обоснованными, однако предположение о линейной зависимости зачастую оказывается неверным в задачах машинного обучения. Оказывается, что линейные методы можно применять и для восстановления нелинейных зависимостей, если предварительно перейти к новым признакам.

Рассмотрим простой пример. На рис. 1 показана двумерная выборка с двумя классами, разделяющая поверхность для которой никак не может быть приближена гиперплоскостью. В то же время, если добавить третий признак $x_3 = x_1^2 + x_2^2$, то выборку можно будет идеально разделить гиперплоскостью вида $x_3 = C$ (рис. 2). Такое пространство называется *спрямляющим*. В новом признаковом пространстве разделяющая поверхность является линейной, однако после ее проецирования на исходное пространства она окажется нелинейной.

Модель, в которой зависимость ищется как линейная комбинация нелинейных функций от выборки, называется *линейной моделью над базисными функциями* [1]. Например, для задачи регрессии она имеет вид

$$a(x) = \sum_{i=1}^{m} w_i \varphi_i(x),$$

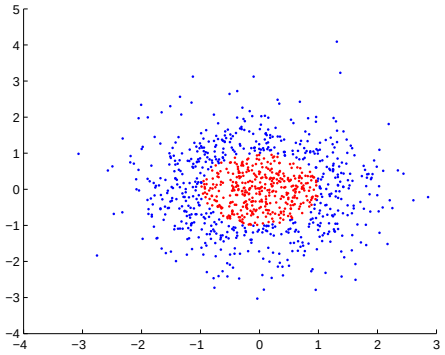

Рис. 1. Выборка с нелинейной разделяющей поверхностью. Разные классы обозначены разными цветами.

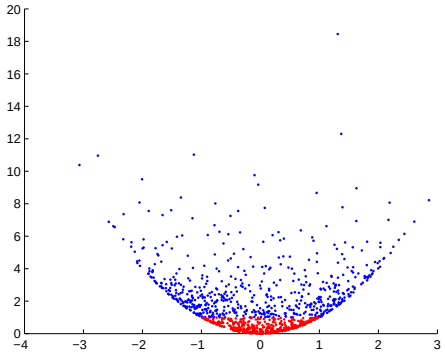

Рис. 2. Выборка после добавления третьего признака. Изображена проекция на первый и третий признаки.

где $\varphi_i(x)$ — произвольные нелинейные функции от признаков (*базисные* функции). Проблема заключается в том, что на практике заранее нельзя сказать, какие именно базисные функции нужно взять, чтобы добиться линейной разделимости, поэтому приходится брать сразу большой набор таких функций (например, все мономы не больше определенной степени). В этом случае число признаков оказывается очень большим, из-за чего процесс обучения становится трудоемким как по времени, так и по памяти. Однако в некоторых случаях оказывается, что достаточно уметь быстро вычислять скалярные произведения объектов друг на друга.

§1.2 Двойственное представление для линейной регрессии

Рассмотрим задачу построения линейной регрессии с квадратичной функцией потерь и квадратичным регуляризатором:

$$Q(w) = \frac{1}{2}\sum_{i=1}^{\ell}\left\{\sum_{j=1}^{m} w_j\varphi_j(x_{ij}) - y_i\right\}^2 + \frac{\lambda}{2}\|w\|^2 = \frac{1}{2}\|\Phi w - y\|^2 + \frac{\lambda}{2}\|w\|^2 \to \min_w,$$

где Φ — матрица, в которой i-я строка представлена вектором $(\varphi_1(x_i), \ldots, \varphi_m(x_i))$. Дифференцируя функционал $Q(w)$ и приравнивая его нулю, получаем

$$w = -\frac{1}{\lambda}\Phi^T(\Phi w - y).$$

Отсюда следует, что решение является линейной комбинацией строк матрицы Φ:

$$w = \Phi^T a,$$

где за a мы обозначили вектор $-\frac{1}{\lambda}(\Phi w - y)$. Подставим это представление в функционал:

$$Q(a) = \frac{1}{2}\|\Phi\Phi^T a - y\|^2 + \frac{\lambda}{2}a^T\Phi\Phi^T a \to \min_a.$$

Заметим, что теперь функционал зависит не от самой матрицы признаков Ф, а от ее произведения на саму себя $\Phi\Phi^T$. Это матрица скалярных произведений всех возможных пар объектов, называемая также *матрицей Грама*. Будем обозначать ее через

$$K = \Phi\Phi^T = (\langle\varphi(x_i), \varphi(x_j)\rangle)_{i,j=1}^{\ell} = (k(x_i, x_j))_{i,j=1}^{\ell},$$

где $\varphi(x_i) = (\varphi_1(x_i), \ldots, \varphi_m(x_i))$, а $k(x_i, x_j)$ — скалярное произведение объектов, называемое также *функцией ядра*.

Можно показать, что оптимальный вектор a имеет вид

$$a = (K + \lambda I)^{-1} y.$$

Функция регрессии при этом запишется как

$$y(x) = \langle w, \varphi(x)\rangle = w^T\varphi(x) = a^T\Phi\varphi(x) = k(x)^T(K + \lambda I)^{-1}y,$$

где $k(x) = (k(x, x_1), \ldots, k(x, x_\ell))$ — вектор скалярных произведений нового объекта x на объекты обучающей выборки.

Итак, нам удалось переписать функционал и модель так, что они зависят лишь от скалярных произведений объектов. В этом случае при росте размерности нового (спрямляющего) признакового пространства количество требуемой памяти остается константным и имеет порядок ℓ^2. Далее мы покажем, что и вычисление скалярного произведения можно организовать так, что оно будет зависеть лишь от размерности исходного признакового пространства.

§1.3 SVM и kernel trick

Переход к новому признаковому пространству можно применять и в задачах классификации:

$$a(x) = \text{sign}(\langle w, \varphi(x)\rangle + b).$$

В частности, к задаче метода опорных векторов можно построить двойственную:

$$\begin{cases} \sum\limits_{i=1}^{\ell} \lambda_i - \dfrac{1}{2}\sum\limits_{i,j=1}^{\ell} \lambda_i\lambda_j y_i y_j \langle x_i, x_j\rangle \to \max\limits_{\lambda} \\ 0 \leqslant \lambda_i \leqslant C, \quad i = 1, \ldots, \ell, \\ \sum\limits_{i=1}^{\ell} \lambda_i y_i = 0. \end{cases}$$

После того, как она решена, новые объекты классифицируются с помощью алгоритма

$$a(x) = \text{sign}\left(\sum_{i=1}^{\ell} \lambda_i y_i \langle x_i, x\rangle + b\right).$$

Заметим, что как оптимизационная задача, так и итоговый классификатор зависят лишь от скалярных произведений объектов. Подставляя вместо скалярного произведения функцию ядра, мы будем настраивать классификатор в произвольном признаковом пространстве. Такая подмена получила в англоязычной литературе название *kernel trick*.

§1.4 Операции в спрямляющем пространстве

Ядром мы будем называть функцию $K(x,z)$, представимую в виде скалярного произведения в некотором пространстве: $K(x,z) = \langle \varphi(x), \varphi(z) \rangle$, где $\varphi : X \to H$ — отображение из исходного признакового пространства в некоторое *спрямляющее пространство*. На семинарах будет показано, что ядро содержит в себе много информации о спрямляющем пространстве и позволяет производить в нем различные операции, не зная самого отображения $\varphi(x)$ — например, находить расстояния между векторами $\varphi(x)$.

§1.5 Построение ядер

Самый простой способ задать ядро — в явном виде построить отображение $\varphi(x)$ в спрямляющее признаковое пространство. Тогда ядро определяется как скалярное произведение в этом пространстве: $K(x,z) = \langle \varphi(x), \varphi(z) \rangle$. При таком способе, однако, возникают проблемы с ростом вычислительной сложности, о которых уже было сказано выше.

Допустим, в качестве новых признаков мы хотим взять всевозможные произведения исходных признаков. Определим соответствующее отображение

$$\varphi(x) = (x_i x_j)_{i,j=1}^{d} \in \mathbb{R}^{d^2}$$

и найдем ядро:

$$\begin{aligned} K(x,z) = \langle \varphi(x), \varphi(z) \rangle = \langle (x_i x_j)_{i,j=1}^{d}, (z_i z_j)_{i,j=1}^{d} \rangle = \\ = \sum_{i,j=1}^{d} x_i x_j z_i z_j = \\ = \sum_{i=1}^{d} x_i z_i \sum_{j=1}^{d} x_j z_j = \\ = \langle x, z \rangle^2. \end{aligned}$$

Таким образом, ядро выражается через скалярное произведение в исходном пространстве, и для его вычисления необходимо порядка d операций (в то время как прямое вычисление ядра потребовало бы $O(d^2)$ операций).

1.5.1 Неявное задание ядра

Пример с мономами показал, что можно определить ядро так, что оно не будет в явном виде использовать отображение объектов в новое признаковое пространство. Но как убедиться, что функция $K(x,z)$ определяет скалярное произведение в некотором пространстве? Ответ на этот вопрос дает *теорема Мерсера*: функция $K(x,z)$ является ядром тогда и только тогда, когда:

1. Она симметрична: $K(x,z) = K(z,x)$.
2. Она неотрицательно определена, то есть для любой конечной выборки $(x_1, \ldots, x_\ell)$ матрица $K = \big(K(x_i, x_j)\big)_{i,j=1}^{\ell}$ неотрицательно определена.

Проверять условия теоремы Мерсера, однако, может быть достаточно трудно. Поэтому для построения ядер, как правило, пользуются несколькими базовыми ядрами и операциями над ними, сохраняющими симметричность и неотрицательную определенность.

Теорема 1.1 ([2]). *Пусть $K_1(x,z)$ и $K_2(x,z)$ — ядра, заданные на множестве X, $f(x)$ — вещественная функция на X, $\varphi : X \to \mathbb{R}^N$ — векторная функция на X, K_3 — ядро, заданное на $\mathbb{R}^N$. Тогда следующие функции являются ядрами:*

1. $K(x,z) = K_1(x,z) + K_2(x,z)$,
2. $K(x,z) = \alpha K_1(x,z)$, $\alpha > 0$,
3. $K(x,z) = K_1(x,z)K_2(x,z)$,
4. $K(x,z) = f(x)f(z)$,
5. $K(x,z) = K_3(\varphi(x), \varphi(z))$.

Теорема 1.2 ([3]). *Пусть $K_1(x,z), K_2(x,z), \ldots$ — последовательность ядер, причем предел*

$$K(x,z) = \lim_{n\to\infty} K_n(x,z)$$

существует для всех x и z. Тогда $K(x,z)$ — ядро.

Рассмотрим некоторые примеры построения ядер.

1.5.2 Полиномиальные ядра

Пусть $p(v)$ — многочлен с положительными коэффициентами. Покажем, что $K(x,z) = p(\langle x,z\rangle)$ — ядро. Пусть многочлен имеет вид

$$p(v) = \sum_{i=0}^{m} a_i v^i.$$

Будем доказывать по шагам.

1. $\langle x,z\rangle$ —ядро по определению ($\varphi(x) = x$);
2. $\langle x,z\rangle^i$ — ядро как произведение ядер;
3. $a_i\langle x,z\rangle^i$ — ядро как произведение положительной константы на ядро;
4. константный член a_0 — ядро по пункту 4 теоремы 1.1, где $f(x) = \sqrt{a_0}$;
5. $\sum_{i=0}^{m} a_i\langle x,z\rangle^i$ — ядро как линейная комбинация ядер.

Аналогично можно показать, что $p(K(x,z))$ — ядро.

Рассмотрим частный случай полиномиального ядра:

$$K_m(x,z) = (\langle x,z\rangle + R)^m.$$

Распишем степень, воспользовавшись формулой бинома Ньютона:

$$K_m(x,z) = \sum_{i=0}^{m} C_m^i R^{m-i}\langle x,z\rangle^i.$$

Поскольку коэффициенты при скалярных произведениях $C_m^i R^{m-i}$ положительны, то данное ядро действительно является ядром согласно последней задаче. Если расписать скалярные произведения, то можно убедиться, что оно соответствует переводу набора признаков во всевозможные мономы над признаками степени не больше m, причем моном степени i имеет вес $\sqrt{C_m^i R^{m-i}}$.

Заметим, что параметр R контролирует относительный вес при мономах больших степеней. Например, отношение веса при мономе степени $m-1$ к весу при мономе первой степени равно

$$\sqrt{\frac{C_m^{m-1}R}{C_m^1 R^{m-1}}} = \sqrt{\frac{1}{R^{m-2}}},$$

то есть по мере увеличения R вес при мономах старших степеней будет становиться очень небольшим по сравнению с весом при остальных мономах. Можно сказать, что параметр R контролирует сложность модели.

1.5.3 Гауссовские ядра

Гауссовское ядро определяется как

$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right).$$

Покажем, что оно действительно является ядром.

Покажем сначала, что функция $\exp(\langle x,z\rangle)$ является ядром. Представим ее в виде предела последовательности:

$$\exp(\langle x,z\rangle) = \sum_{k=0}^{\infty}\frac{\langle x,z\rangle^k}{k!} = \lim_{n\to\infty}\sum_{k=0}^{n}\frac{\langle x,z\rangle^k}{k!}.$$

Каждый член предельной последовательности является многочленом с положительными коэффициентами, и поэтому является ядром. Предел существует во всех точках (x,z), поскольку ряд Тейлора для функции e^x сходится на всей числовой прямой. Значит, по теореме 1.2 данная функция является ядром.

Аналогично можно доказать, что функция $\exp(\langle x,z\rangle/\sigma^2)$ также является ядром. Гауссовское ядро легко получить из данного путём замены преобразования $\varphi(x)$ на $\varphi(x)/\|\varphi(x)\|$.

Заметим, что можно построить гауссово ядро, используя любое другое ядро $K(x,z)$. В этом случае оно примет вид

$$\exp\left(-\frac{\|\varphi(x)-\varphi(z)\|^2}{2\sigma^2}\right) = \exp\left(-\frac{K(x,x)-2K(x,z)+K(z,z)}{2\sigma^2}\right).$$

Здесь мы расписали расстояние между векторами $\|\varphi(x)-\varphi(z)\|^2$ в спрямляющем пространстве через функцию ядра.

Спрямляющее пространство. Какому спрямляющему пространству соответствует гауссовское ядро? Оно является пределом последовательности полиномиальных ядер при стремлении степени ядра к бесконечности, что наталкивает на мысль, что и спрямляющее пространство будет бесконечномерным. Чтобы показать это формально, нам понадобится следующее утверждение.

Утв. 1.3 ([4]). *Пусть $x_1,\ldots,x_\ell$ — различные точки пространства $\mathbb{R}^d$. Тогда матрица*

$$G = \left[\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right)\right]_{i,j=1}^{\ell}$$

является невырожденной при $\sigma > 0$.

Вспомним также факт из линейной алгебры: матрица Грама системы точек $x_1,\ldots,x_\ell$ невырождена тогда и только тогда, когда эти точки линейно независимы. Поскольку матрица из утверждения 1.3 является матрицей Грама для точек $x_1,\ldots,x_\ell$ в спрямляющем пространстве гауссова ядра, то заключаем, что в данном пространстве существует сколь угодно много линейно независимых точек. Значит, данное пространство является бесконечномерным.

Можно показать это и менее формально. Распишем функцию ядра:

$$\begin{aligned} K(x,z) &= \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|z\|^2}{2\sigma^2}\right)\exp\left(-\frac{\langle x,z\rangle}{\sigma^2}\right) = \\ &= \{\text{раскладываем экспоненту в ряд}\} = \\ &= \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|z\|^2}{2\sigma^2}\right)\sum_{k=0}^{\infty}\frac{\langle x,z\rangle^k}{k!\sigma^{2k}}. \end{aligned}$$

Легко видеть, что для получения k-го слагаемого в спрямляющем пространстве должны быть все мономы степени k над исходными признаками. Поскольку всего в сумме бесконечное число слагаемых, то и размерность спрямляющего пространства должна быть бесконечной.

Роль параметра σ. Заметим, что параметр σ в разложении гауссова ядра в ряд Тейлора входит в коэффициент перед слагаемым $\langle x,z\rangle^k$ как $1/\sigma^{2k}$. Его роль аналогична параметру R в полиномиальных ядрах. Маленькие значения σ соответствуют большим значениям R: чем меньше σ, тем больше вес при мономах большой степени, тем больше риск переобучения.

2 Понижение размерности и метод главных компонент

В машинном обучении часто возникает задача уменьшения размерности признакового пространства. Для этого можно, например, удалять признаки, которые слабо коррелируют с целевой переменной; выбрасывать признаки по одному и проверять качество модели на тестовой выборке; перебирать случайные подмножества признаков в поисках лучших наборов. Ещё одним из подходов к решению задачи является поиск новых признаков, каждый из которых является линейной комбинацией исходных признаков. В случае использования квадратичной функции ошибки при поиске такого приближения получается *метод главных компонент* (principal component analysis, PCA), о котором и пойдет речь.

Пусть $X \in \mathbb{R}^{\ell \times D}$ — матрица «объекты-признаки», где ℓ — число объектов, а D — число признаков. Поставим задачу уменьшить размерность пространства до d. Будем считать, что данные являются центрированными — то есть среднее в каждом столбце матрицы X равно нулю.

Будем искать главные компоненты $u_1, \ldots, u_D \in \mathbb{R}^D$, которые удовлетворяют следующим требованиям:

1. Они ортогональны: $\langle u_i, u_j \rangle = 0$, $i \neq j$;
2. Они нормированы: $\|u_i\|^2 = 1$;
3. При проецировании выборки на компоненты $u_1, \ldots, u_d$ получается максимальная дисперсия среди всех возможных способов выбрать d компонент.

Дисперсия проецированной выборки показывает, как много информации нам удалось сохранить после понижения размерности — и поэтому мы требуем максимальной дисперсии от проекций.

Проекция объекта x на компоненту u_i вычисляется как $\langle x, u_i \rangle$, а проекция всей выборки на эту компоненту — как Xu_i. Если за U_d обозначить матрицу, столбцы которой равны первым d компонентам, проекция выборки на них будет записываться как XU_d, а дисперсия проецированной выборки будет вычисляться как след ковариационной матрицы:

$$\operatorname{tr} U_d^T X^T X U_d = \sum_{i=1}^{d} \|Xu_i\|^2.$$

Начнём с первой компоненты. Сведём все требования к ней в оптимизационную задачу:

$$\begin{cases} \|Xu_1\|^2 \to \max_{u_1} \\ \|u_1\|^2 = 1 \end{cases}$$

Запишем лагранжиан:

$$L(u_1, \lambda) = \|Xu_1\|^2 + \lambda(\|u_1\|^2 - 1).$$

Продифференцируем его и приравняем нулю:

$$\frac{\partial L}{\partial u_1} = 2X^T X u_1 + 2\lambda u_1 = 0.$$

Отсюда получаем, что u_1 должен быть собственным вектором ковариационной матрицы $X^T X$. Учтём это и преобразуем функционал:

$$\|Xu_1\|^2 = u_1^T X^T X u_1 = \lambda u_1^T u_1 = \lambda \to \max_{u_1}$$

Значит, собственный вектор u_1 должен соответствовать максимальному собственному значению.

Для следующих компонент к оптимизационной задаче будут добавляться требования ортогональности предыдущим компонентам. Решая эти задачи, мы получим, что главная компонента u_i равна собственному вектору, соответствующему i-му собственному значению.

После того, как найдены главные компоненты, можно проецировать на них и новые данные. Если нам нужно работать с тестовой выборкой X', то её проекции вычисляются как $Z' = X'U_d$. Отметим также, что в методе главных компонент новые признаки вычисляются как линейные комбинации старых:

$$z'_{ij} = \sum_{k=1}^{D} x'_{ik} u_{kj}.$$

Альтернативные постановки. Существует несколько других постановок задачи понижения размерности, приводящих к методу главных компонент.

Первый способ основан на матричном разложении. Будем искать матрицу с новыми признаковыми описаниями $Z \in \mathbb{R}^{\ell \times d}$ и матрицу проецирования $U \in \mathbb{R}^{D \times d}$, произведение которых даёт лучшее приближение исходной матрицы X:

$$\|X - ZU^T\|^2 \to \min_{Z,U}$$

Решением данной задачи также являются собственные векторы ковариационной матрицы.

Второй способ состоит в поиске такого линейного подпространства, что расстояние от исходных объектов до их проекций на это подпространство будет минимальным. В этом случае задача оказывается эквивалентной задаче максимизации дисперсии проекций.

Список литературы

[1] *Bishop, C.M.* Pattern Recognition and Machine Learning. // Springer, 2006.

[2] *Shawe-Taylor, J., Cristianini, N.* Kernel Methods for Pattern Analysis. // Cambridge University Press, 2004.

[3] *Sholkopf, B.A., Smola, A.J.* Learning with kernels. // MIT Press, 2002.

[4] *Micchelli, C.A.* Algebraic aspects of interpolation. // Proceedings of Symposia in Applied Mathematics, 36:81-102, 1986.