Санкт-Петербургский государственный университет

Прикладная математика и информатика

Кафедра статистического моделирования

Обучение с учителем. Метод опорных векторов. Выбор модели с помощью кросс-валидации.

Петраков Михаил
Лунев Иван

Санкт-Петербург
2019

Содержание

1 Введение

Метод опорных векторов по сути является линейным классификатором использующий кусочно-линейную функцию потерь и L_2-регуляризатор. Поэтому мы сначала разберем постановку задачи обучения с учителем, из которой получим постановку задачи обучения линейного классификатора, что даст нам возможность дать определение SVM.

Но на самом деле метод был придуман не из общего вида линейных классификаторов и не из обобщения с функциями потерь и регуляризаторами. Он был придуман из других довольно простых соображений, а именно из соображений построения разделяющей полосы, которые также обсудим.

В конце обсудим выбор модели с помощью кросс-валидации.

2 Обучение с учителем. Постановка задачи

Пусть X — множество объектов, Y — множество ответов, и $\mathsf{y} : \mathsf{X} \to \mathsf{Y}$ — неизвестная зависимость (target function).

Располагаем обучающей выборкой — $(\mathsf{x}_1, \ldots, \mathsf{x}_\mathsf{n}) \subset \mathsf{X}$, где $\mathsf{y}_\mathsf{i} = \mathsf{y}(\mathsf{x}_\mathsf{i}),\ \mathsf{i} = 1, \ldots, \mathsf{n}$ — известные ответы. Требуется найти $\mathsf{a} : \mathsf{X} \to \mathsf{Y}$ — функцию (decision function), приближающую y на всем множестве X.

Вероятностная постановка задачи: имеется неизвестное распределение на множестве $\mathsf{X} \times \mathsf{Y}$ с плотностью $\mathsf{p}(\mathsf{x}, \mathsf{y})$, из которого случайно выбираются $\mathbb{X}_\mathsf{n} = (\mathsf{x}_\mathsf{i}, \mathsf{y}_\mathsf{i})_{\mathsf{i}=1}^{\mathsf{n}}$ (независимые).

Виды задач классификации:

- $\mathsf{Y} = \{-1, +1\}$ — классификация на 2 класса;
- $\mathsf{Y} = \{1, \ldots, K\}$ — классификация на K классов.

3 Задача обучения лин. классификатора

Дано:

- Обучающая выборка $X^n = (x_i, y_i)_{i=1}^n$
- x_i — объекты, векторы из множества $X = \mathbb{R}^n$
- y_i — метки классов, элементы множества $Y = \{-1, 1\}$

Найти:
параметры $\mathsf{w} \in \mathbb{R}^p, \mathsf{w}_0 \in \mathbb{R}$ линейной модели классификации

$$a(x; \mathsf{w}, \mathsf{w}_0) = \operatorname{sign}(< x, \mathsf{w} > - \mathsf{w}_0).$$

Критерий — минимизация :

$$\sum_{i=1}^{n} \mathrm{L}(M_i) \to \min_{\mathsf{w}, \mathsf{w}_0},$$

где $M_i(\mathsf{w}, \mathsf{w}_0) = (< x, \mathsf{w} > - \mathsf{w}_0) y_i$ — отступ (margin) объекта x_i, L — функция потерь.

Часто добавляют регуляризацию:

$$\sum_{i=1}^{n} \mathrm{L}(M_i) + \gamma \mathrm{R}(\mathsf{w}) \to \min_{\mathsf{w},\mathsf{w}_0},$$

где R — регуляризатор, который штрафует неустойчивые решения в случае мультиколлинеарности.

4 Метод опорных векторов

Таким образом, теперь мы можем определить SVM. Метод опорных векторов — это линейный классификатор

$$a(x;\mathsf{w},\mathsf{w}_0) = \mathrm{sign}(< x, \mathsf{w} > - \mathsf{w}_0),$$

использующий кусочно-линейную функцию потерь

$$\mathrm{L}(M_i) = \max\{0, 1 - M_i\} = (1 - M_i)_+,$$

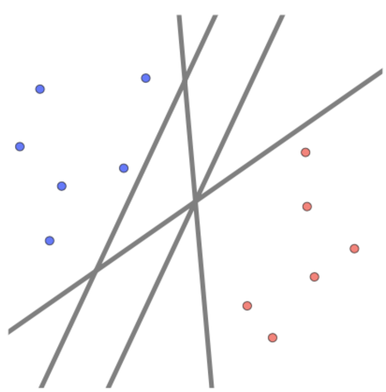
и L_2-регуляризатор:

$$\sum_{i=1}^{n} (1 - M_i)_+ + \gamma ||\,\mathsf{w}\,||^2 \to \min_{\mathsf{w},\mathsf{w}_0}.$$

4.1 Разделяющая полоса в случае линейно разделимой выборки

Теперь обсудим подход объяснения метода опорный векторов через разделяющую полосу.

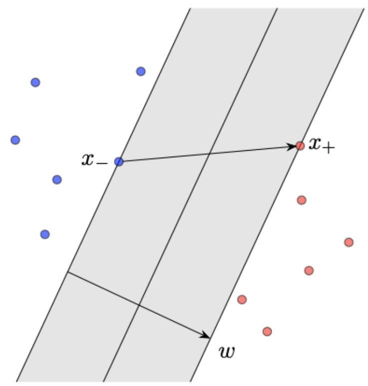
Пусть для простоты рассматривается задача бинарной классификации и некоторая линейно разделимая выборка. Выборка называется линейно разделимой, если в пространстве признаков существует такая гиперплоскость, что объекты разных классов будут находиться по разные стороны от этой плоскости.



Разделяющая гиперплоскость не единственна.

При этом гиперплоскость может быть проведена не единственным образом и возникает задача отыскания оптимальной разделяющей гиперплоскости.

Пусть разделяющая гиперплоскость существует и задается уравнением $\langle \mathsf{w}, x\rangle - \mathsf{w}_0 = 0$. Можно выбрать две параллельные ей и расположенные по разные стороны от нее гиперплоскости так, чтобы между ними не было объектов выборки, а расстояние между ними было максимальным. В таком случае каждая из двух получившихся граничных плоскостей будет «приставлена» к соотвествующему классу.



Разделяющая полоса в случае линейно разделимой выборки

Поскольку уравнение плоскости можно умножать на ненулевое число без изменения соответствующей плоскости, всегда можно выбрать (отнормировать) w и w_0 таким образом, чтобы уравнения граничных плоскостей имели вид:

$$\langle \mathsf{w}, x\rangle - \mathsf{w}_0 = \pm 1.$$

Это условие нормировки можно также сформулировать следующим образом:

$$\min_{i=1,\ldots,n} y_i(\langle \mathsf{w}, x\rangle - w_0) = 1.$$

На каждой из двух граничных плоскостей будет лежать как минимум один объект из соответствующего ей класса (иначе расстояние между плоскостями можно увеличить). Пусть x_+ и x_- — два таких вектора, лежащие на построенных плоскостях и принадлежащие соответствующим классам.

Тогда для ширины разделяющей полосы будет справедливо выражение (как это следует из аналитической геометрии):

$$\left\langle (x_+ - x-), \frac{\mathsf{w}}{||\,\mathsf{w}\,||} \right\rangle = \frac{2}{||\,\mathsf{w}\,||}.$$

Правая часть равенства получена в предположении, что используется описанная выше нормировка.

Теперь можно поставить задачу построения такой разделяющей гиперплоскости, что расстояние между соответствующими ей граничными плоскостями будет максимальным:

$$\begin{cases} < \mathsf{w}, \mathsf{w} > \to \min, \\ y_i(\langle \mathsf{w}, x\rangle - \mathsf{w}_0) \geq 1, i = 1, \ldots, n. \end{cases}$$

4.2 Случай линейно неразделимой выборки

Поскольку в случае линейно неразделимой выборки по определению любой линейный классификатор будет ошибаться, условие $y_i(\langle \mathsf{w}, x\rangle - \mathsf{w}_0) \geq 1$ не может быть выполнено для всех i. Естественным обобщением задачи построения оптимальной гиперплоскости на случай линейно неразделимой выборки является введение ошибок $\xi_i \geq 0$ алгоритма и штрафов за эти ошибки в минимизируемую функцию следующим образом:

$$\begin{cases} \frac{1}{2} < \mathsf{w}, \mathsf{w} > + \sum_{i=1}^{n} \xi_i \to \min_{\mathsf{w}, \mathsf{w}_0, \xi}, \\ y_i(\langle \mathsf{w}, x\rangle - \mathsf{w}_0) \geq 1 - \xi_i, i = 1, \dots, n, \\ \xi_i \geq 0, i = 1, \dots, n. \end{cases}$$

Множитель 1/2 был введен для удобства, а C задает размер штрафа за ошибки.

4.3 Оптимизационная задача для метода опорных векторов

Получившаяся оптимизационная задача:

$$\begin{cases} \frac{1}{2} < \mathsf{w}, \mathsf{w} > + \sum_{i=1}^{n} \xi_i \to \min_{\mathsf{w}, \mathsf{w}_0, \xi}, \\ y_i(\langle \mathsf{w}, x\rangle - \mathsf{w}_0) \geq 1 - \xi_i, i = 1, \dots, n, \\ \xi_i \geq 0, i = 1, \dots, n \end{cases}$$

является оптимизационной задачей в методе опорных векторов (SVM) и непосредственно связана с задачей линейной классификации. Действительно, поскольку $M_i = y_i(\langle \mathsf{w}, x\rangle - \mathsf{w}_0)$ — отступ на i-ом объекте выборки:

$$y_i(\langle \mathsf{w}, x\rangle - \mathsf{w}_0) \geq 1 - \xi_i \Rightarrow \xi_i \geq 1 - M_i.$$

Учитывая также условие $\xi_i \geq 0$, можно получить:

$$\begin{cases} \frac{1}{2} < \mathsf{w}, \mathsf{w} > + \sum_{i=1}^{n} \xi_i \to \min_{\mathsf{w}, \mathsf{w}_0, \xi}, \\ \xi_i \geq \max\{0, 1 - M_i\}, i = 1, \dots, n. \end{cases}$$

При фиксированных w и w_0 задача оптимизации по ξ имеет следующий вид:

$$\sum_{i=1}^{n} \xi_i \to \min, \text{ при условии } \xi_i \geq \max\{0, 1 - M_i\}, i = 1, \dots, n,$$

а ее решением будет $\xi_i = \max\{0, 1 - M_i\} = (1 - M_i)_+$.

Теперь можно вернуться к общей задаче минимизации и переписать ее виде:

$$Q(\mathsf{w}, \mathsf{w}_0) = \sum_{i=1}^{n} (1 - M_i(\mathsf{w}, \mathsf{w}_0))_+ + \frac{1}{2C} || \, \mathsf{w} \, ||^2 \to \min_{\mathsf{w}, \mathsf{w}_0}.$$

Последнее выражение называется безусловной оптимизационной задачей в SVN. В такой формулировке отчетливо видно и функцию потерь, и L^2-регуляризатор.

4.4 Решение задачи минимизации

Ниже представлено решение задачи минимизации.

4.4.1 Напоминание. Условия Каруша-Куна-Таккера

Задача математического программирования:

$$\begin{cases} f(x) \to \min\limits_{x} \\ g_i(x) \leq 0, \quad i = 1, \dots, m; \\ h_j(x) = 0, \quad i = 1, \dots, k; \end{cases}$$

Необходимые условия. Если x — точка локального минимума, то существуют множители μ_i, $i = 1, \dots, k$, λ_j, $j = 1, \dots, k$:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x} = 0, \quad \mathcal{L}(x; \mu, \lambda) = f(x) + \sum_{i=1}^{m} \mu_i g_i(x) + \sum_{j=1}^{k} \lambda_j h_j(x); \\ g_j(x) \leq 0; h_j(x) = 0; \quad \text{(исходные ограничения)} \\ \mu_i \geq 0; \ \text{(двойственные ограничения)} \\ \mu_i g_j(x) = 0. \ \text{(условия дополняющей нежесткости)} \end{cases}$$

4.4.2 Применение условий ККТ к задаче SVM

Функция Лагранжа:

$$\mathcal{L}(\mathsf{w}, \mathsf{w}_0, \xi; \lambda, \mu) = \frac{1}{2} || \mathsf{w} ||^2 - \sum_{i=1}^{n} \lambda_i (M_i(\mathsf{w}, \mathsf{w}_0) - 1) - \sum_{i=1}^{n} \xi_i (\lambda_i + \mu_i - C),$$

λ_i — переменные, двойственные к ограничениям $M_i \geq 1 - \xi_i$;
μ_i — переменные, двойственные к ограничениям $\xi_i \geq 0$.

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathsf{w}} = 0, \frac{\partial \mathcal{L}}{\partial \mathsf{w}_0} = 0, \frac{\partial \mathcal{L}}{\partial \xi} = 0; \\ \xi_i \geq 0, \quad \lambda_i \geq 0, \quad \mu_i \geq 0, \quad i = 1, \dots, n; \\ \lambda_i = 0 \text{ либо } M_i(\mathsf{w}, \mathsf{w}_0) = 1 - \xi_i, \quad i = 1, \dots, n; \\ \mu_i = 0 \text{ либо } \xi_i = 0, \quad i = 1, \dots, n. \end{cases}$$

4.4.3 Необходимые условия седловой точки Лагранжа

Функция Лагранжа:

$$\mathcal{L}(\mathsf{w}, \mathsf{w}_0, \xi; \lambda, \mu) = \frac{1}{2} || \mathsf{w} ||^2 - \sum_{i=1}^{n} \lambda_i (M_i(\mathsf{w}, \mathsf{w}_0) - 1) - \sum_{i=1}^{n} \xi_i (\lambda_i + \mu_i - C),$$

Необходимые условия седловой точки Лагранжа:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathsf{w}} &= \mathsf{w} - \sum_{i=1}^{n} \lambda_i y_i x_i = 0 && \Longrightarrow && w = \sum_{i=1}^{n} \lambda_i y_i x_i;\\
\frac{\partial \mathcal{L}}{\partial \mathsf{w}_0} &= -\sum_{i=1}^{n} \lambda_i y_= 0 && \Longrightarrow && \sum_{i=1}^{n} \lambda_i y_i = 0;\\
\frac{\partial \mathcal{L}}{\partial \xi_i} &= -\lambda_i - \mu_i + C = 0 && \Longrightarrow && \lambda_i + \mu_i = C, \quad i = 1, \dots, n.
\end{aligned}
$$

4.4.4 Понятие опорного вектора

Типизация объектов:

- $\alpha_i = 0$; $\mu_i = C$; $\xi_i = 0$; $M_i \geq 1$ — периферийные (неинформативные) объекты;
- $0 < \alpha_i < C$; $0 < \mu_i < C$; $\xi_i = 0$; $M_i = 1$ — опорные граничные объекты;
- $\alpha_i = C$; $\mu_i = 0$; $\xi_i > 0$; $M_i < 1$ — опорные-нарушители.

Объект x_i называется опорным, если $\lambda_i \neq 0$.

4.4.5 Двойственная задача

$$
\begin{cases}
-\mathcal{L}(\lambda) = -\sum_{i=1}^{n} \lambda_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j < x_i, x_j > \to \min\limits_{\lambda};\\
0 \leq \lambda_i \leq C \quad i = 1, \dots, n;\\
\sum_{i=1}^{n} \lambda_i y_i = 0.
\end{cases}
$$

Решение прямой задачи выражается через решение двойственной:

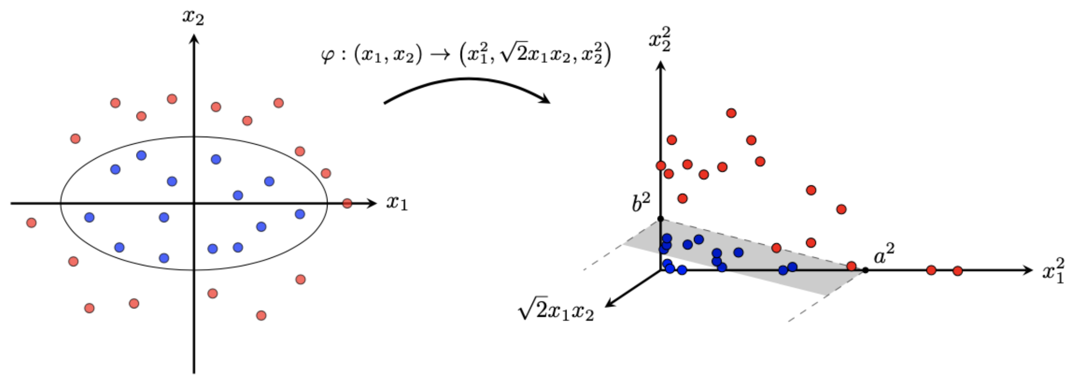
$$
\begin{cases}
\mathsf{w} = \sum_{=1}^{n} \lambda_i y_i x_i;\\
\mathsf{w}_0 =< \mathsf{w}, x_i > -y_i, \text{ для любого i: } \lambda_i > 0, M_i = 1.
\end{cases}
$$

Линейный классификатор:

$$
a(x) = \operatorname{sign}\left(\sum_{i=1}^{n} \lambda_i y_i < x_i, x > - \mathsf{w}_0 \right).
$$

5 Ядра в методе опорных векторов (Kernel trick)

Пока ансамбли решающих деревьев не набрали своей популярности, SVM очень часто использовали даже в тех задачах, где разделяющая поверхность не похожа на линейную.



Пример построения спрямляющего пространства.

5.1 Добавление новых признаков и kernel trick

Чтобы применять SVN в нелинейном случае, строилось спрямляющее пространство. В основе этого лежит очень простая и очень красивая идея: если в каком-то исходном пространстве признаков классы не являются линейно разделимыми, то может быть можно отобразить это пространство признаков в какое-то новое, в котором классы уже будут линейно разделимы.

Не обязательно задавать это отображение явно, так как в SVM везде фигурирует только скалярное про- изведение вида $\langle \mathsf{w}, x\rangle$.

Пусть $\phi(x)$ — спрямляющее отображение, тогда, чтобы записать SVM в спрямляющем пространстве, необходимо во всех формулах сделать следующие подстановки:

$$x \to \phi(x), \quad w \to \phi(\mathsf{w}), \quad \langle \mathsf{w}, x\rangle \to \langle \phi(\mathsf{w}), \phi(x)\rangle.$$

Тогда метод SVM может быть сформулирован в исходном пространстве, если в качестве скалярного произведения использовать, возможно, нелинейную симметричную функцию

$$K(\mathsf{w}, x) = \langle \phi(\mathsf{w}), \phi(x)\rangle$$

и таким образом получать нелинейную разделяющую поверхность. Эта идея называется в англоязычной литературе kernel trick.

5.2 Линейное ядро

В простейшем случае ядро совпадает со скалярным произведением:

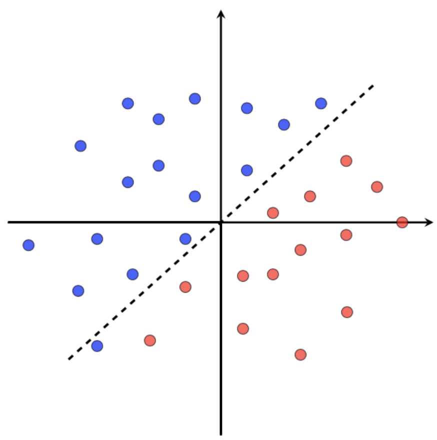
$$K(\mathsf{w}, x) = \langle \mathsf{w}, x\rangle.$$

Следует отметить, что линейное ядро в некоторых задачах — самый лучший выбор, например в задачах классификации текстов, и не стоит выкидывать его из рассмотрения.

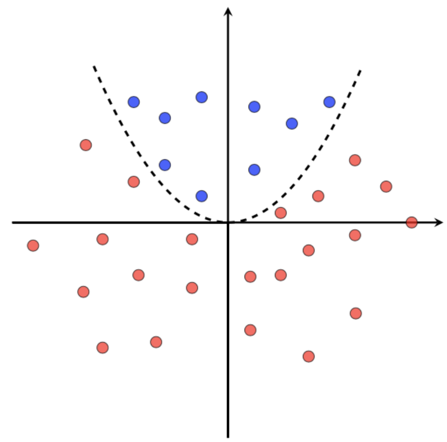
5.3 Полиномиальное ядро

Другой пример — это полиномиальное ядро:

$$K(\mathsf{w}, x) = (\langle \mathsf{w}, x\rangle + r)^d.$$



Линейное ядро



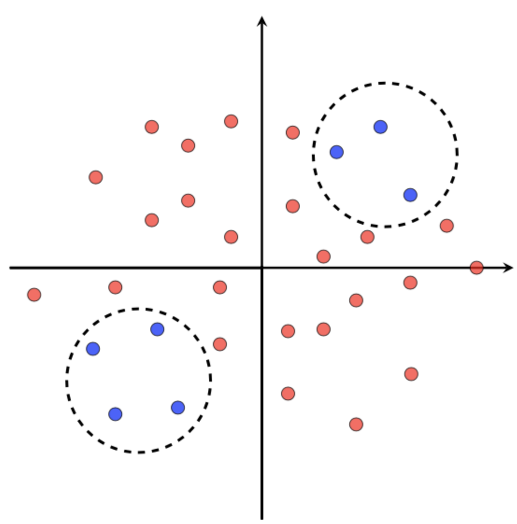
Полиномиальное ядро

Полиномиальное ядро получается, если в качестве спрямляемого пространство выступает пространство многочленов не выше определенной степени.

5.4 Радиальное ядро

И другое часто используемое ядро — это радиальное ядро:

$$K(\mathsf{w}, x) = \exp(-\gamma || \mathsf{w} - x||^2).$$



Радиальное ядро

Поскольку радиальное ядро выражается через евклидово расстояние, будут проявляться основные проблемы метрических алгоритмов, в том числе проклятие размерности.

Именно поэтому не стоит применять это ядро, если признаков действительно очень-очень много.

Но так или иначе, оно позволяет строить очень сложные границы классов. Спрямляющее пространство, которое соответствует данному ядру, является бесконечномерным.

6 Мультиклассовая SVM

Для обобщения метода на случай нескольких классов используют одну из двух стратегий: «one-against-one» или «one-vs-th-rest». Пусть N — число классов. Тогда в первом случае будет построено $N(N-1)/2$ классификаторов, каждый из которых натренирован лишь на двух классах. Такую стратегию использует метод svm.SVC() из библиотеки scikit-learn. Во втором случае очевидно строится N классификаторов, каждый из которых отделяет один класс от всех остальных. Такая стратегия реализована в методе svm.LinearSVC().

7 Cross-validation

Дано:

- Имеется выборка $(X_n,\, Y_n)$;
- Умеем строить модель, зависящую от параметра θ и минимизирующую ошибку $J(X_n, Y_n; \theta, \lambda)$, где λ — параметр регуляризации;

Хотим подобрать такой параметр θ_0, чтобы минимизировать ошибку $J(X_{new}, Y_{new}; \theta_0, 0)$ на новых индивидах.

Алгоритм:

- Делим выборку $(X_n,\, Y_n)$ случайным образом на три набора: $(X_{train},\, Y_{train})$, $(X_{CV},\, Y_{CV})$ и $(X_{test},\, Y_{test})$;
- Перебираем набор параметров $\lambda_1, \ldots, \lambda_m$;
- Для каждого параметра λ_i строим модель на $(\mathsf{X}_{train},\, Y_{train})$ (то есть находим оптимальное θ_{i0}) и считаем ошибку на $J(X_{CV}, Y_{CV}; \theta_{i0}, 0)$;
- Берем λ_0 с минимальной ошибкой (ему соответствует θ_0);
- Считаем ошибку модели $J(X_{test}, Y_{test}; \theta_0, 0)$.

8 K-fold Cross-validation

Основные условия, как в предыдущей секции.

Алгоритм:

- Делим выборку $(X_n,\, Y_n)$ случайным образом на K частей: $(X_1,\, Y_1)$,...,$(X_K,\, Y_K)$;
- Обозначим за $(X_k',\, Y_k')$ набор, содержащий всех индивидов, кроме $(X_k,\, Y_k)$;
- Перебираем набор параметров $\lambda_1, \ldots, \lambda_m$;

- Для каждого параметра λ_i считаем

$$CV_i = \sum_{j=1}^{K} \frac{n_j}{n} J(X_j, Y_j; \theta_j, 0),$$

где θ_j минимизирует $J(X_j', Y_j'; \theta, \lambda_i)$, n_j — число индивидов в (X_j, Y_j);

- Берем λ_0 с минимальной ошибкой CV_i;

- Берем θ_0, которое минимизирует $J(X_n, Y_n; \theta, \lambda_0)$.