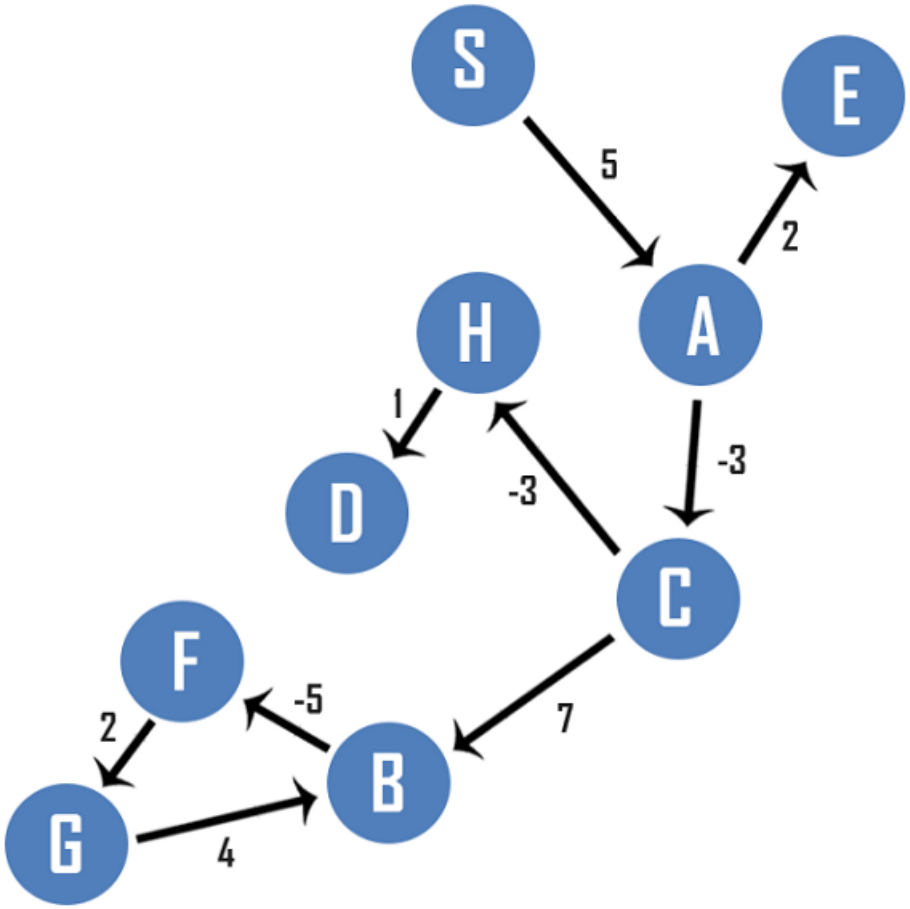
Наглядное объяснение алгоритма Беллмана-Форда

28.08.2020

Алгоритм Беллмана-Форда находит в ориентированном графе кратчайшие пути от исходной вершины до всех остальных. В отличие от алгоритма Дейкстры, в алгоритме Беллмана-Форда могут быть рёбра с отрицательным весом.



Начнём с того, что все исходящие ребра записываются в таблице в алфавитном порядке.

Edge	A-C	A-E	B-F	C-B	C-H	F-G	G-B	H-D	S-A
Weight	-3	2	-5	7	-3	2	4	1	5

Как и в алгоритме Дейкстры, создаётся таблица, в которой записывается расстояние до каждой вершины и предыдущая вершина. Расстояния для всех вершин, кроме исходной, инициализируются значением бесконечности. Расстояние обозначено переменной d, а предыдущая вершина—переменной п.

	S	A	B	C	D	E	F	G	H
D	0	∞	∞	∞	∞	∞	∞	∞	∞
П	-	-	-	-	-	-	-	-	-

Алгоритм Беллмана-Форда проходит итеративный путь через все рёбра. Всего рёбер 9, поэтому путь этот будет насчитывать до 9 итераций. Во время каждой итерации какое-то одно ребро ослабляется. Во время первой итерации при переходе из вершины A в вершину C вес соответствующего ребра AC составляет -3. Текущее расстояние от исходной вершины до A равно бесконечности. При добавлении -3 к бесконечности результатом будет бесконечность, поэтому значение C будет равно бесконечности. Аналогично, при переходе от A до E вес соответствующего ребра AE составляет 2. Но,

так как расстояние до A бесконечно, значение E будет равно бесконечности. То же верно и в отношении рёбер BF, CB, CH, FG, GB и HD. Все они дают один и тот же результат—бесконечность. И только последнее ребро, SA, даёт другой результат. Вес ребра SA равен 5. Текущее расстояние до S равно 0, поэтому расстояние от S до A составит 0 + 5 = 5. Вершина, предыдущая по отношению к вершине A,—это вершина S. После первой итерации алгоритм Беллмана-Форда нашёл путь от S до A.

Iteration 1	S	A	B	C	D	E	F	G	H
d	0	5	∞	∞	∞	∞	∞	∞	∞
π	-	S	-	-	-	-	-	-	-

Так как все рёбра были ослаблены, алгоритм начинает вторую итерацию. Нам важны рёбра, отходящие от S и A, поскольку расстояние до этих вершин уже известно. Расстояние до всех остальных вершин равно бесконечности. Обращаясь к таблице, содержащей рёбра, мы начинаем с ослабления ребра AC.

Edge	A-C	A-E	B-F	C-B	C-H	F-G	G-B	H-D	S-A
Weight	-3	2	-5	7	-3	2	4	1	5

Вес ребра AC равен -3. Текущее расстояние до вершины A равно 5 (через ребро SA), поэтому расстояние до вершины C составит 5 + (-3) = 2. Вершина-предшественница C —это вершина A. Вес ребра AE равен 2. Расстояние до E составит 5 + 2 = 7 (через ребро SA). Вершиной, предыдущей по отношению к вершине E, становится теперь вершина A. Ребро BF пока ещё не может быть ослаблено.

Iteration 2	S	A	B	C	D	E	F	G	H
d	0	5	∞	2	∞	7	∞	∞	∞
π	-	S	-	A	-	A	-	-	-

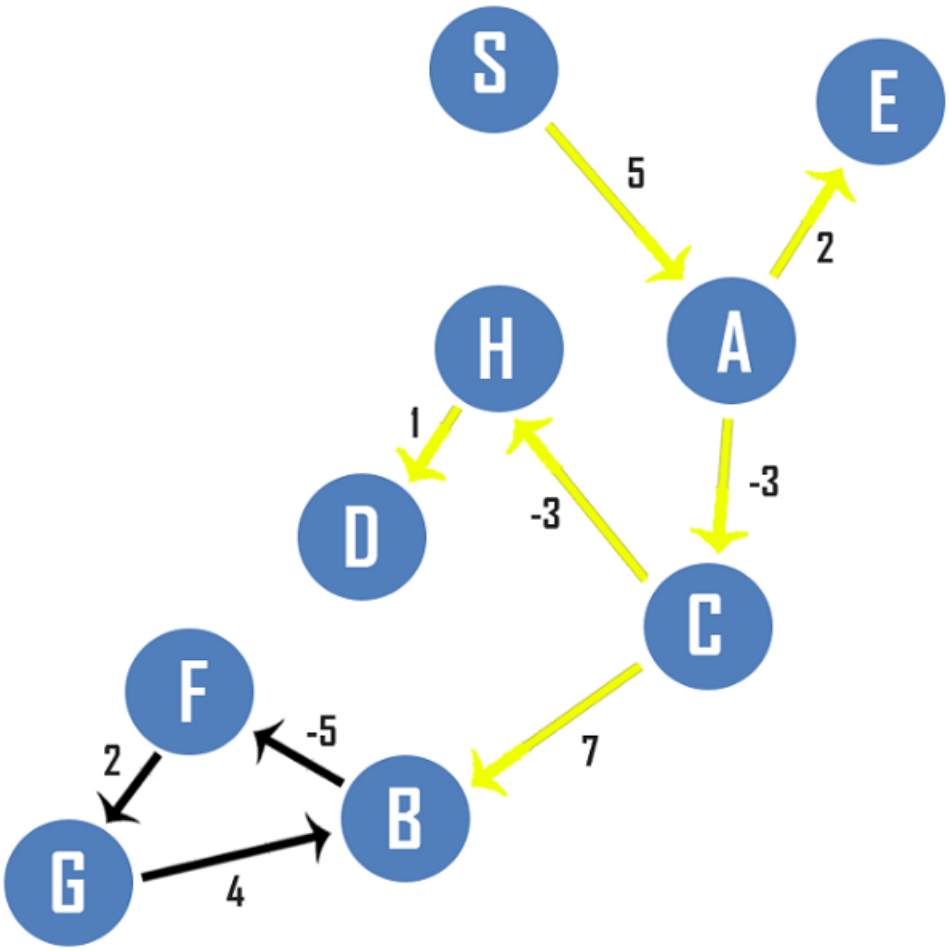
Ребро CB может быть ослаблено, так как мы знаем расстояние до C. Расстояние до B составит 2 + 7 = 9, а вершина, предыдущая по отношению к вершине B,—это вершина C. Ребро CH может быть ослаблено, так как мы знаем расстояние до C. Расстояние до H составит 2 + (-3) = -1, а вершина-предшественница H—это вершина C.

Iteration 2	S	A	B	C	D	E	F	G	H
d	0	5	9	2	∞	7	∞	∞	-1
π	-	S	C	A	-	A	-	-	C

Ребро FG ещё не может быть ослаблено. Ребро GB тоже не может быть ослаблено. А вот ребро HD можно ослабить, так как мы знаем, что расстояние до вершины H равно -1. Расстояние до вершины D составит -1 + 1 = 0, а вершина, предыдущая по отношению к вершине D,—это вершина H.

Iteration 2	S	A	B	C	D	E	F	G	H
d	0	5	9	2	0	7	∞	∞	-1
π	-	S	C	A	H	A	-	-	C

Расстояние до A от ребра SA уже посчитано и равно 5, так что никаких обновлений здесь не требуется. На этом заканчивается итерация 2.



Edge	A-C	A-E	B-F	C-B	C-H	F-G	G-B	H-D	S-A
Weight	-3	2	-5	7	-3	2	4	1	5

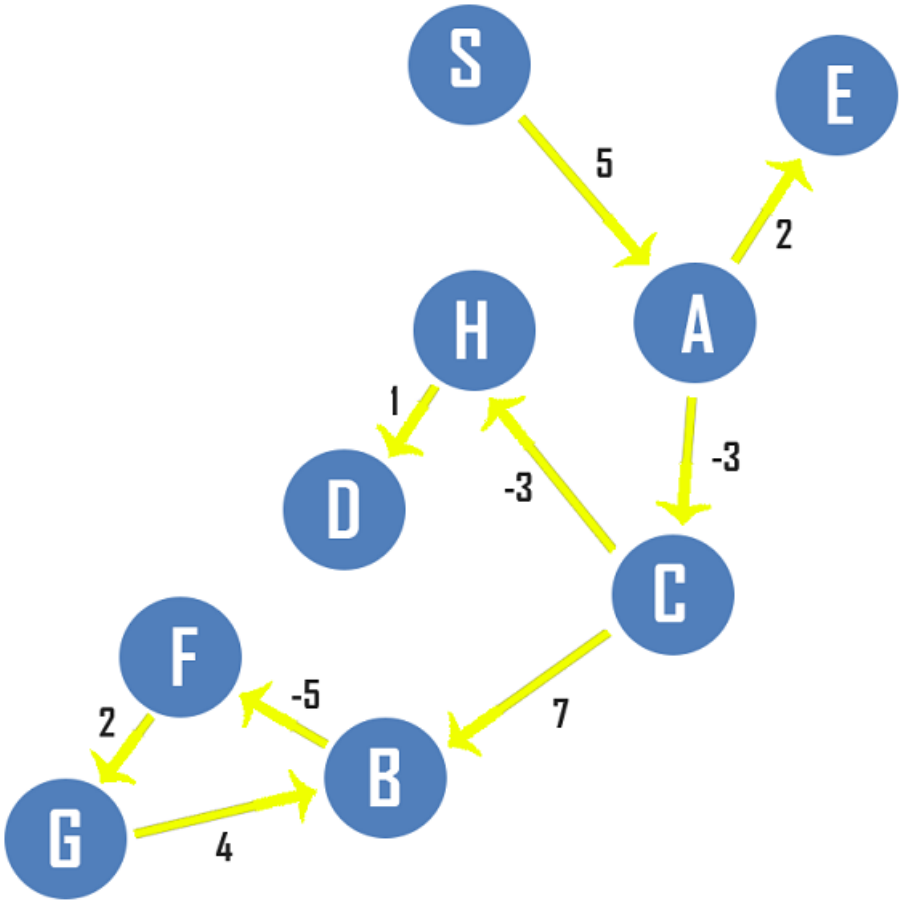
Во время третьей итерации алгоритм Беллмана-Форда снова перебирает все рёбра. Ребра AC и AE дают одинаковые результаты. Ребро BF теперь можно ослабить. Расстояние до B равно 9, поэтому расстояние до вершины F составит 9 + (-5) = 4. Вершина-предшественница F—это вершина B.

Iteration 3	S	A	B	C	D	E	F	G	H
d	0	5	9	2	0	7	4	∞	-1
π	-	S	C	A	H	A	B	-	C

Рёбра CB и CH дают одинаковые результаты, поэтому таблица остаётся прежней. Ребро FG теперь можно ослабить. Расстояние до вершины F равно 4, поэтому расстояние до вершины G составит 4 + 2 = 6. Вершина, предыдущая по отношению к вершине G,— это вершина F.

Iteration 3	S	A	B	C	D	E	F	G	H
d	0	5	9	2	0	7	4	6	-1
π	-	S	C	A	H	A	B	F	C

Ребро GB теперь можно ослабить. Расстояние до вершины G равно 6, поэтому расстояние до B составит 6 + 4 = 10. Так как до вершины B можно добраться более коротким путём (через ребро CB), таблица остаётся прежней.

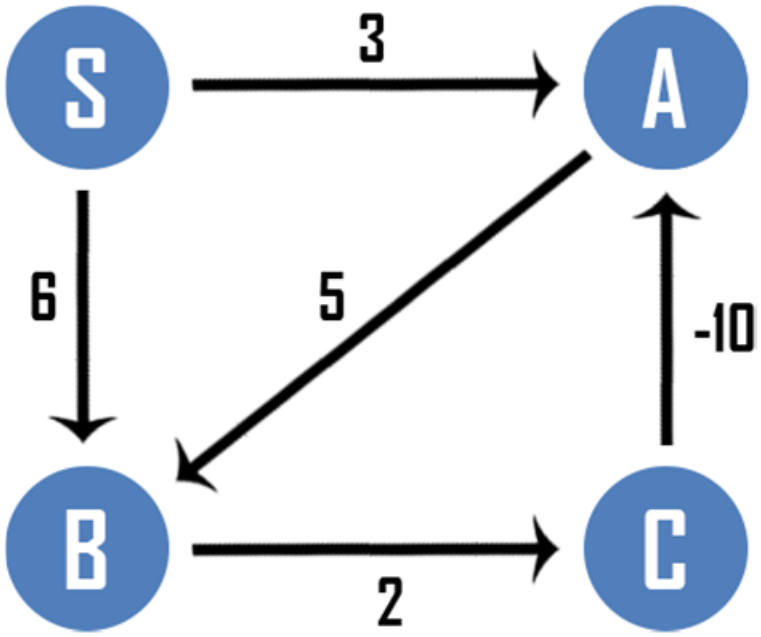


Edge	A-C	A-E	B-F	C-B	C-H	F-G	G-B	H-D	S-A
Weight	-3	2	-5	7	-3	2	4	1	5

Во время четвёртой итерации перебираются все рёбра. Алгоритм видит, что изменений нет, поэтому на четвёртой итерации он завершается.

Если бы в этом графе существовал отрицательный цикл, то после n-1 итераций алгоритм Беллмана-Форда теоретически должен был бы найти кратчайшие пути ко всем вершинам. Во время n-й итерации, где n обозначает число вершин, при существовании

отрицательного цикла расстояние, по крайней мере, до одной вершины изменится. Давайте рассмотрим небольшой пример.



Edge	A–B	B–C	C–A	S–A	S–B
Weight	5	2	-10	3	6

Строится таблица с расстояниями и вершинами-предшественницами. Расстояния инициализируются значением бесконечности для вершин A, B и C. Расстояние до S равно 0.

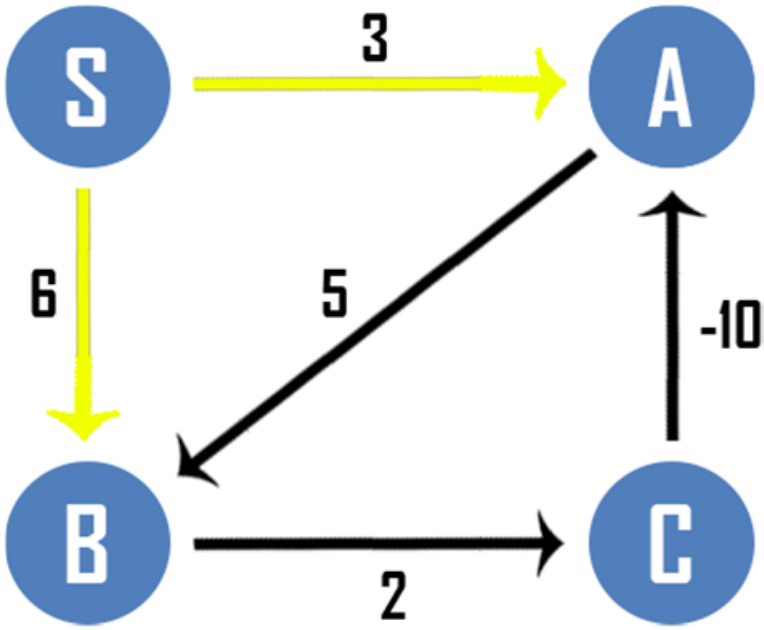
	S	A	B	C
D	0	∞	∞	∞
П	-	-	-	-

Видим, что первое ребро AB ещё не может быть ослаблено, так же как и рёбра BC и CA. Зато может быть ослаблено ребро SA. Расстояние до S равно 0, поэтому расстояние до A составит 0 + 3 = 3. Вершина, предыдущая по отношению к вершине A,—это вершина S.

Iteration 1	S	A	B	C
D	0	3	∞	∞
П	-	S	-	-

Ребро SB тоже можно ослабить. Расстояние до вершины B составит 0 + 6 = 6. Вершина-предшественница B—это вершина S.

Iteration 1	S	A	B	C
d	0	3	6	∞
π	-	S	S	-

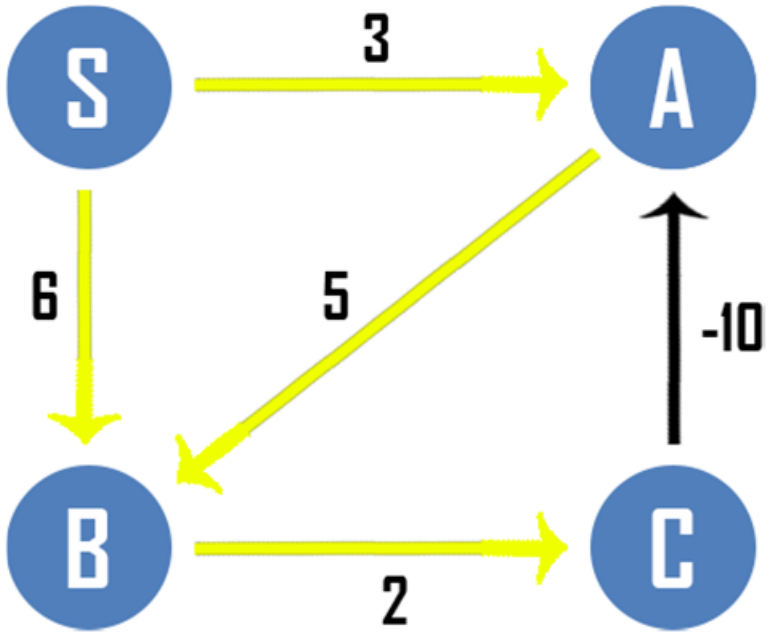


Первая итерация завершена. Во время второй итерации снова перебираются все рёбра.

Edge	A-B	B-C	C-A	S-A	S-B
Weight	5	2	-10	3	6

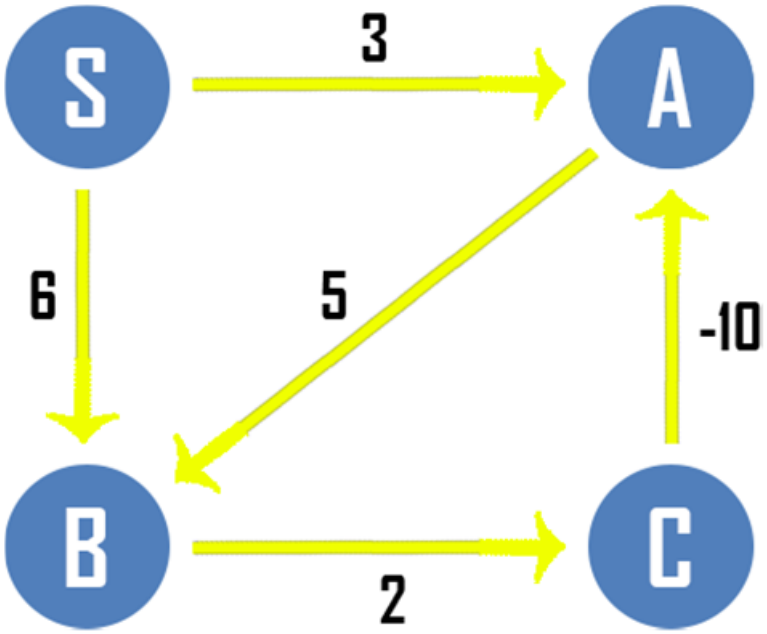
Ребро AB может быть ослаблено во время второй итерации. Расстояние до A равно 3, поэтому расстояние до вершины B составит 3 + 5 = 8. Так как расстояние до B уже меньше нового значения, значение B сохраняется. До ребра BC можно добраться, пройдя расстояние 6 + 2 = 8. Вершина, предыдущая по отношению к вершине C,—это вершина B.

Iteration 2	S	A	B	C
d	0	3	6	8
π	-	S	S	B



Дальше перебирается ребро CA. Расстояние до C равно 8 единицам, поэтому расстояние до A (через ребро BC) составит 8 + (-10) = -2. Так как расстояние до A через ребро CA меньше, чем через ребро SA, расстояние до A обновляется.

Iteration 2	S	A	B	C
D	0	-2	6	8
П	-	C	S	B



Рёбра SA и SB не дают ничего лучшего, поэтому вторая итерация завершена. Начинается третья итерация.

Edge	A-B	B-C	C-A	S-A	S-B
Weight	5	2	-10	3	6

Ребро AB ослаблено. Расстояние до A в настоящее время равно -2, поэтому расстояние до B через ребро AB составит -2 + 5 = 3. Так как расстояние до B через AB меньше, чем через SB, расстояние становится теперь равным 3. Вершиной, предыдущей по отношению к вершине B, становится теперь вершина A.

Iteration 3	S	A	B	C
d	0	-2	3	8
π	-	C	A	B

Дальше ослабляется ребро BC. Текущее расстояние до B равно 3, поэтому расстояние до C составит 3 + 2 = 5. Расстояние до C становится теперь равным 5.

Iteration 3	S	A	B	C
d	0	-2	3	5
π	-	C	A	B

Ребро CA ослабляется. Расстояние до C составит 5 + (-10) = -5. Расстояние до вершины A обновляется на -5 единиц.

Iteration 3	S	A	B	C
d	0	-5	3	5
π	-	C	A	B

Рёбра SA и SB не дают результатов лучше. В это время все кратчайшие пути должны были быть найдены. В других итерациях никаких изменений быть не должно.

Edge	A-B	B-C	C-A	S-A	S-B
Weight	5	2	-10	3	6

Ребро AB ослаблено. Расстояние до A равно -5, поэтому расстояние до B составит -5 + 5 = 0. Расстояние до B становится теперь равным 0. Так как значение меняется на n-й итерации, значения будут меняться и на n+1-й итерации. Значения будут продолжать меняться неопределённое время. Если мы внимательно изучим граф, то увидим, что ABC даёт отрицательное значение: 5 + 2 + (-10) = -3.

Читайте также:

- Обзор шаблонов SnapML и их возможностей в Lens Studio
- Графы: основы теории, алгоритмы поиска
- Почему мы создали платформу для инженерии машинного обучения, а не науки о данных

Читайте нас в Telegram, VK и Яндекс.Дзен

Перевод статьи Dino Cajic: Bellman-Ford Algorithm Visually Explained