17. Способы борьбы с переобучением. Байесовская сеть доверия: определение и применение. Уменьшение размерности пространства признаков.

http://logic.pdmi.ras.ru/csclub/sites/default/files/slides/20080330_machine_learning_nikolenko_lecture07.pdf

http://www.habarov.spb.ru/new_es/exp_sys/es06/es6.htm

Способы борьбы с переобучением

Минимизацию эмпирического риска следует применять с известной долей осторожности. Если минимум функционала $Q(a, X^l)$ достигается на алгоритме a, то это ещё не гарантирует, что a будет хорошо приближать целевую зависимость на произвольной контрольной выборке $X^k = (x_i, y_i)_{i=1}^k$. Когда качество работы алгоритма на новых объектах, не вошедших в состав обучения, оказывается существенно хуже, чем на обучающей выборке, говорят об эффекте переобучения (overtraining).

Избавиться от него нельзя. Как его минимизировать?

- минимизировать одну из теоретических оценок;
- накладывать ограничения на Σ (регуляризация);
- минимизировать HoldOut, LOO или CV

Читать: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf

http://www.machinelearning.ru/wiki/images/2/2d/Voron-ML-Modeling.pdf

Байесовская сеть доверия

Байесовская сеть – графическая вероятностная модель, представляющая собой множество переменных и их вероятностных зависимостей. Является в некотором роде продолжением байесовского классификатора. Б.К. основывается на предположении об условной независимости атрибутов при условии данного целевого значения. Б.С. представляет собой направленный граф, в котором стрелки показывают причинно-следственную связь. В вершинах графа заданы условные вероятности при условии всего множества предков. Если предков нет, вероятности

не условные, а маргинальные. В графе запрещены направленные циклы. Вся эта информация дает возможность вычислять любую вероятность в сети, т.е. единственным образом задает распределение.

`http://habrahabr.ru/company/surfingbird/blog/176461/`

Суть рассуждений в байсовской сети – пропагация свидетельств. Обычно пропагация идёт снизу вверх, от следствий к причинам.

Теорема о декомпозиции. Для БСД общее распределение вероятностей $p(X) = p(x_1, \ldots, x_n) = \prod_{x \in X} p(x|pa(x))$, где $pa(x)$ – множество родителей узла x в графе.

Маргинальные, совместные и условные распределения являются факторами (factor) – функциями от нескольких переменных. Над факторами можно производить некоторые операции: перемножать (multiply), маргинализировать по переменной(marginalize) и уменьшать (reduce).

0.1 Variable elimination

Совместное распределение в сети задается через перемножение нескольких факторов, соответствующих вершинам. По входным свидетельствам хотим получить апостериорные вероятности событий в сети.

1. Если есть свидетельства, делаем редукцию факторов по ним;

2. Выбираем событие, содержащееся в наименьшем числе факторов;

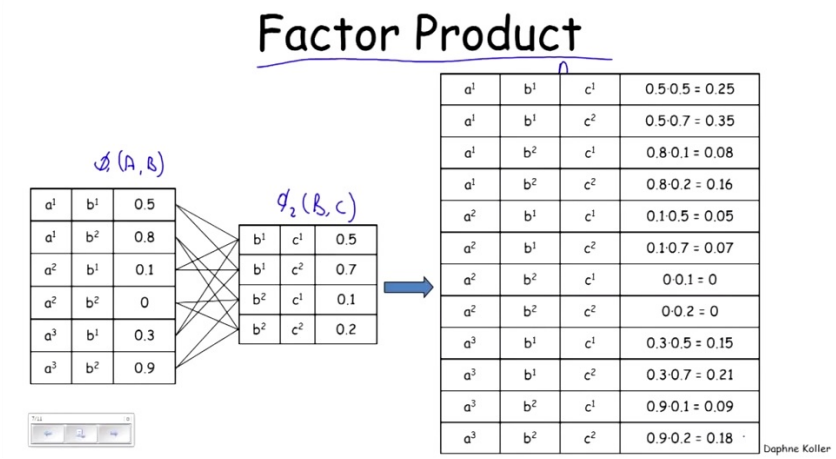


Рис. 1: Произведение факторов

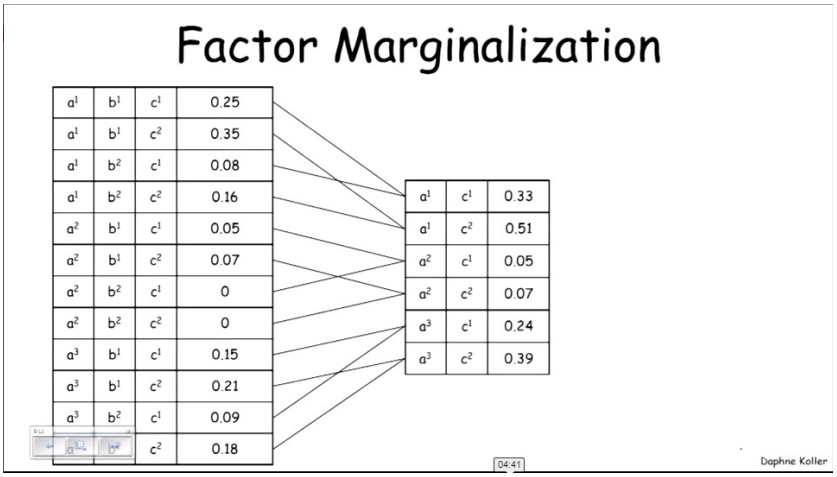


Рис. 2: Маргинализация

3. Перемножаем и нормируем полученный фактор;
4. Маргинализуем по выбранному событию;
5. Если остались события, по которым суммирование еще не делали, возвращаемся на шаг 2.

Уменьшение размерности пространства признаков

`http://logic.pdmi.ras.ru/~sergey/teaching/mlauii12/15-pca.pdf`

Factor Reduction

a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

a^1	b^1	c^1	0.25
a^1	b^2	c^1	0.08
a^2	b^1	c^1	0.05
a^2	b^2	c^1	0
a^3	b^1	c^1	0.15
a^3	b^2	c^1	0.09

Daphne Koller

Рис. 3: Редукция

Variable Elimination

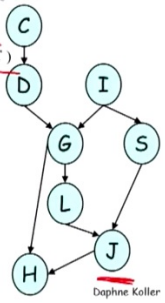
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L

$$\sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$

$$\sum_{I,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\sum_C \phi_D(C,D)\phi_C(C)$$

$\tau_1(D)$

Compute $\tau_1(D) = \sum_C \phi_C(C)\phi_D(C,D)$



Daphne Koller

Рис. 4: Variable elimination

Метод главных компонент – один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации.

1. Нормируем матрицу $X(d \times n)$ построчно(мат.ожидание 0, дисперсия 1);
2. Строим матрицу ковариаций $\Sigma = XX^T$;
3. Вычисляем собственные значения;
4. Берем собственные вектора, соответстующие k самым большим собственным значениям, и формируем из них матрицу W.

5. Перемножаем полученную матрицу с исходной $Y = WX$. Y имеет размерность $k \times n$

Выбирать k можно разными способами:

- k = $\sum_i [\lambda_i > \varepsilon]$ То есть берем все вектора, собственные значения которых больше ε.
- k = $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > 0.9$

Суть метода в нахождении проекции, в которой максимизируется дисперсия и минимизируется суммарное расстояние до проекций точек. Оказывается, что необходимо делать проекцию на пространство, базисом в котором являются собственные вектора. При этом некоторые вектора (с малым собственным значением) можно отбросить, тем самым уменьшив размерность пространства и потеряв минимум информации. PCA позволяет получить матрицу перехода в новое пространство, обнаружить зависимости между признаками исходных данных и сделать некоторый препроцессинг данных (после которого могут быть лучше видны характерные особенности).