Determining Probabilities of Handwriting Formations using PGMs

Alina Vereshchaka
Department of Computer Science and Engineering
State University of New York at Buffalo
306 Davis Hall, Buffalo, NY, 14228
avereshc@buffalo.edu

1 Task 1

For this task we needed to evaluate pairwise correlations and independences that exist in the data. Thus if $p(x, y) \approx p(x)p(y)$ that means that the data is independent, otherwise we can estimate how much one data is correlated to another. The approximate correlation was calculated using Eq. 1.

$$\sum \Big| ((P(x, y) - P(x)P(y)) \Big| \tag{1}$$

Thus, the higher the parameter, the higher correlation between the pair. Results of the pairwise features correlation are in Table 1.

Table 1: Pairwise correlations of the features

Pair	Correlation
x1x2	0.1598
x1x4	0.1194
x1x6	0.1601
x2x3	0.2185
x2x5	0.1293
x3x2	0.2187
x3x5	0.1155
x3x6	0.1132
x4x1	0.1195
x4x2	0.1157
x4x6	0.1435
x5x2	0.1293
x5x3	0.1160
x6x1	0.1604
x6x2	0.1753
x6x3	0.0943
x6x4	0.1431

As we can see the highest correlation is between height relationship of t to h (x1) and shape of loop of h (x2) features, while the lowers is between shape of t (x6) and shape of arch of h (x3).

2 Task 2

For this task we need to construct a Bayesian network with the fewest number of edges that maximizes the likelihood. The threshold was set to 1.12 in order to include all the features to the final graph.

Table 2: Pairs with correlations > 0.12

Pair	Correlation
x1x2	0.1598
x1x6	0.1601
x2x3	0.2185
x2x5	0.1293
x3x2	0.2187
x4x6	0.1435
x5x2	0.1293
x6x1	0.1604
x6x2	0.1753
x6x4	0.1431

I begin constructing the Bayesian network with the most correlated pairs, by increasing the number of edges and trying various directions. To estimate the constructed network, I generate samples using ancestral sampling and applied K2 algorithm, a well-known score-based algorithm to estimate Bayesian network (Eq.2). It recovers the underlying distribution in the form of DAG efficiently.

Some of the helpful equations used to construct and estimate BNs.

Bayes Rule:

$$p(x1|x2) = \frac{p(x2|x1)p(x1)}{p(x2)}$$

K2 Bayseian scoring function:

$$g_{K2}(G:D) = log(p(G)) + \sum_{i=1}^{n}\left[\sum_{j=1}^{q_i}\left[log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\right) + \sum_{k=1}^{r_i} log(N_{ijk}!)\right]\right]$$

, where $p(G)$ represents the prior probability of the DAG G.

The full list of constructed BNs and their evaluation is presented in Table 3, the image representation is on Fig. 1.

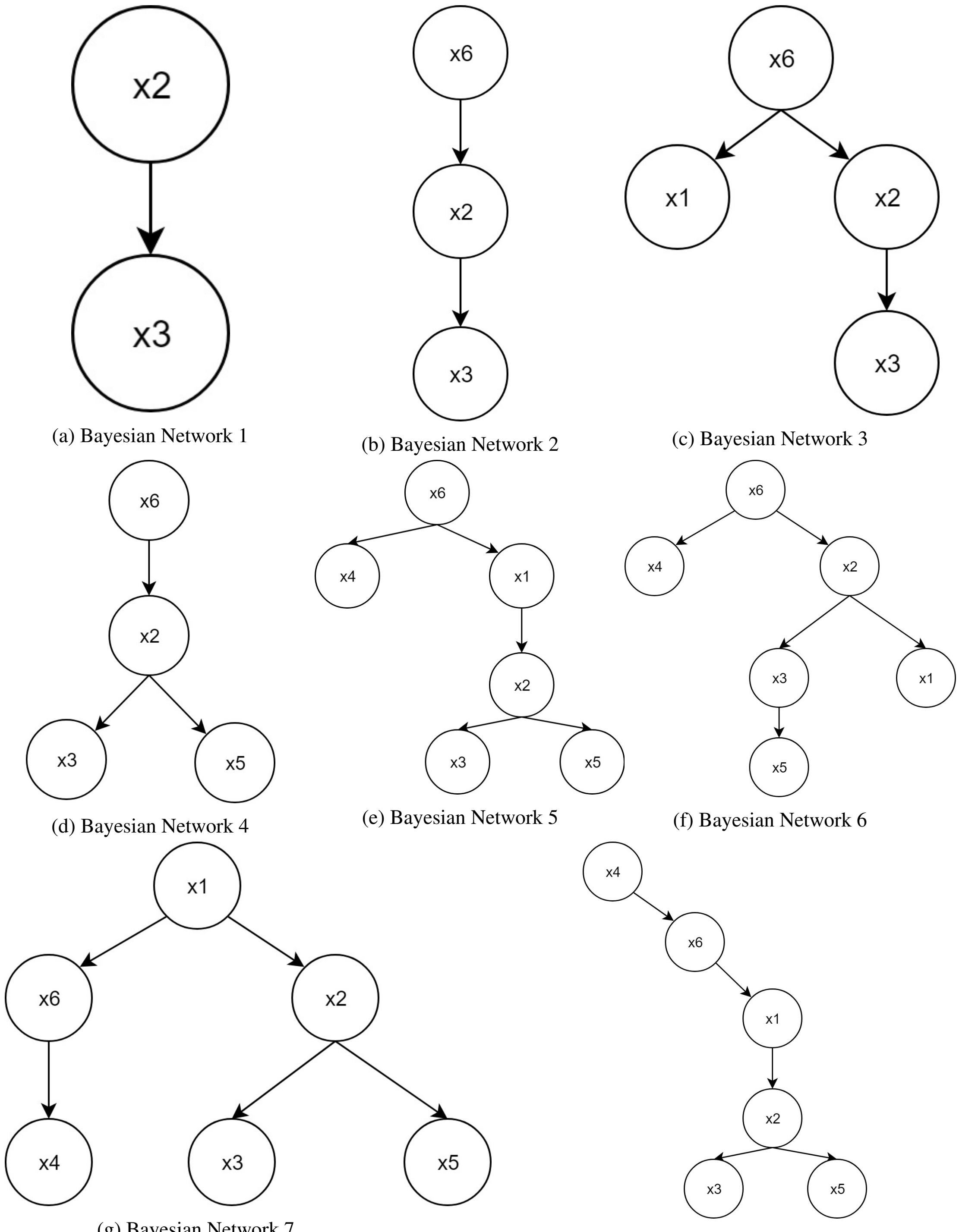
Table 3: Constructed Bayesian Networks evaluation

Bayesian Network	Edges	K2Score
1	('x2', 'x3')	-2304
2	('x6', 'x2'), ('x2', 'x3')	-3641
3	('x6', 'x2'), ('x6', 'x1'), ('x2', 'x3')	-4399
4	('x6', 'x2'), ('x2', 'x3'), ('x2', 'x5')	-4863
5	('x6', 'x4'), ('x6', 'x1'), ('x1', 'x2'), ('x2', 'x3'), ('x2', 'x5')	-6465
6	('x6', 'x4'), ('x6', 'x2'), ('x2', 'x3'), ('x2', 'x1'), ('x3', 'x5')	-6429
7	('x1', 'x6'), ('x1', 'x2'), ('x6', 'x4'), ('x2', 'x3'), ('x2', 'x5')	-6481
8	('x4', 'x6'), ('x6', 'x1'), ('x1', 'x2'), ('x2', 'x3'), ('x2', 'x5')	-6461

3 Task 3

Bayesian Network 7 was converted to Markov network using pgmpy fucntion **to_markov_model()**. Obtained Markov network:

('x1', 'x6'), ('x1', 'x2'), ('x6', 'x4'), ('x2', 'x3'), ('x2', 'x5')



(a) Bayesian Network 1
(b) Bayesian Network 2
(c) Bayesian Network 3
(d) Bayesian Network 4
(e) Bayesian Network 5
(f) Bayesian Network 6
(g) Bayesian Network 7
(h) Bayesian Network 8

Figure 1: Image representations of constructed Bayesian networks

4 Task 4

For this task, using the "and" image dataset we needed to construct a Bayesian network and evaluate the goodness score of several Bayesian networks. The data has 8 features and 1025 samples. The Bayesian Network was constructed using Hill Climb Search algorithms using pgmpy liblary in Python. This algorithm performs local hill climb search to estimates the BayesianModel structure that has optimal score, according to the scoring method supplied in the constructor.

4.1 Constructing Bayesian networks using BicScore

Using BicScore as a scoring method for constructing Bayesian network, the algorithm returned the best network, given the data.

Nodes used in the network:

'f1', 'f2', 'f3', 'f4', 'f5', 'f6', 'f7', 'f8', 'f9'

Edges:

[('f3', 'f5'), ('f4', 'f3'), ('f8', 'f9'), ('f9', 'f1'), ('f9', 'f5'), ('f9', 'f6'), ('f9', 'f7')]

Network, where all nodes have at most 1 parents:

[('f4', 'f3'), ('f8', 'f9'), ('f9', 'f5'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6')]

Network, where all nodes have at most 2 parents:

[('f3', 'f5'), ('f4', 'f3'), ('f8', 'f9'), ('f9', 'f5'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6')]

K2Score

-8175.27

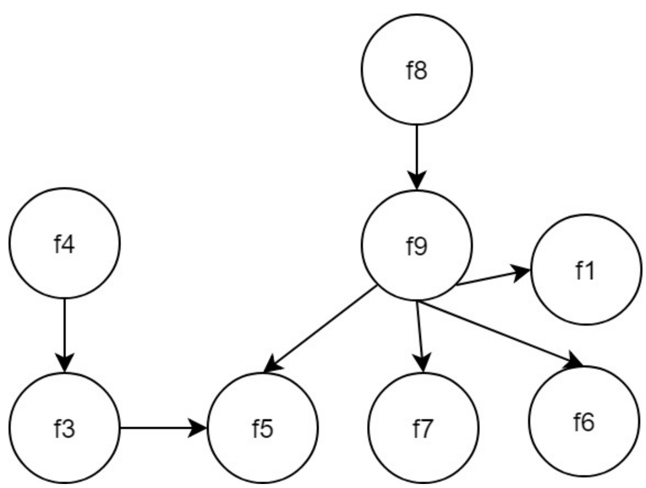


Figure 2: Best Bayesian Network for "and" image dataset using BicScore

MLE for all features

Below are the maximum likelihood estimation for each fearure, based on the Bayesian network presented on Fig. 2.

f9	f9(1)	f9(2)
f1(0)	0.19823788546255505	0.10638297872340426
f1(1)	0.44933920704845814	0.26032540675844806
f1(2)	0.2643171806167401	0.3692115143929912
f1(3)	0.0881057268722467	0.2640801001251564

Figure 3: MLE for feature f1

f4	f4(0)	f4(1)	f4(2)	f4(3)	f4(4)
f3(0)	0.061971830985915494	0.1415525114155251	0.5555555555555556	0.05063291139240506	0.10714285714285714
f3(1)	0.9323943661971831	0.8264840182648402	0.4351851851851852	0.9177215189873418	0.2857142857142857
f3(2)	0.005633802816901409	0.0319634703196347	0.009259259259259259	0.03164556962025317	0.6071428571428571

Figure 4: MLE for feature f2

f4(0)	0.346004
f4(1)	0.21345
f4(2)	0.105263
f4(3)	0.307992
f4(4)	0.0272904

Figure 5: MLE for feature f4

f3	f3(0)	f3(0)	f3(1)	f3(1)	f3(2)	f3(2)
f9	f9(1)	f9(2)	f9(1)	f9(2)	f9(1)	f9(2)
f5(0)	0.9615384615384616	0.12264150943396226	0.3641025641025641	0.09969788519637462	0.3333333333333333	0.1935483870967742
f5(1)	0.0	0.0	0.005128205128205128	0.055891238670694864	0.0	0.0
f5(2)	0.038461538461538464	0.8679245283018868	0.6256410256410256	0.8383685800604229	0.3333333333333333	0.6451612903225806
f5(3)	0.0	0.009433962264150943	0.005128205128205128	0.006042296072507553	0.3333333333333333	0.16129032258064516

Figure 6: MLE for feature 5

f9	f9(1)	f9(2)
f6(0)	0.004405286343612335	0.017521902377972465
f6(1)	0.31718061674008813	0.16896120150187735
f6(2)	0.5947136563876652	0.60450563204005
f6(3)	0.08370044052863436	0.20901126408010012

Figure 7: MLE for feature f6

f9	f9(1)	f9(2)
f7(0)	0.2511013215859031	0.5281602002503129
f7(1)	0.4669603524229075	0.3078848560700876
f7(2)	0.23348017621145375	0.04380475594493116
f7(3)	0.048458149779735685	0.12015018773466833

Figure 8: MLE for feature f7

f8(0)	0.170565
f8(1)	0.232943
f8(2)	0.255361
f8(3)	0.206628
f8(4)	0.134503

Figure 9: MLE for feature f8

f8	f8(0)	f8(1)	f8(2)	f8(3)	f8(4)
f9(1)	0.21714285714285714	0.3598326359832636	0.35877862595419846	0.009433962264150943	0.050724637681159424
f9(2)	0.7828571428571428	0.6401673640167364	0.6412213740458015	0.9905660377358491	0.9492753623188406

Figure 10: MLE for feature f9

4.2 Constructing Bayesian networks using K2Score

Nodes used in the network:

’f1’, ’f2’, ’f3’, ’f4’, ’f5’, ’f6’, ’f7’, ’f8’, ’f9’

Edges:

(’f3’, ’f4’), (’f3’, ’f8’), (’f3’, ’f9’), (’f5’, ’f3’), (’f5’, ’f9’), (’f9’, ’f1’), (’f9’, ’f2’), (’f9’, ’f4’), (’f9’, ’f6’), (’f9’, ’f7’), (’f9’, ’f8’)

Network, where all nodes have at most 1 parents:

(’f3’, ’f4’), (’f5’, ’f9’), (’f5’, ’f3’), (’f9’, ’f8’), (’f9’, ’f7’), (’f9’, ’f1’), (’f9’, ’f6’), (’f9’, ’f2’)

Network, where all nodes have at most 2 parents:

(’f3’, ’f4’), (’f3’, ’f9’), (’f3’, ’f8’), (’f5’, ’f9’), (’f5’, ’f3’), (’f9’, ’f8’), (’f9’, ’f7’), (’f9’, ’f1’), (’f9’, ’f6’), (’f9’, ’f2’), (’f9’, ’f4’)

K2Score

-9462.704892371386

MLE for all features

Below are the maximum likelihood estimation for each fearure, based on the Bayesian network presented on Fig. 11.

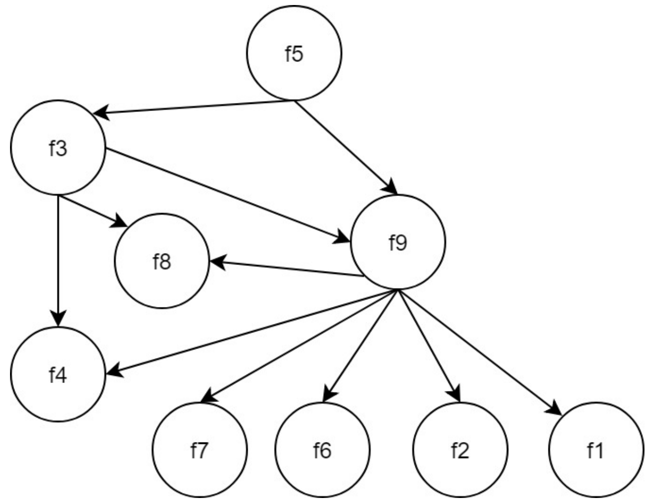


Figure 11: Best Bayesian Network for "and" image dataset using K2Score

f9	f9(1)	f9(2)
f1(0)	0.19823788546255505	0.10638297872340426
f1(1)	0.44933920704845814	0.26032540675844806
f1(2)	0.2643171806167401	0.3692115143929912
f1(3)	0.0881057268722467	0.2640801001251564

Figure 12: MLE for feature f1

f9	f9(1)	f9(2)
f2(0)	0.14537444933920704	0.18648310387984982
f2(1)	0.6255506607929515	0.4856070087609512
f2(2)	0.14537444933920704	0.12640801001251564
f2(3)	0.013215859030837005	0.011264080100125156
f2(4)	0.07048458149779736	0.1902377972465582

Figure 13: MLE for feature f2

f5	f5(0)	f5(1)	f5(2)	f5(3)
f3(0)	0.20765027322404372	0.0	0.11742424242424243	0.07692307692307693
f3(1)	0.7486338797814208	1.0	0.8547979797979798	0.38461538461538464
f3(2)	0.04371584699453552	0.0	0.027777777777777776	0.5384615384615384

Figure 14: MLE for feature f3

f3	f3(0)	f3(0)	f3(1)	f3(1)	f3(2)	f3(2)
f9	f9(1)	f9(2)	f9(1)	f9(2)	f9(1)	f9(2)
f4(0)	0.11538461538461539	0.1792452830188679	0.3641025641025641	0.39274924471299094	0.16666666666666666	0.03225806451612903
f4(1)	0.5769230769230769	0.1509433962264151	0.28717948717948716	0.18882175226586104	0.3333333333333333	0.16129032258064516
f4(2)	0.23076923076923078	0.5094339622641509	0.07692307692307693	0.04833836858006042	0.0	0.03225806451612903
f4(3)	0.07692307692307693	0.1320754716981132	0.26666666666666666	0.3595166163141994	0.0	0.3225806451612903
f4(4)	0.0	0.02830188679245283	0.005128205128205128	0.010574018126888218	0.5	0.45161290322580644

Figure 15: MLE for feature f4

f5(0)	0.178363
f5(1)	0.037037
f5(2)	0.77193
f5(3)	0.0126706

Figure 16: MLE for feature f5

f9	f9(1)	f9(2)
f6(0)	0.004405286343612335	0.017521902377972465
f6(1)	0.31718061674008813	0.16896120150187735
f6(2)	0.5947136563876652	0.60450563204005
f6(3)	0.08370044052863436	0.20901126408010012

Figure 17: MLE for feature f6

f9	f9(1)	f9(2)
f7(0)	0.2511013215859031	0.5281602002503129
f7(1)	0.4669603524229075	0.3078848560700876
f7(2)	0.23348017621145375	0.04380475594493116
f7(3)	0.048458149779735685	0.12015018773466833

Figure 18: MLE for feature f7

4.2.1 Converting to Markov Network

Bayesian Network that was generated using K2Score: ('f3', 'f4'), ('f3', 'f8'), ('f3', 'f9'), ('f5', 'f3'), ('f5', 'f9'), ('f9', 'f1'), ('f9', 'f2'), ('f9', 'f4'), ('f9','f6'), ('f9', 'f7'), ('f9', 'f8')

was converted to Markov network using pgmpy fucntion **to_markov_model()**. Obtained Markov network: ('f3', 'f4'), ('f3', 'f8'), ('f3', 'f9'), ('f3', 'f5'), ('f4', 'f9'), ('f8', 'f9'), ('f9', 'f1'), ('f9', 'f2'), ('f9', 'f6'), ('f9', 'f7'), ('f9', 'f5')

f3	f3(0)	f3(0)	f3(1)	f3(1)	f3(2)	f3(2)
f9	f9(1)	f9(2)	f9(1)	f9(2)	f9(1)	f9(2)
f8(0)	0.19230769230769232	0.16037735849056603	0.16923076923076924	0.18126888217522658	0.0	0.0
f8(1)	0.3076923076923077	0.11320754716981132	0.37948717948717947	0.2084592145015106	0.6666666666666666	0.0967741935483871
f8(2)	0.4230769230769231	0.20754716981132076	0.4153846153846154	0.19939577039274925	0.3333333333333333	0.45161290322580644
f8(3)	0.0	0.42452830188679247	0.010256410256410256	0.23564954682779457	0.0	0.2903225806451613
f8(4)	0.07692307692307693	0.09433962264150944	0.02564102564102564	0.17522658610271905	0.0	0.16129032258064516

Figure 19: MLE for feature f8

f3	f3(0)	f3(0)	f3(0)	f3(0)	f3(1)	f3(1)	f3(1)	f3(1)	f3(2)	f3(2)	f3(2)	f3(2)
f5	f5(0)	f5(1)	f5(2)	f5(3)	f5(0)	f5(1)	f5(2)	f5(3)	f5(0)	f5(1)	f5(2)	f5(3)
f9(1)	0.6578947368421053	0.5	0.010752688172043012	0.0	0.5182481751824818	0.02631578947368421	0.18020679468242246	0.2	0.25	0.5	0.09090909090909091	0.2857142857142857
f9(2)	0.34210526315789475	0.5	0.989247311827957	1.0	0.48175182481751827	0.9736842105263158	0.8197932053175776	0.8	0.75	0.5	0.9090909090909091	0.7142857142857143

Figure 20: MLE for feature f9