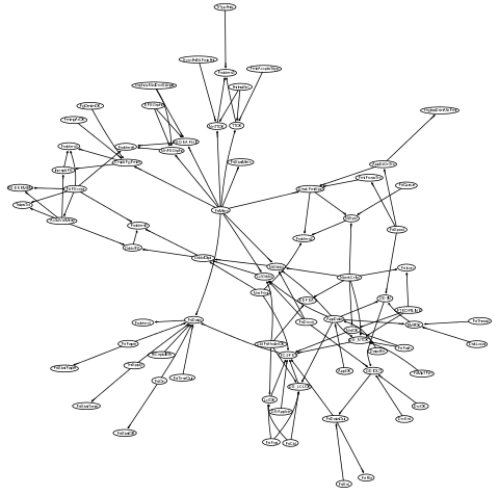
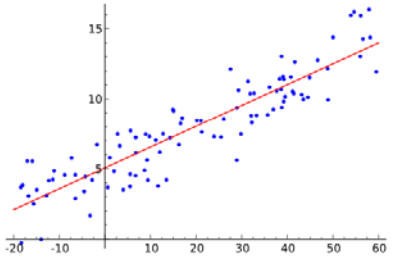
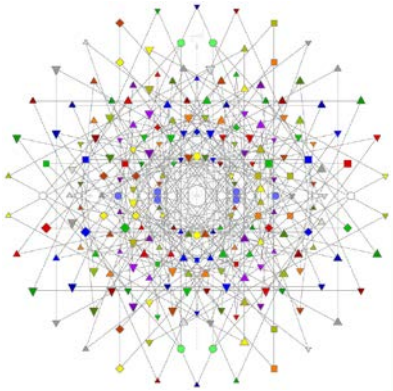






ModelWizard

Interactive Model Construction for Tabular

Dylan Hutchison

Advisors Andy Gordon and Claudio Russo

15 August 2014

Wild_Propagate	redord_ID_	Genus	Species	On_CITES	Country	conserve_concern	conserve_priority	price_USD
P	1	Adenia	cladosepala	FALSE	ND	H	H	8.29
P	2	Adenia	cladosepala	FALSE	ND	H	H	71.82
P	3	Adenia	cladosepala	FALSE	DE	H	H	4.56
P	4	Adenia	cladosepala	FALSE	ND	H	H	17.89
P	5	Adenia	cladosepala	FALSE	CZ	H	H	1
P	6	Adenia	cladosepala	FALSE	USA	H	H	38
P	7	Adenia	cladosepala	FALSE	CYPRUS	H	H	7.58
P	8	Adenia	cladosepala	FALSE	ND	H	H	17.85
P	9	Adenia	cladosepala	FALSE	CZ	H	H	1
P	10	Adenia	cladosepala	FALSE	DE	H	H	7.58
P	11	Adenia	cladosepala	FALSE	UK	H	H	37.13
P	12	Adenia	cladosepala	FALSE	DE	H	H	7.15
P	13	Adenia	cladosepala	FALSE	CYPRUS	H	H	7.57
P	14	Adenia	cladosepala	FALSE	DE	H	H	7.15
P	15	Adenia	cladosepala	FALSE	ND	H	H	17.82
P	16	Adenia	cladosepala	FALSE	CZ	H	H	1
P	17	Adenia	cladosepala	FALSE	USA	H	H	38
P	18	Adenia	cladosepala	FALSE	DE	H	H	4.54
P	19	Adenia	cladosepala	FALSE	USA	H	H	9.95
P	20	Adenia	cladosepala	FALSE	GE	H	H	1.37

tmain

ID	int	input
Country	string	input
Genus	string	input
On_CITES	bool	input
Species	string	input
Wild_Propagate	string	input
conserve_concern	string	input
conserve_priority	string	input
price_USD	real	input
redord_ID_	int	input

```
W.TypeInfer "tmain" ""
W.ExactInfer "tmain" ["Genus"; "Species"]
W.NaiveBayes "tmain" "Wild_Propagate"

["conserve_priority"; "conserve_concern"; "On_CITES"]
|> List.map (W.Model "T_Genus_Species")
```

T_conserve_priority

ID	int	input	
conserve_priority	string	input	pk

T_conserve_concern

ID	int	input	
conserve_concern	string	input	pk

T_Wild_Propagate

ID	int	input	
Wild_Propagate	string	input	pk

T_Species

ID	int	input	
Species	string	input	pk

T_On_CITES

ID	int	input	
On_CITES	string	input	pk

T_Genus

ID	int	input	
Genus	string	input	pk

T_Country

ID	int	input	
Country	string	input	pk

T_Genus_Species Visible Uniques Table

ID	int	input	
Genus	link(T_Genus)	input	pk
Species	link(T_Species)	input	pk
conserve_priority	link(T_conserve_priority)	output	CDiscrete(N=SizeOf(T_conserve_priority))
conserve_concern	link(T_conserve_concern)	output	CDiscrete(N=SizeOf(T_conserve_concern))
On_CITES	link(T_On_CITES)	output	CDiscrete(N=SizeOf(T_On_CITES))

tmain

ID	int	input	
Wild_Propagate	link(T_Wild_Propagate)	output	CDiscrete(N=SizeOf(T_Wild_Propagate))
Genus_Species	link(T_Genus_Species)	output	CDiscrete(N=SizeOf(T_Genus_Species))[Wild_Propagate]
Country	link(T_Country)	output	CDiscrete(N=SizeOf(T_Country))[Wild_Propagate]
Genus	link(T_Genus)	output	Genus_Species.Genus
On_CITES	link(T_On_CITES)	output	Genus_Species.On_CITES
Species	link(T_Species)	output	Genus_Species.Species
conserve_concern	link(T_conserve_concern)	output	Genus_Species.conserve_concern
conserve_priority	link(T_conserve_priority)	output	Genus_Species.conserve_priority
price_USD	real	output	CGaussian(MeanMean=45.53844295,MeanPrec=0.0001470609862)[Wild_Propagate]

ID	Wild_Propagate	Genus	Species	On_CITES	Country	conserve_concern	conserve_priority	price_USD	Genus_Species
0	3					12		9.08	16
1	3					4		1.2	17
2	3					12		33.42	17
3	2					16		80	24
4	2					16		35	21
5	2					7		93.57	18
6	2					6		123.85	21
7	2					5		3.85	33
8	2					15		4.63	27
9	2					13		53.67	21
10	2					5		40.6	33
11	0					16		75	3
12	3					5		4.54	16
13	2					12		50.36	21
14	2					7		38.53	33
15	2					12		462.39	21
16	3					15		5.89	16
17	3					4		8.97	16
18	2					5		14.44	21
19	3					4		29.87	17
20	2					16		10.99	24

ID	conserve_priority
0	H
1	H-M
2	L
3	M
4	M-H
5	U

ID	conserve_concern
0	H
1	H-M
2	L
3	M
4	M-H
5	U

ID	Wild_Propagate
0	P
1	U
2	W
3	W-P

ID	Species
0	abbreviata
1	aprevalii
2	borealis
3	cladosepala
4	cornigera
5	decaryi
6	ecirrosa
7	elephantophus
8	epigea
9	firingavalensis
10	grandidieri
11	guillauminii
12	hirsutissima
13	humbertii
14	hyphaenoides
15	ihlenfeldtiana
16	isaloensis
17	laza
18	leptocarpa
19	monadelpha
20	monstruosa

ID	On_CITES
0	FALSE
1	TRUE

ID	Genus
0	Adenia
1	Commiphora
2	Cyphostemma
3	Operculicarya
4	Uncarina

ID	Country
0	AFRICA
1	AFRICA ZA
2	AUSTRALIA
3	CYPRUS
4	CZ
5	DE
6	ES
7	FR
8	GE
9	HUNGARY
10	INDIA
11	IT
12	ND
13	REUNION
14	THAILAND
15	UK
16	USA
17	ZA

ID	Genus	Species	conserve_priority	conserve_concern	On_CITES
0	0	3	0	0	0
1	0	6	4	5	0
2	0	8	0	0	0
3	0	9	0	0	1
4	0	16	0	0	0
5	0	19	0	0	0
6	0	21	0	0	1
7	0	25	3	3	0
8	0	31	1	1	1
9	1	1	1	1	0
10	1	11	1	0	0
11	1	13	1	0	0
12	1	20	1	0	0
13	1	22	1	0	0
14	1	29	2	3	0
15	2	4	4	5	0
16	2	7	0	0	1
17	2	17	0	4	1
18	2	23	0	0	0
19	2	27	3	0	0
20	3	2	0	0	0

Tabular: Probabilistic Models on Schemas

① Get Data into Excel, clean data with tools like Power Query

② Add in query-by-missing-value rows

③ Write a Tabular Model

④ Compile to Infer.NET and profit!

Player	Name
0	Alice
1	Bob
2	Cynthia

Match	Player1	Player2	Win1
0	0	1	FALSE
1	1	2	FALSE
2	0	2	

①

②

Players

Skill	real	latent	GaussianFromMeanAndPrecision(25.0,0.01)

Matches

③

Player1	link(Players)	input	
Player2	link(Players)	input	
Perf1	real	latent	GaussianFromMeanAndPrecision(Player1.Skill,0.01)
Perf2	real	latent	GaussianFromMeanAndPrecision(Player2.Skill,0.01)
Win1	bool	output	Perf1 > Perf2

④

Log Evidence

-1.56

Players

Player	Skill
0	Gaussian(20.25, 82.28)
1	Gaussian(25, 70.66)
2	Gaussian(29.75, 82.28)

Matches

Match	Perf1	Perf2	Win1
0	Gaussian(15.49, 129.1)	Gaussian(29.75, 123.6)	Bernoulli(0)
1	Gaussian(20.25, 123.6)	Gaussian(34.51, 129.1)	Bernoulli(0)
2	Gaussian(20.25, 182.3)	Gaussian(29.75, 182.3)	Bernoulli(0.3092)

Tabular: Probabilistic M

① Get Data into Excel, clean data wit

② Add in query-by-missing-valu ov

③ Write a Tabular Model

④ Compile to Infer.NET and profit!

③: Easier than writing Infer.NET but still hard for Data Scientists. Which models make sense? Which perform best?

Player	Name
0	Alice
1	Bob
2	Cynthia

Match	Player1	Player2	Win1
0	0	1	FALSE
1	1	2	FALSE
2	0	2	

①

②

Players

Skill	real	latent	GaussianFromMeanAndPrecision(25.0,0.01)

Matches

③

Player1	link(Players)	input	
Player2	link(Players)	input	
Perf1	real	latent	GaussianFromMeanAndPrecision(Player1.Skill,0.01)
Perf2	real	latent	GaussianFromMeanAndPrecision(Player2.Skill,0.01)
Win1	bool	output	Perf1 > Perf2

④

Infer.NET

Log Evidence

-1.56

Players

Player	Skill
0	Gaussian(20.25, 82.28)
1	Gaussian(25, 70.66)
2	Gaussian(29.75, 82.28)

Matches

Match	Perf1	Perf2	Win1
0	Gaussian(15.49, 129.1)	Gaussian(29.75, 123.6)	Bernoulli(0)
1	Gaussian(20.25, 123.6)	Gaussian(34.51, 129.1)	Bernoulli(0)
2	Gaussian(20.25, 182.3)	Gaussian(29.75, 182.3)	Bernoulli(0.3092)

Ex1: Clouds, Rain, Sprinklers, WetGrass

- 4 Boolean Variables
- Space of Models = Bayesian Networks
  - Let "➔" mean "*governs the distribution behind*"
  - Rain ➔ WetGrass?
  - Sprinklers ➔ Rain??
- Use ModelWizard to explore & compare

cloudy	sprinkler	rain	wetGrass
TRUE	FALSE		FALSE
FALSE	FALSE	TRUE	TRUE
FALSE	TRUE	FALSE	FALSE
FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	FALSE
TRUE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE

Classic model, taken from Kevin Murphey 1998.
Data generated by sampling an Infer.NET program.

tmain

ID	int	input
cloudy	string	input
rain	string	input
sprinkler	string	input
wetGrass	string	input

1. TypeInfer

T_wetGrass

ID	int	input	
wetGrass	string	input	pk

T_sprinkler

ID	int	input	
sprinkler	string	input	pk

T_rain

ID	int	input	
rain	string	input	pk

T_cloudy

ID	int	input	
cloudy	string	input	pk

2. Model

tmain

ID	int	input	
cloudy	link(T_cloudy)	output	CDiscrete(N=SizeOf(T_cloudy))
rain	link(T_rain)	output	CDiscrete(N=SizeOf(T_rain))
sprinkler	link(T_sprinkler)	output	CDiscrete(N=SizeOf(T_sprinkler))
wetGrass	link(T_wetGrass)	output	CDiscrete(N=SizeOf(T_wetGrass))

tmain

ID	int	input	
cloudy	link(T_cloudy)	input	
rain	link(T_rain)	input	
sprinkler	link(T_sprinkler)	input	
wetGrass	link(T_wetGrass)	input	

3. Approx

tmain

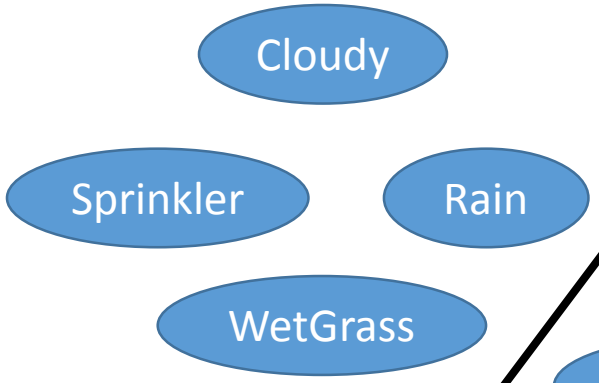
ID	int	input	
cloudy	link(T_cloudy)	output	CDiscrete(N=SizeOf(T_cloudy))
rain	link(T_rain)	output	CDiscrete(N=SizeOf(T_rain))[cloudy]
sprinkler	link(T_sprinkler)	output	CDiscrete(N=SizeOf(T_sprinkler))[cloudy]
wetGrass	link(T_wetGrass)	output	CDiscrete(N=SizeOf(T_wetGrass))[sprinkler][rain]

Model Improvement: Predicting Cloudy

All-Independent Model

Log Evidence
-2661.41

Predict \ Truth	0	1
0	19.6	27.4
1	78.2	74.8

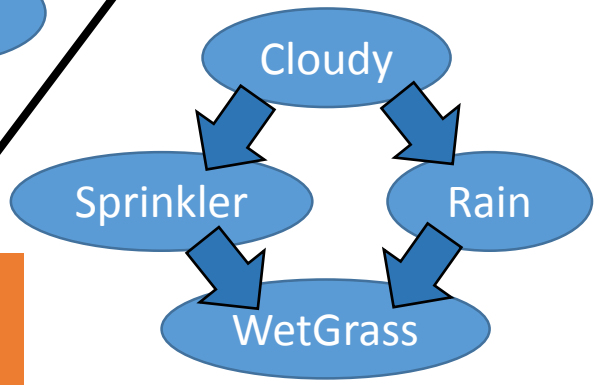


cloudy	link(T_cloudy)	output	CDiscrete(N=SizeOf(T_cloudy))
rain	link(T_rain)	output	CDiscrete(N=SizeOf(T_rain))
sprinkler	link(T_sprinkler)	output	CDiscrete(N=SizeOf(T_sprinkler))
wetGrass	link(T_wetGrass)	output	CDiscrete(N=SizeOf(T_wetGrass))

Final Model

Log Evidence
-1963.57

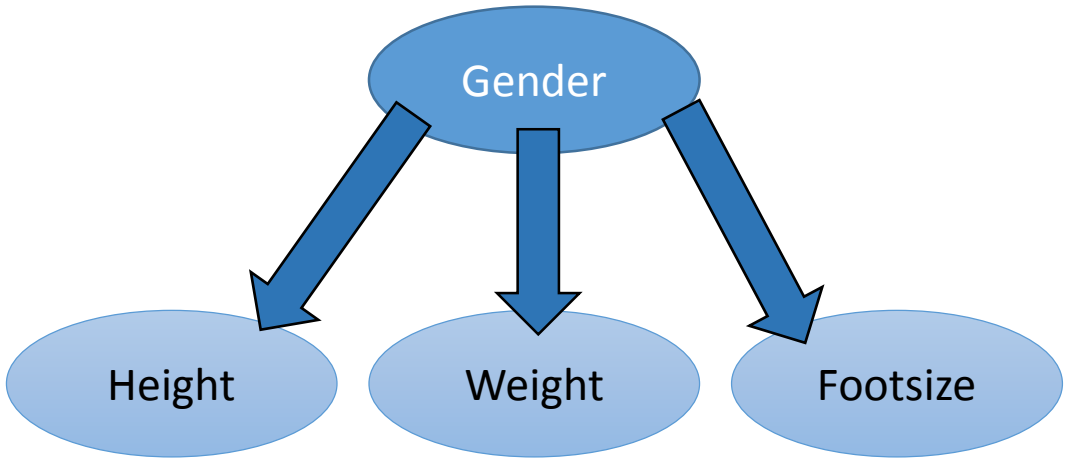
Predict \ Truth	0	1
0	87.6 ↑	28.6
1	10.2 ↓	73.6



1000 rows of data
5-fold cross-validation
⇒ 20% data held out
= 200 test predictions
average per fold

cloudy	link(T_cloudy)	output	CDiscrete(N=SizeOf(T_cloudy))
rain	link(T_rain)	output	CDiscrete(N=SizeOf(T_rain))**[cloudy]**
sprinkler	link(T_sprinkler)	output	CDiscrete(N=SizeOf(T_sprinkler))**[cloudy]**
wetGrass	link(T_wetGrass)	output	CDiscrete(N=SizeOf(T_wetGrass))**[sprinkler][rain]**

Ex2: Naive Bayes Classifier: Gender



ID	gender	height	weight	footsize
0	1	6	180	12
1	1	5.92	190	11
2	1	5.58	170	12
3		5.92	165	10
4	0	5	100	6
5	0	5.5	150	8
6	0	5.42	130	7
7		5.75	150	9

Data from Wikipedia Naive Bayes example

tmain

ID	int	input
footsize	int	input
gender	int	input
height	real	input
weight	int	input

1. NaiveBayes

```
NaiveBayes tmain (+)
  ReorderColumns tmain
  Type tmain gender Nominal
  Model tmain gender
  Model tmain footsize
  Model tmain height
  Model tmain weight
  Approx tmain gender footsize
  Approx tmain gender height
  Approx tmain gender weight
  EstimateHyper tmain (+)
    EstimateHyper tmain weight
    EstimateHyper tmain height
    EstimateHyper tmain footsize
```

T_gender

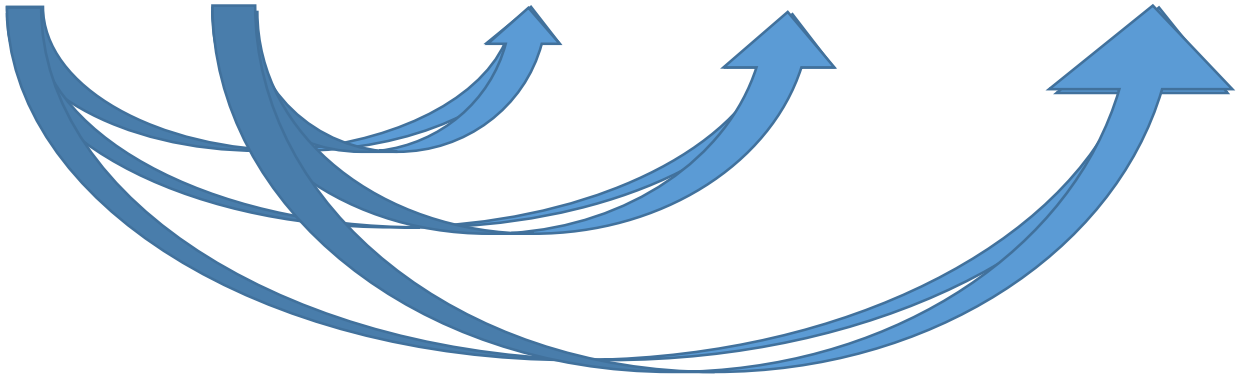
ID	int	input	
gender	string	input	pk

tmain

ID	int	input	
gender	link(T_gender)	output	CDiscrete(N=SizeOf(T_gender))
footsize	real	output	CGaussian(MeanMean=9.375,MeanPrec=0.01667)[gender]
height	real	output	CGaussian(MeanMean=5.63625,MeanPrec=0.1)[gender]
weight	real	output	CGaussian(MeanMean=154.375,MeanPrec=0.00111)[gender]

Ex3: Exact Functional Dependencies: Plant Sales in Madagascar

Genus	Species	On_CITES	Country	conserve_concern	conserve_priority	price_USD
Adenia	olaboensis	TRUE	ND	H	H	22
Adenia	olaboensis	TRUE	DE	H	H	5.91
Adenia	olaboensis	TRUE	CZ	H	H	8.96
Adenia	perrieri	FALSE	FR	M	M	16.51
Adenia	perrieri	FALSE	ND	M	M	17.81
Adenia	perrieri	FALSE	ND	M	M	17.81



Thanks to Matt Smith for the dataset!

T_conserve_priority

ID	int	input	
conserve_priority	string	input	pk

T_conserve_concern

ID	int	input	
conserve_concern	string	input	pk

T_Wild_Propagate

ID	int	input	
Wild_Propagate	string	input	pk

T_Species

ID	int	input	
Species	string	input	pk

T_On_CITES

ID	int	input	
On_CITES	string	input	pk

T_Genus

ID	int	input	
Genus	string	input	pk

T_Country

ID	int	input	
Country	string	input	pk

tmain

ID	int	input	
Country	link(T_Country)	input	
Genus	link(T_Genus)	input	
On_CITES	link(T_On_CITES)	input	
Species	link(T_Species)	input	
Wild_Propagate	link(T_Wild_Propagate)	input	
conserve_concern	link(T_conserve_concern)	input	
conserve_priority	link(T_conserve_priority)	input	
price_USD	real	input	

1. **TypeInfer** tmain

tmain

ID	int	input
Country	string	input
Genus	string	input
On_CITES	bool	input
Species	string	input
Wild_Propagate	string	input
conserve_concern	string	input
conserve_priority	string	input
price_USD	real	input

2. **ExactInfer** [Genus,Species]

T_Genus_Species Visible Uniques Table

ID	int	input	
Genus	link(T_Genus)	input	pk
Species	link(T_Species)	input	pk
conserve_priority	link(T_conserve_priority)	input	
conserve_concern	link(T_conserve_concern)	input	
On_CITES	link(T_On_CITES)	input	

Columns exactly determined by (Genus, Species)

tmain

ID	int	input	
Genus_Species	link(T_Genus_Species)	output	CDiscrete(N=SizeOf(T_Genus_Species))
Country	link(T_Country)	input	
Genus	link(T_Genus)	output	Genus_Species.Genus
On_CITES	link(T_On_CITES)	output	Genus_Species.On_CITES
Species	link(T_Species)	output	Genus_Species.Species
Wild_Propagate	link(T_Wild_Propagate)	input	
conserve_concern	link(T_conserve_concern)	output	Genus_Species.conserve_concern
conserve_priority	link(T_conserve_priority)	output	Genus_Species.conserve_priority
price_USD	real	input	

3. **NaiveBayes** tmain
4. **Model** T_Genus_Species ...

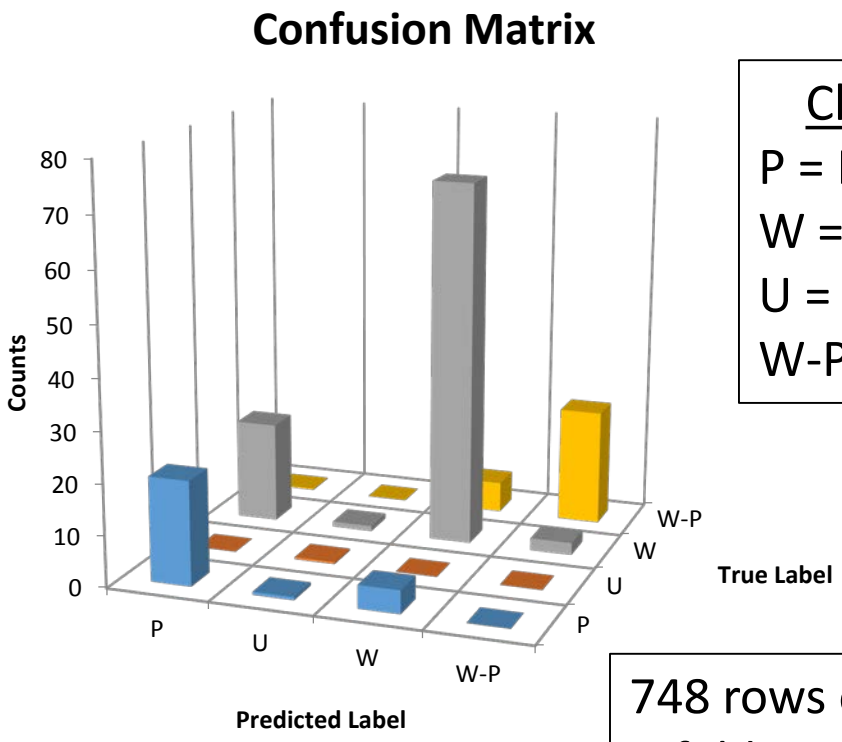
T_Genus_Species	Visible Uniques Table		
ID	int	input	
Genus	link(T_Genus)	input	pk
Species	link(T_Species)	input	pk
conserve_priority	link(T_conserve_priority)	output	CDiscrete(N=SizeOf(T_conserve_priority))
conserve_concern	link(T_conserve_concern)	output	CDiscrete(N=SizeOf(T_conserve_concern))
On_CITES	link(T_On_CITES)	output	CDiscrete(N=SizeOf(T_On_CITES))

tmain

ID	int	input	
Wild_Propagate	link(T_Wild_Propagate)	output	CDiscrete(N=SizeOf(T_Wild_Propagate))
Genus_Species	link(T_Genus_Species)	output	CDiscrete(N=SizeOf(T_Genus_Species))[Wild_Propagate]
Country	link(T_Country)	output	CDiscrete(N=SizeOf(T_Country))[Wild_Propagate]
Genus	link(T_Genus)	output	Genus_Species.Genus
On_CITES	link(T_On_CITES)	output	Genus_Species.On_CITES
Species	link(T_Species)	output	Genus_Species.Species
conserve_concern	link(T_conserve_concern)	output	Genus_Species.conserve_concern
conserve_priority	link(T_conserve_priority)	output	Genus_Species.conserve_priority
price_USD	real	output	CGaussian(MeanMean=45.5384,MeanPrec=0.00014706)[Wild_Propagate]

Better Models by Capturing Exact FDs

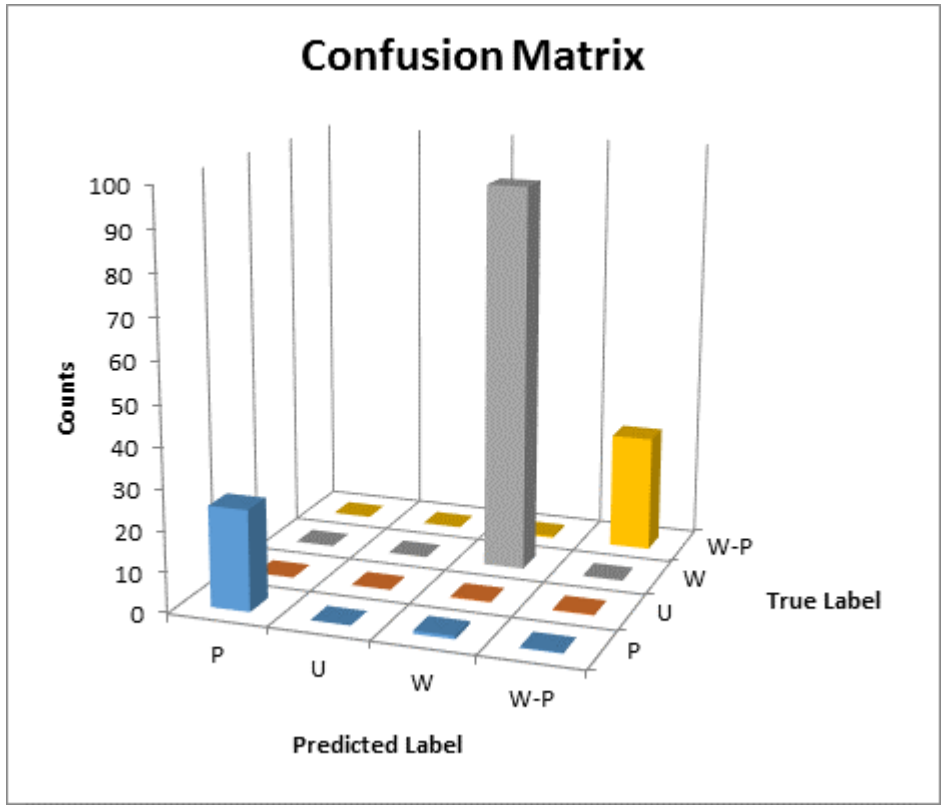
Without EFDs



Class Labels
P = Propagate
W = Wild
U = Unknown
W-P = Unknown

748 rows of data
5-fold cross-validation
⇒ 20% data held out
= 149.6 test predictions
average per fold

With EFDs



```
type State = Schema * Data
type Operation = State -> State option
```

Primitive Operations

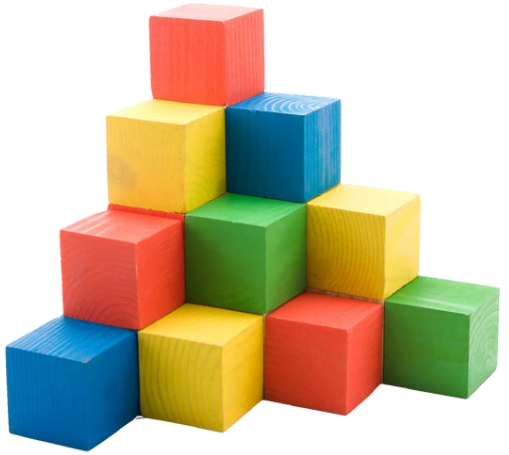
- Type
- Model
- Approx
- JoinDomain
- Exact

Compound Operations

- TypeInfer
- EstimateHyper
- NaiveBayes
- ExactInfer

Later...

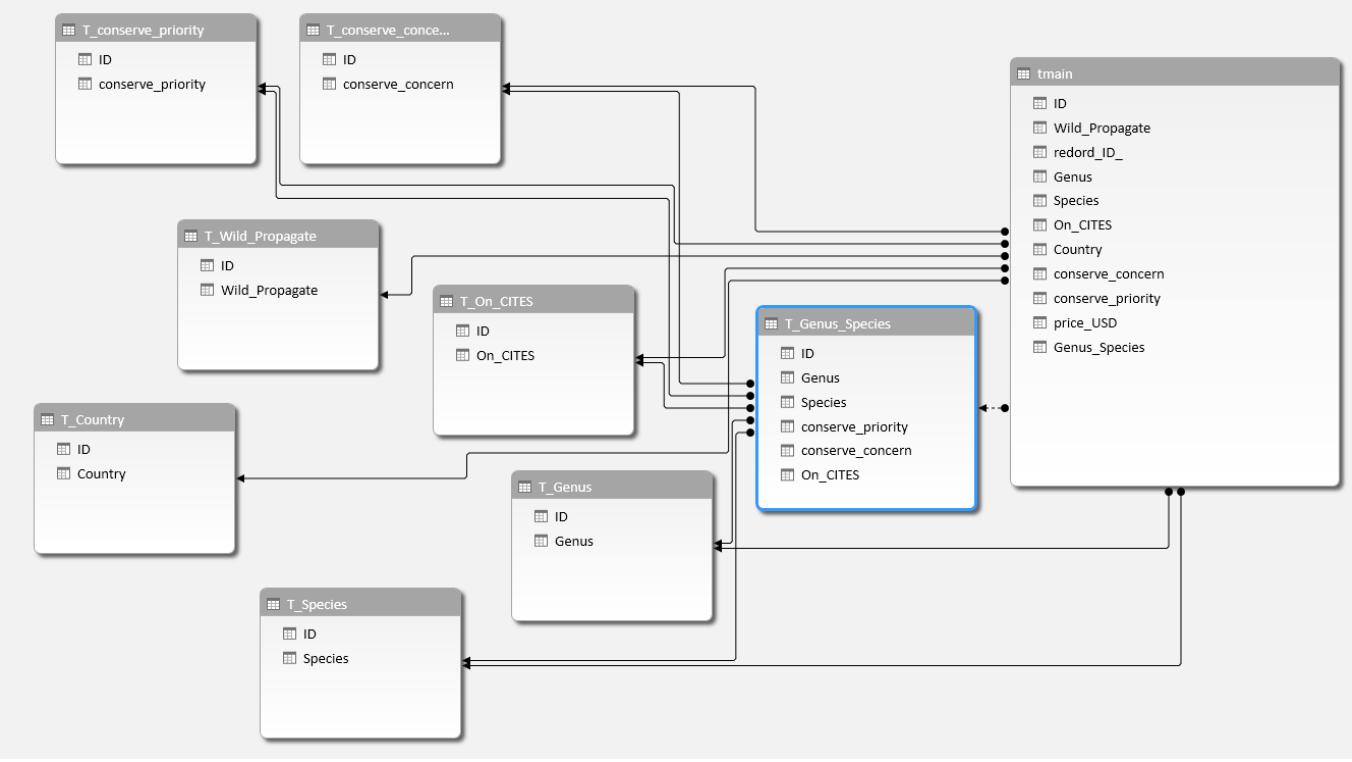
- Regression
- Clustering
- Time
- Matrix Factorization

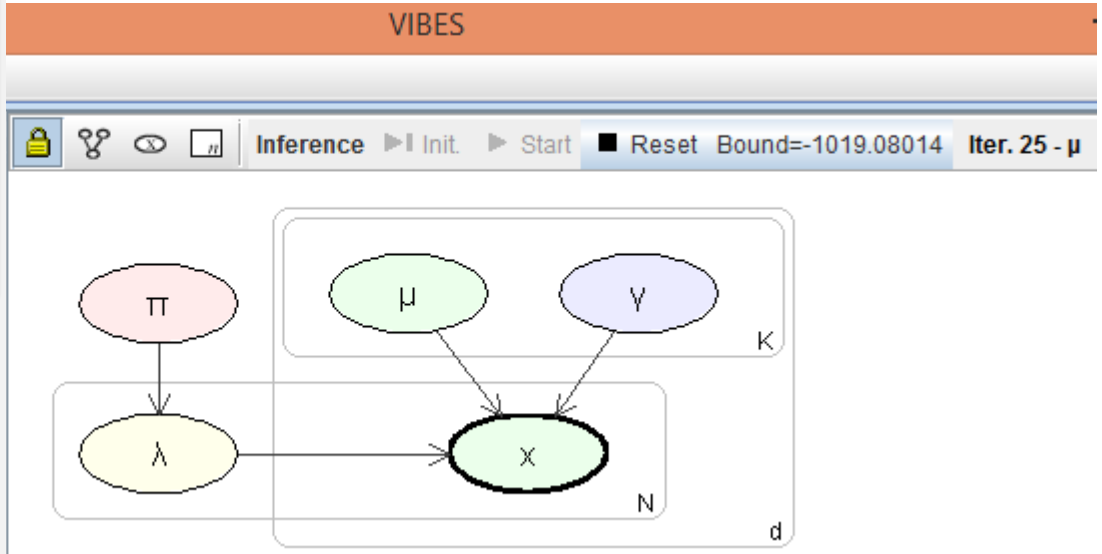


Future: ModelWizard GUI

- Related to 2002 VIBES graphical model builder by John Winn
- Integration with existing tools
  - Excel Data Model, Factor Graph view

➔ Default models within reach of Data Scientists

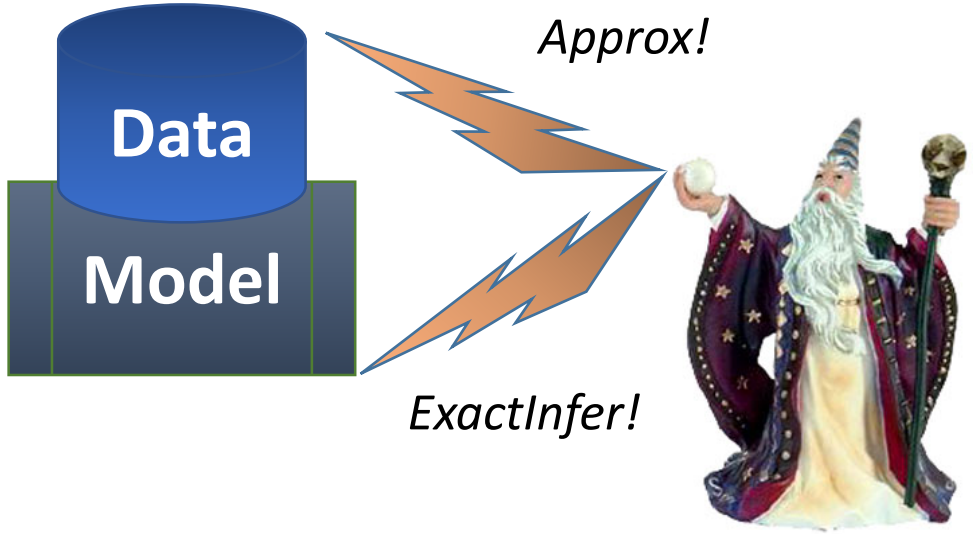




ModelWizard

Interactive Model Construction

New abstraction
to simultaneously refine...



	Your Expertise
+	**Automation & Search**
+	**Default Models**
=	**Discovery** ✓

```
State = Schema * Data
PrimOp = State -> State option
```

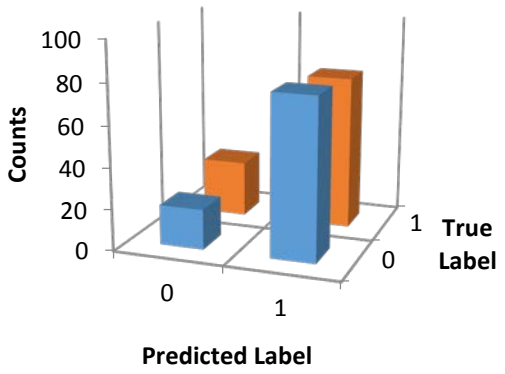
Backup

Bayesian Network: Inferred Parameters

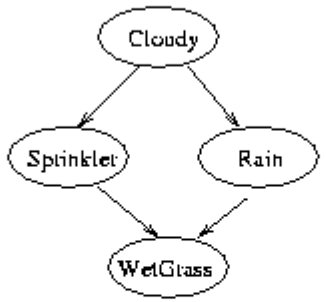
All-independent model: Marginal Distributions

cloudy_V	Dirichlet(490 512)	0.51
rain_V	Dirichlet(489 512)	0.51
sprinkler_V	Dirichlet(698 304)	0.30
wetGrass_V	Dirichlet(355 647)	0.65

All-Independent Model



P(C=F)	P(C=T)
0.5	0.5



C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

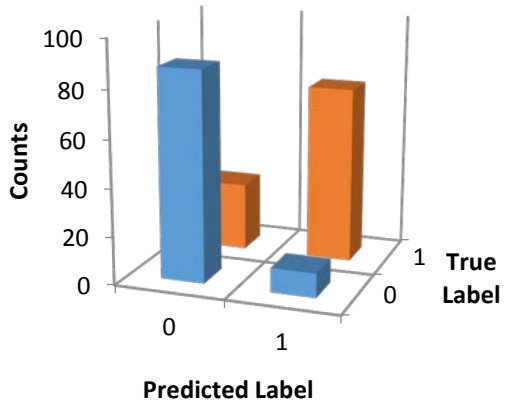
C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Full model: Conditional Distributions

cloudy_V	Dirichlet(490 512)	0.51
rain_V_0	Dirichlet(390 101)	0.21
rain_V_1	Dirichlet(100.6 412)	0.80
sprinkler_V_0	Dirichlet(239 252)	0.51
sprinkler_V_1	Dirichlet(460 53)	0.10
wetGrass_V_0_0	Dirichlet(278.7 1)	0.00
wetGrass_V_1_0	Dirichlet(25 188)	0.88
wetGrass_V_0_1	Dirichlet(53.34 368)	0.87
wetGrass_V_1_1	Dirichlet(1 93)	0.99

Final Model



```
type State       = Schema * Data
type Operation   = PrimOp | CompoundOp
type PrimOp      = State -> State option
type CompoundOp  = State -> Operation list
type Program     = Operation list
```

Primitive Operations

- Type
- Model
- Approx
- JoinDomain
- Exact

Compound Operations

- TypeInfer
- EstimateHyper
- NaiveBayes
- ExactInfer

Later...

- Regression
- Clustering
- Matrix Factorization