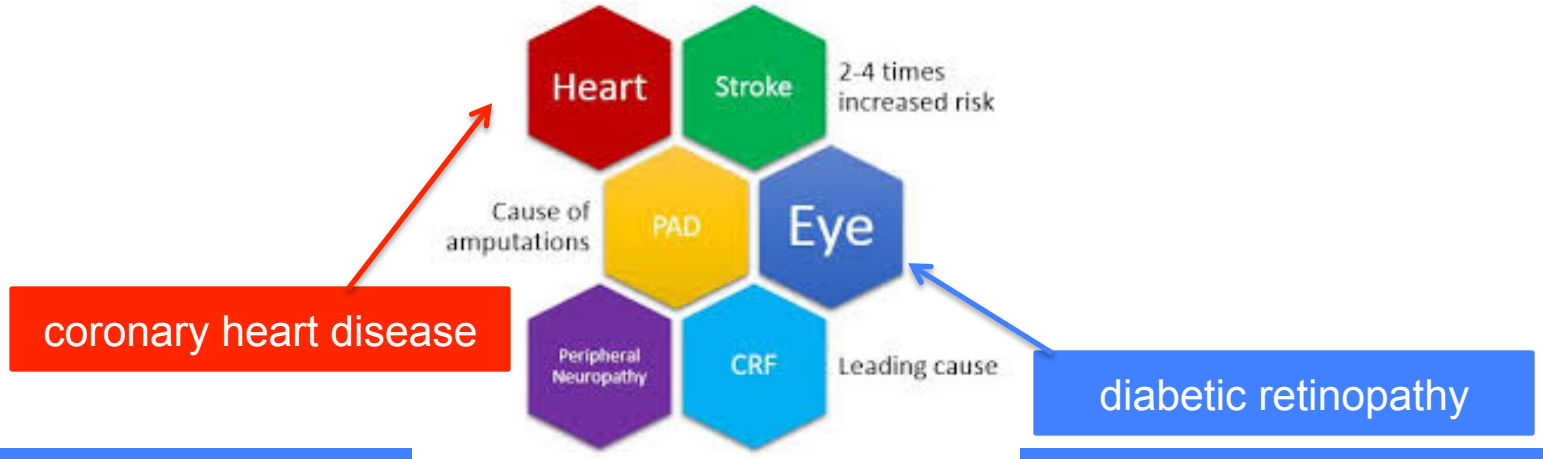
Forecasting Potential Diabetes Complications

Yang Yang, Jie Tang, Juanzi Li
Tsinghua University
Walter Luyten, Marie-Francine Moens
Katholieke Universiteit Leuven
Lu Liu
Northwestern University

Diabetes Complications

- ***Life-Threatening***
  - Over **4.8 million** people died in 2012 due to diabetes[1].
  - Over **68%** of diabetes-related mortality is caused by diabetes complications[2].
  - **471 billion** USD, while **185 million** patients remain undiagnosed[1].
- ***Need to be diagnosed in time***



[1] http://www.diabetes.org/
[2] http://www.idf.org/diabetesatlas/

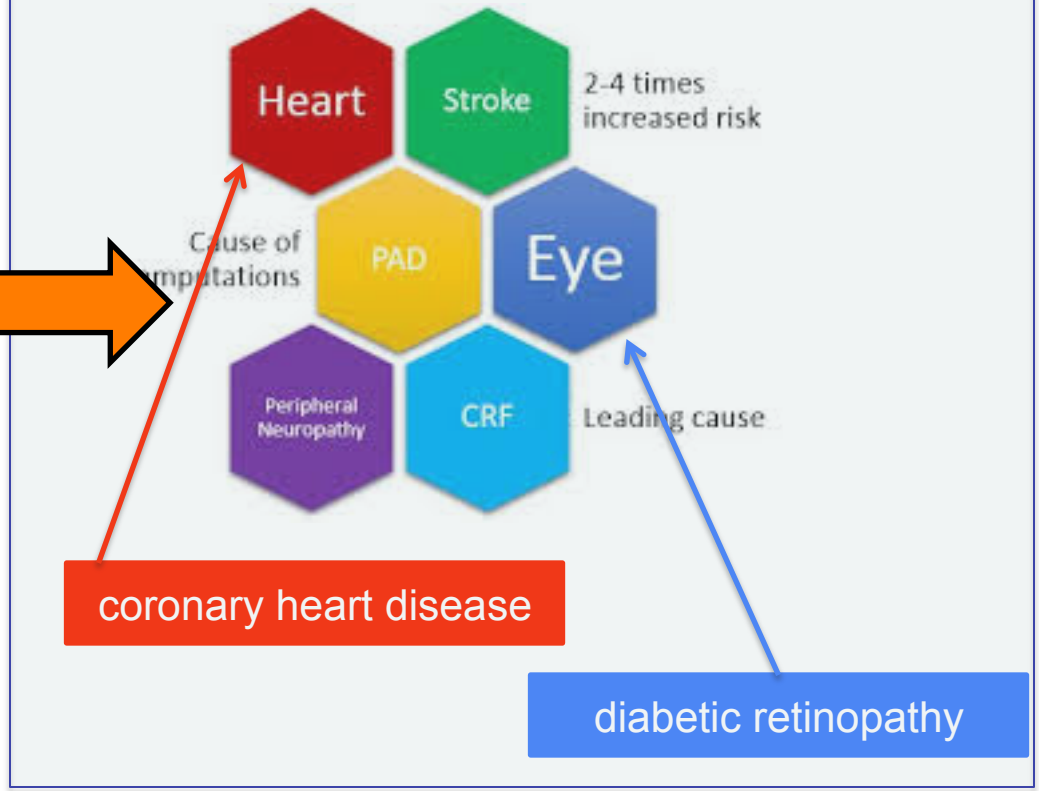
Forecasting Diabetes Complication

Input: a patient's lab test results

Bilirubin example

Routine urine analysis

Output: diabetes complications



Data Set

- A collection of real clinical records from a hospital in Beijing, China over one year.

Item	Statistics
Clinical records	181,933
Patient	35,525
Lab tests	1,945

Clinical record

ALP	81	LDL	2.89
GGT	24.7	APOB	1.31
TP	68.5	APOA	0.97
ALB	44.8	BUN	5.4
GLO	23.7	CREA	41.7
A/G	1.9	UA	248.9
TBIL	12.2	GLU	6.7 ↑

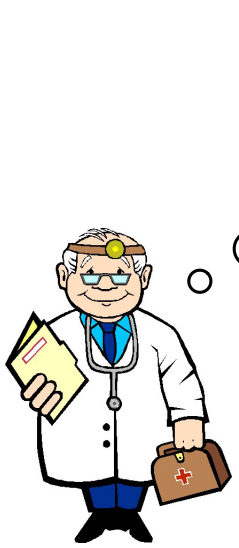
Challenge: feature sparseness

- Each clinical record only contains **24.43** different lab tests
- **65.5%** of lab tests exist in < **10** clinical records (**0.00054%**).

Our Approach

Baseline Model I

Learning task: $f(x_i) \rightarrow y_i$



Complication

CHD

y_i

x_i

Clinical Record

WBC	RBC	PRO	HBV	…
0.5	0.3	/	P	…

Feature vector

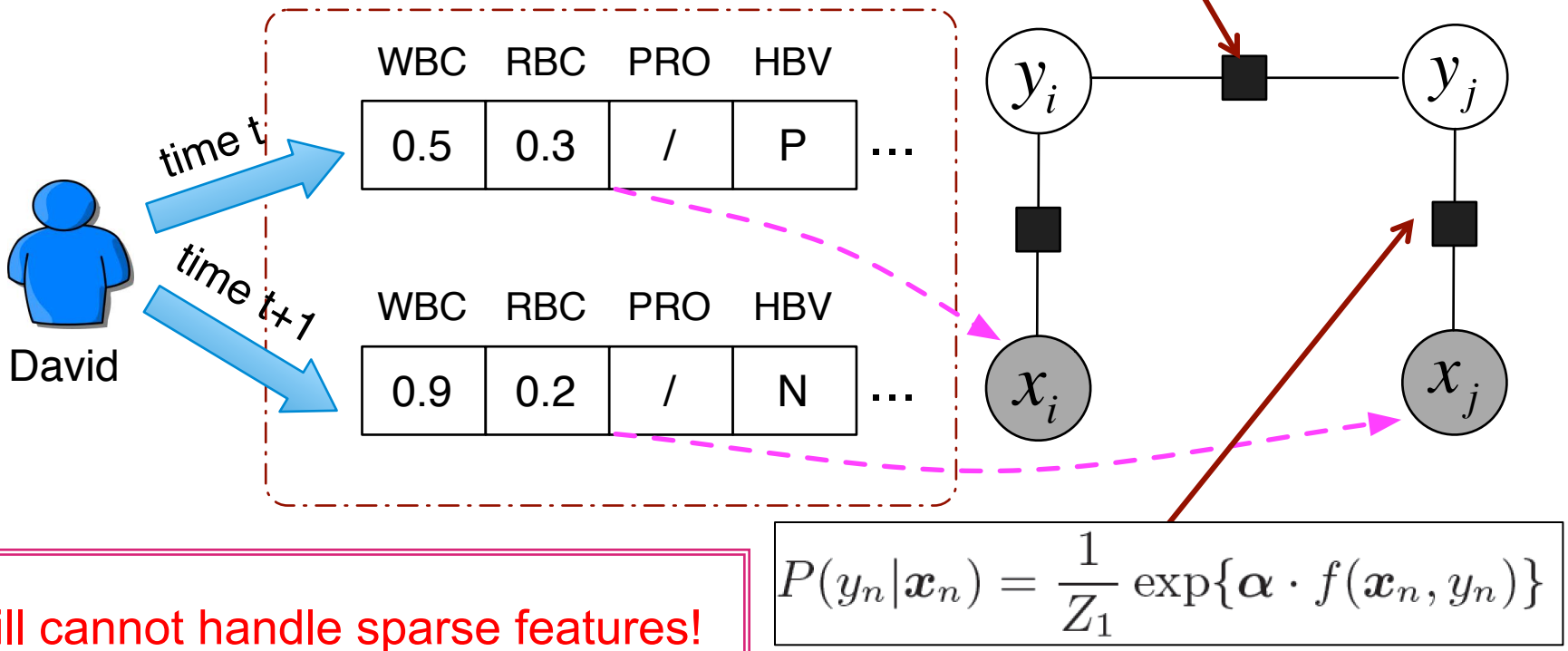
0.5	0.3	/	1	…

Limitations:

1. Cannot model correlations between *y*
2. Cannot handle sparse features

Baseline Model II

$$P(y_n, y_{n'}) = \frac{1}{Z_2} \exp\{\beta \cdot g(y_n, y_{n'})\}$$

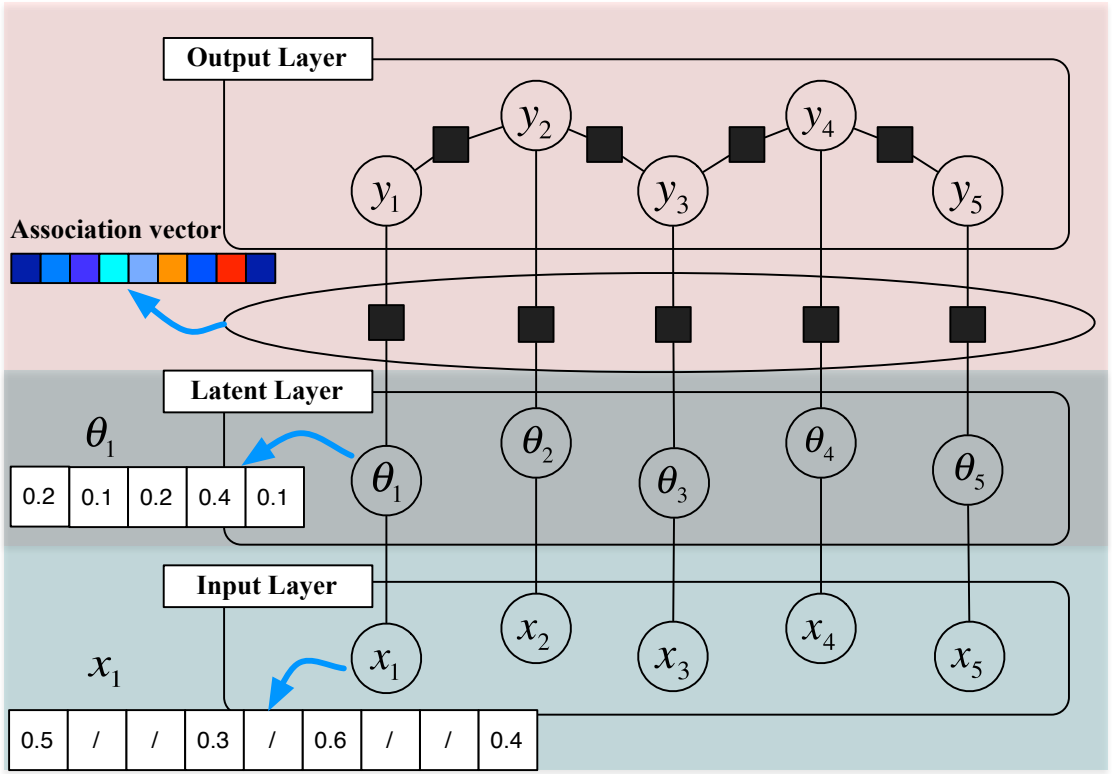


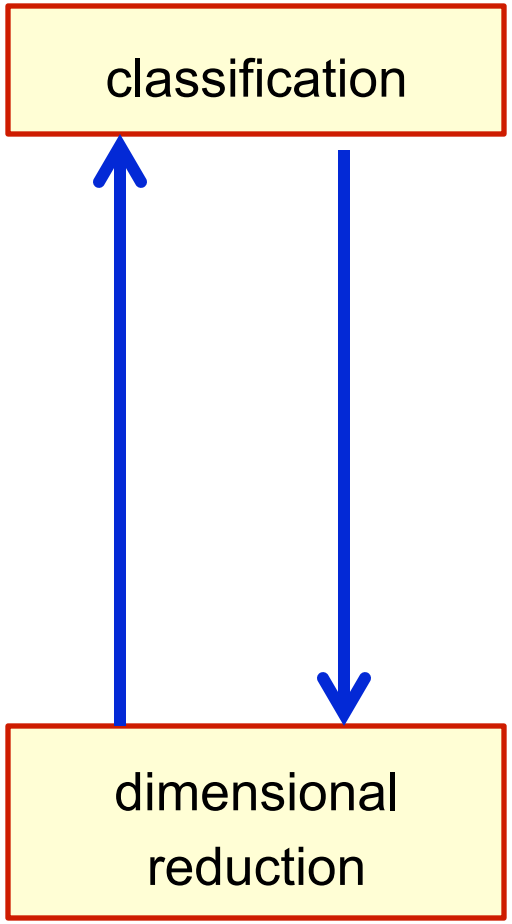
$$P(y_n|\boldsymbol{x}_n) = \frac{1}{Z_1} \exp\{\boldsymbol{\alpha} \cdot f(\boldsymbol{x}_n, y_n)\}$$

Still cannot handle sparse features!

Objective function: $P(Y|\boldsymbol{X}) = \prod_n P(y_n|\boldsymbol{x}_n) \prod_c P(y_{c_1}, y_{c_2})$

Proposed Model

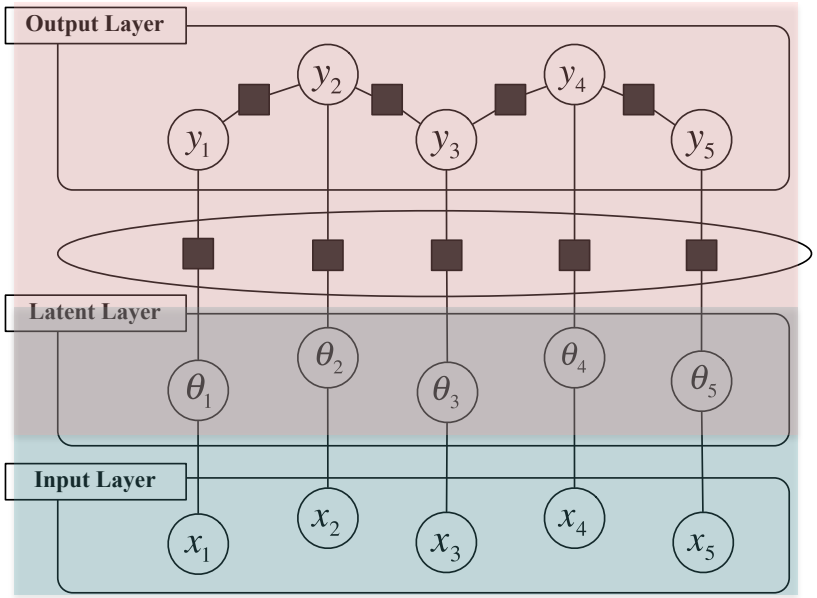




Objective function:

$$O(\lambda) = \sum_n \log P(y_n|\boldsymbol{\theta}_n, \boldsymbol{x}_n) + \sum_c \log P(y_{c_1}, y_{c_2})$$
$$= \sum_n \boldsymbol{\alpha} f(\boldsymbol{\theta}_n, y_n) + \sum_c \beta g(y_{c_1}, y_{c_2})$$
$$+ \sum_n \sum_l \log \sum_k \theta_{nk} \boldsymbol{\Omega}_{k,l,x_{nl}} - \log Z$$

Learning Algorithm



Input: a feature matrix X, learning rate η
Output: estimated parameters λ

Initialize $\alpha, \beta, \theta, \mu, \phi$ randomly;
Initialize $\delta \leftarrow 1$;

repeat

1. Calculate $P(k_{nl}|x_n, \lambda_n)$ according to Eq. 8;
   Update $\theta, \mu, \delta, \phi$ according to Eq. 9-12;
2. Call LBP to calculate $E[\sum_n f(\theta_n, y_n)]$ and $E[\sum_c g(y_{c_1}, y_{c_2})]$;
   Call LBP to calculate $E_{P_\alpha(\boldsymbol{y}|\boldsymbol{\theta})}[\sum_n f(\theta_n, y_n)]$ and $E_{P_\beta(\boldsymbol{y})}[\sum_c g(y_{c_1}, y_{c_2})]$;
   Calculate $\frac{\partial O(\alpha,\beta)}{\partial \alpha}$ and $\frac{\partial O(\alpha,\beta)}{\partial \beta}$ according to Eq. 14;
   $\alpha_{\text{new}} = \alpha_{\text{old}} + \eta \frac{\partial O(\alpha,\beta)}{\partial \alpha}$;
   $\beta_{\text{new}} = \beta_{\text{old}} + \eta \frac{\partial O(\alpha,\beta)}{\partial \beta}$

until *Convergence* ;

Algorithm 1: Learning algorithm of SparseFGM.

Learning Algorithm (cont.)

- Update the dimensional reduction parameters
  - The remaining part of SparseFGM could be regarded as a mixture generative model, with the log-likelihood

1 $$O(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\delta}) = \sum_n \sum_{l_1} \log \sum_k \theta_{nk} \frac{\exp\{\frac{-(r_{nl_1} - \mu_{kl})^2}{2\delta_k^2}\}}{\delta_k \sqrt{2\pi}} + \sum_n \sum_{l_2} \log \sum_k \theta_{nk} \phi_{kl_2 r_{nl_2}}$$

  - Jensen's inequality tells us that

1 $$\geq \sum_{n,l_1,k} P(k_{nl}|\boldsymbol{x}_n, \lambda_n)[\log \theta_{nk} - \frac{(r_{nl_1} - \mu_{kl_1})^2}{2\delta^2} - \log(\delta\sqrt{2\pi P(k_{nl}|\boldsymbol{x}_n, \lambda_n)})] + \sum_{n,l_2,k} P(k_{nl}|\boldsymbol{x}_n, \lambda_n) \times [\log \theta_{nk} + \log \phi_{kl_2 r} - \log P(k_{nl}|\boldsymbol{x}_n, \lambda_n)]$$

$$\begin{aligned} \theta_{nk} &= \frac{\sum_l P(k_{nl}|\boldsymbol{x}_n, \lambda_n)}{\sum_l \sum_{k_{nl}} P(k_{nl}|\boldsymbol{x}_n, \lambda_n)} + \alpha_k y_n \\ \mu_{kl} &= \frac{\sum_n P(k_{nl}|\boldsymbol{x}_n, \lambda_n) r_{nl}}{\sum_n P(k_{nl}|\boldsymbol{x}_n, \lambda_n)} \\ \delta_k^2 &= \frac{\sum_n \sum_l (r_{nl} - \mu_{kl})^2}{N \times L_1} \\ \phi_{klr} &= \frac{\sum_n P(k_{nl}|\boldsymbol{x}_n, \lambda_n)}{\sum_n \sum_r P(k_{nl}|\boldsymbol{x}_n, \lambda_n)} \end{aligned}$$

  - Derivate with respect to each parameters, set them to zero, and get the update equations.

Learning Algorithm (cont.)

- Update the classification parameters
  - New log-likelihood

$$O(\boldsymbol{\alpha}, \beta) = \sum_n \boldsymbol{\alpha} f(\boldsymbol{\theta}_n, y_n) + \sum_c \beta g(y_{c_1}, y_{c_2}) - \log Z$$

  - Adopt a gradient descent method to optimize the new log-likelihood

$$\frac{\partial O(\boldsymbol{\alpha}, \beta)}{\partial \boldsymbol{\alpha}} = \boldsymbol{E}[\sum_n f(\boldsymbol{\theta}_n, y_n)] - \boldsymbol{E}_{P_\alpha(\boldsymbol{y}|\boldsymbol{\theta})}[\sum_n f(\boldsymbol{\theta}_n, y_n)]$$

$$\frac{\partial O(\boldsymbol{\alpha}, \beta)}{\partial \beta} = \boldsymbol{E}[\sum_c g(y_{c_1}, y_{c_2})] - \boldsymbol{E}_{P_\beta(\boldsymbol{y})}[\sum_c g(y_{c_1}, y_{c_2})]$$

Input: a feature matrix X, learning rate η
Output: estimated parameters λ

Initialize $\boldsymbol{\alpha}, \beta, \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\phi}$ randomly;
Initialize $\boldsymbol{\delta} \leftarrow 1$;
repeat
  Update $P(k_{nl}|\boldsymbol{x}_n, \lambda_n), \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\delta}, \boldsymbol{\phi}$ according to Eq 4.16-4.20;
  Call LBP to calculate $\boldsymbol{E}[\sum_n f(\boldsymbol{\theta}_n, y_n)]$ and $\boldsymbol{E}[\sum_c g(y_{c_1}, y_{c_2})]$;
  Call LBP to calculate $\boldsymbol{E}_{P_\alpha(\boldsymbol{y}|\boldsymbol{\theta})}[\sum_n f(\boldsymbol{\theta}_n, y_n)]$ and $\boldsymbol{E}_{P_\beta(\boldsymbol{y})}[\sum_c g(y_{c_1}, y_{c_2})]$;
  $\boldsymbol{\alpha}_{\text{new}} = \boldsymbol{\alpha}_{\text{old}} + \eta \frac{\partial O(\boldsymbol{\alpha}, \beta)}{\partial \boldsymbol{\alpha}}$;
  $\beta_{\text{new}} = \beta_{\text{old}} + \eta \frac{\partial O(\boldsymbol{\alpha}, \beta)}{\partial \beta}$
until *Convergence*;

Algorithm 1: Learning algorithm of SparseFGM.

Theoretical Analysis

THEOREM 3.1. *The maximal log-likelihood of SparseFGM is no less than FGM when* $K \geq \max\{|t_n|\}$

$t_n = \{l | x_{nl} \neq /\}$

Proof. Assuming we have a parameter configuration of FGM $\widetilde{\lambda} = (\widetilde{\alpha}, \widetilde{\beta})$, which maximizes FGM's objective function.

$$\widetilde{O}(\widetilde{\lambda}) = \sum_n \widetilde{\alpha} f(\boldsymbol{r}_n, y_n) + \sum_c \widetilde{\beta} g(y_{c_1}, y_{c_2}) - \log \widetilde{Z}$$

Let $\theta_{nk} = \frac{\widetilde{\alpha}_{t_{nk}} r_{t_{nk}}}{Z_k}$ for $k \leq |t_n|$ and $\theta_{nk} = 0$ for $k \geq |t_n|$, where Z_k is a normalization term. Also let $\alpha_k = Z_k$, $\beta = \widetilde{\beta}$, and for each $1 \leq k \leq K$, we select a particular distribution as Ω_k such that $\forall n$, we have $\sum_k \theta_{nk} \Omega = 1$. Thus, we have

1

$$\begin{aligned}\sum_n \alpha f(\boldsymbol{\theta}_n, y_n) &= \sum_n y_n \sum_k \alpha_k \theta_{nk} \\ &= \sum_n y_n \sum_{i=1}^{t(n)} Z_i \times \frac{\widetilde{\alpha}_{t_{nk}} r_{t_{nk}}}{Z_k} \\ &= \sum_n \widetilde{\alpha} f(\boldsymbol{r}_n, y_n)\end{aligned}$$

2

$$\begin{aligned}\log Z &= \sum_n \sum_{y_n} \alpha f(\boldsymbol{\theta}_n, y_n) + \sum_c \sum_{y_{c_1}, y_{c_2}} \beta g(y_{c_1}, y_{c_2}) \\ &= \sum_n \sum_{y_n} \widetilde{\alpha} f(\boldsymbol{r}_n, y_n) + \sum_c \sum_{y_{c_1}, y_{c_2}} \widetilde{\beta} g(y_{c_1}, y_{c_2}) \\ &= \log \widetilde{Z}\end{aligned}$$

3

$$\begin{aligned}O(\lambda) &= \sum_n \alpha f(\boldsymbol{\theta}_n, y_n) + \sum_c \beta g(y_{c_1}, y_{c_2}) \\ &+ \sum_n \sum_l \log \sum_k \theta_{nk} \Omega_{k,l,r_{nl}} - \log Z \\ &= \sum_n \widetilde{\alpha} f(\boldsymbol{r}_n, y_n) + \sum_c \widetilde{\beta} g(y_{c_1}, y_{c_2}) - \log \widetilde{Z} \\ &= \widetilde{O}(\widetilde{\lambda})\end{aligned}$$

1、2、3 indicate

$$\max_\lambda \{O(\lambda)\} \geq O(\lambda) = \widetilde{O}(\widetilde{\lambda}) = \max_\lambda \{\widetilde{O}(\lambda)\}$$

Experiments

Setting

- Experiments
  - Is our model effective?
  - How do different diabetes complications associate with each lab test?
  - Can we forecast all diabetes complications well?
- Comparison Methods
  - SVM (model I)
  - FGM (model II)
  - FGM+PCA (an alternative method to handle feature sparseness)
  - SparseFGM (our approach)

Experimental Results

HTN: hypertension, **CHD:** coronary heart disease, **HPL:** hyperlipidemia

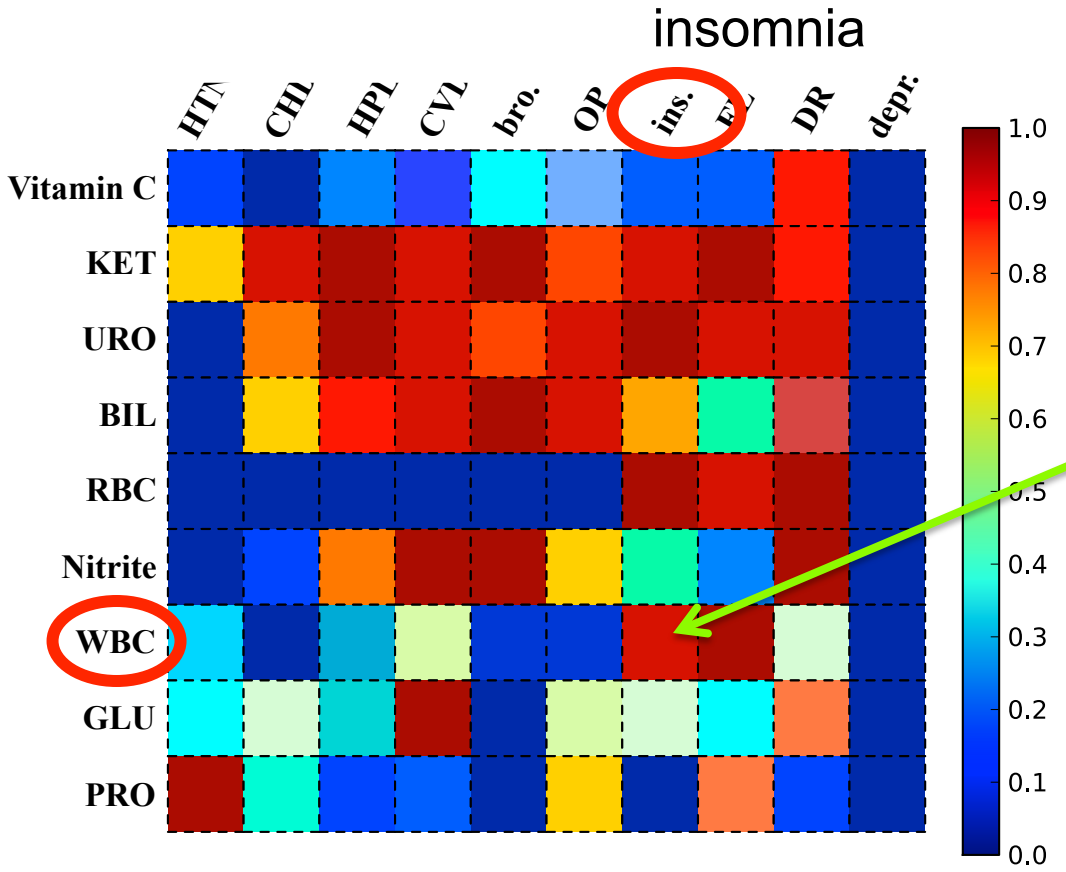
SVM and FGM suffer from feature sparseness. **-59.9%** in recall.

FGM vs. FGM + PCA (increase **+40.3%** in recall)

PGM+PCA vs. SparseFGM (increase **+13.5%** in F1)

cation	Method	Precision	Recall	F1
HTN	SVM	0.3804	0.4789	0.4241
	FGM	0.5666	0.4959	0.5075
	FGM+PCA	0.5741	0.3284	0.4178
	SparseFGM	0.4714	0.6319	**0.5400**
D	SVM	0.2132	0.0636	0.0980
	FGM	0.6264	0.1369	0.2247
	FGM+PCA	0.2425	0.8367	0.3761
	SparseFGM	0.2522	0.7972	**0.3832**
L	SVM	0.2208	0.0460	0.0761
	FGM	0.6557	0.0591	0.1084
	FGM+PCA	0.2047	0.8035	0.3262
	SparseFGM	0.2796	0.8396	**0.4195**

Association Pattern Illustration

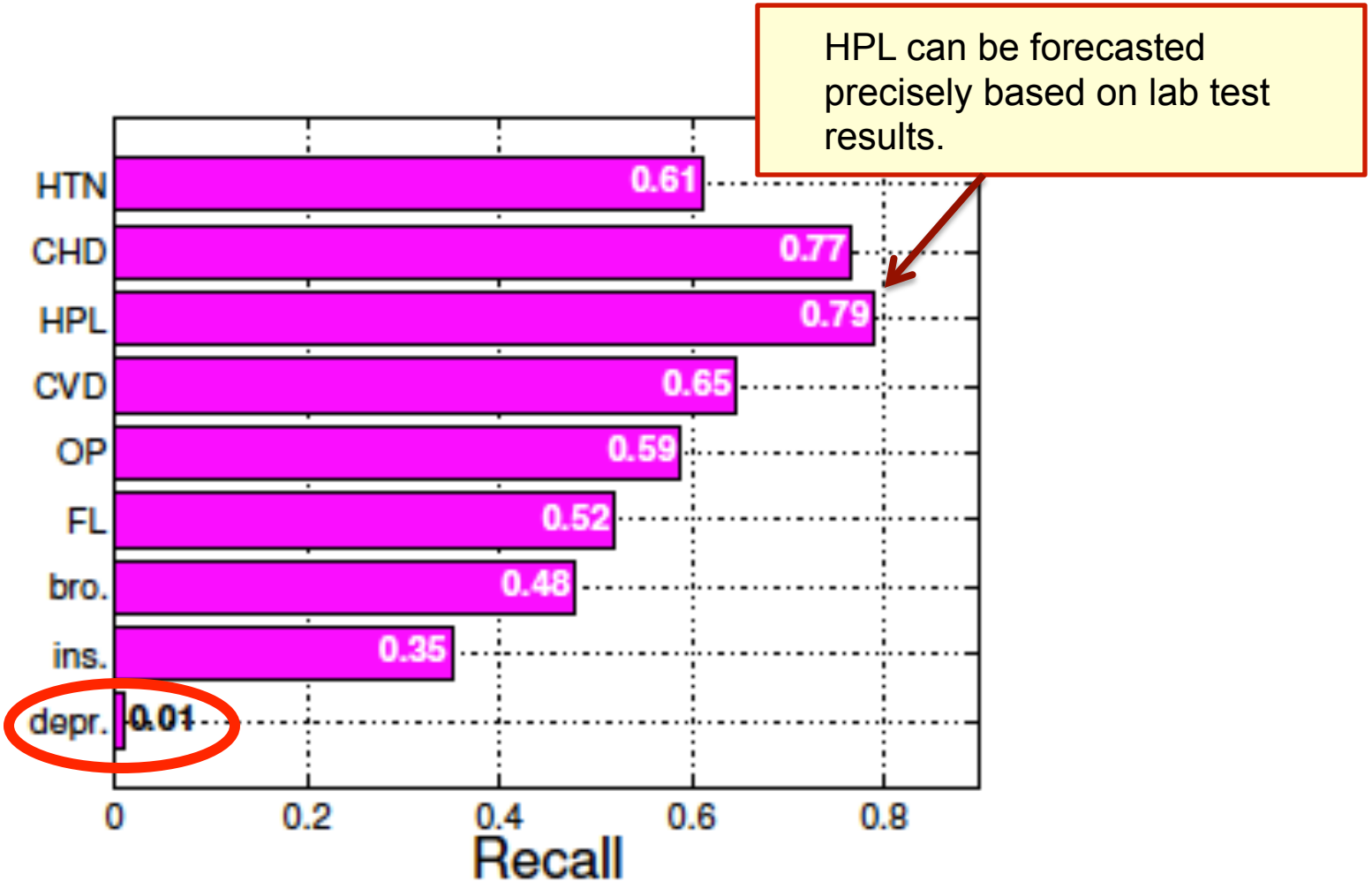


WBC in the urine causes frequent voiding -> no good sleep at night

Association score: $AS(c,e) = \sum_k \alpha_k^c \theta_{ek}$

c: complication, *e*: lab test

Can We Forecast All Diabetes Complications?



Conclusion

- We study the problem of forecasting diabetes complications.
- We propose a graphical model which integrates dimensional reduction and classification into a uniform framework.
- We further study the underlying associations between different diabetes complications and lab test types.

Thanks!

Q&A?

@ Yang Yang

http://yangy.org/