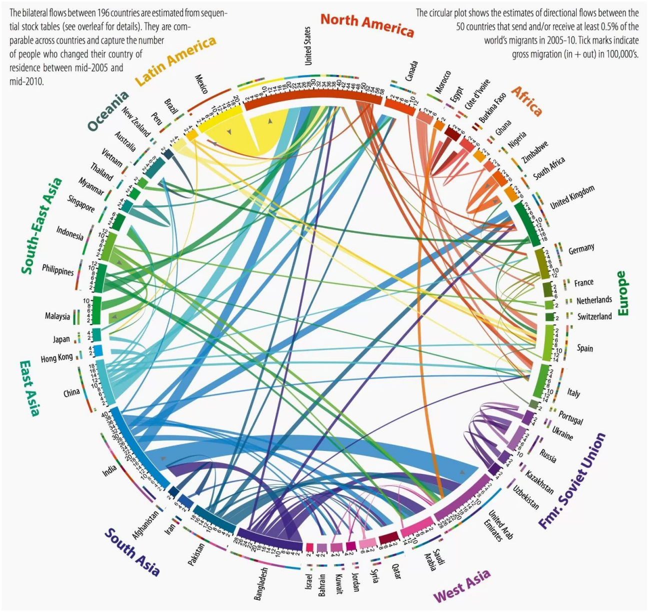
Migration Prediction

Team: Yellow Fish

資工三 {于建民、鄭筱樺、林祐萱}
B039020 {22, 24, 55}

What's "migration" prediction?

- Given total number of immigrants and emigrants of each country, can we predict the detailed migration from a country to another?
- To look deeper, can we predict migration of country that has no data recorded, such as North Korea or Taiwan?
    - First use some Supervised method to predict the total immigrant & emigrant
    - Then use our method to predict detailed migration.



Future work

Move forward to other fields, for example:

- The migration between HTC and Apple.
- Employee migration between companies.
- etc.

Data collection

Data collection – Feature

Features: total 741 = 738 individual + 3 inter-country

Individual country:

- Economy Development (Worldbank)
- Health, Society index (Worldbank)
- Climate (Worldbank)
- Religion (United nation)

Inter-country:

- Language similarity (CEPII research)
- Geometric distance (CEPII research)
- Trading (Comtrade)

Data collection – Validation

Migration data:

- From: United nation Department of Economic and Social Affairs
- Year: 2005, 2010, 2015

Feature extraction

Feature extraction – PCA

- Linear extraction
- Fill in missing values with mean
- All data is numerical (no categorical data)
- Tool : scikit-learn

Feature extraction – Autoencoder

- One hidden layer
    - input: x, output: y
    - $x' = sigmoid(w\ x)$
    - $y = sigmoid(w_2\ x')$
- Loss function: $(x - y)^2$
- Extracted features: $x' = sigmoid(wx)$
- tool : Tensorflow

Basic method (for comparison)

Baseline

Notation: let $x_{i,j}$ be the migration number predicted move from C_i to C_j, where C_i and C_j are countries.

Concept: Solve the linear equation.

$$\forall i, \sum_j x_{ij} = Mout_i \qquad \forall j, \sum_i x_{ij} = Min_j$$

Solution: Proportional allocation

$$\forall C_i, x_{i,i} = 0$$

$$\forall i, j \text{ where } i \neq j, \text{ we have } \quad x_{i,j} = \frac{Mout_i}{\sum_{k \neq i} Mout_k} Min_j$$

Supervised (not baseline)

Concept: Use year A with features & real migration number to predict year B with only features.

Solution: SVM (Support vector machine) with feature extraction method mentioned before.

Unsupervised method

Linear score function & Quadratic Programming (1)

- After dimension reduction, we get k-dimension features f_{ij} for each pair of countries.
- **linear score function** : $S_{ij} = \sum_k w_k f_{ijk} = w^T f_{ij}$
- **objective function** : $min\, f = \sum_i ((\sum_j S_{ij}) - O_i)^2 + \sum_j ((\sum_i S_{ij}) - I_j)^2$,
  subject to $(f_{ij})^T w \geq 0, \forall i, j$ and $(f_{ii})^T w = 0, \forall i$

$$\begin{bmatrix} S_{11} & S_{12} & S_{13} & \dots & S_{1n} \\ S_{21} & S_{22} & S_{23} & \dots & S_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{n1} & S_{n2} & S_{n3} & \dots & S_{nn} \end{bmatrix} \begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_n \end{bmatrix}$$

$$\begin{bmatrix} I_1 & I_2 & I_3 & \dots & I_n \end{bmatrix}$$

Linear score function & Quadratic Programming (2)

- Some mathematical derivation : let's focus on the left part

$\sum_i((\sum_j S_{ij}) - O_i)^2$

$= \sum_i((\sum_j S_{ij})^2 - 2O_i(\sum_j S_{ij}) + O_i^2)$

$= \sum_i((\sum_j w^T f_{ij})^2 - 2O_i(\sum_j w^T f_{ij}) + O_i^2)$

$= \sum_i(w^T M_i w + (-2O_i \sum_j f_{ij})^T w + O_i^2)$

, where M_i is a $k * k$ diagonal matrix with $M_{i,kk} = (\sum_j f_{ijk})^2$

$= w^T (\sum_i M_i) w + (-2 \sum_j O_i \sum_j f_{ij})^T w + \sum_i O_i^2$

Linear score function & Quadratic Programming (3)

- After omitting the constants,
  $f = w^T(\sum_i M_i + \sum_j M_j)w + (-2(\sum_j O_i \sum_j f_{ij} + \sum_i I_j \sum_i f_{ij}))^T w$
  and the problem become a **QP**(quadratic programming) problem with linear constraints!
- Since $(\sum_i M_i + \sum_j M_j)$ is **positive definite**, the problem can be solved in polynomial time.
- tool - cvxopt

Experiment result

Evaluation method

- L1 loss devided by #entries

$$loss = \frac{\sum_{x \in X} |f(x) - y|}{|X|}$$

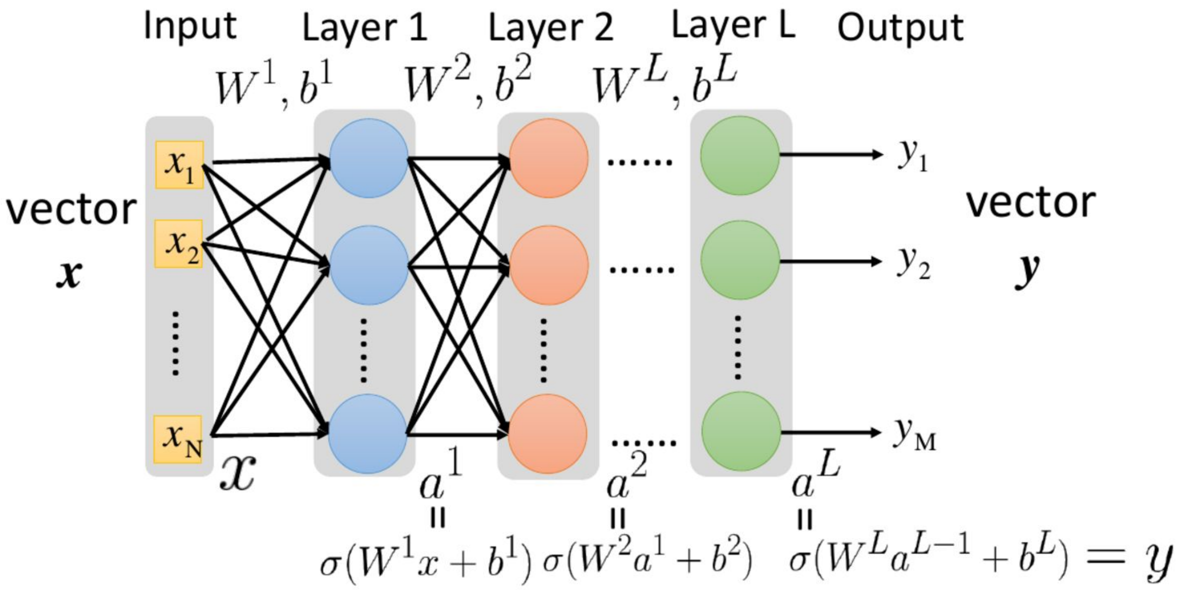
Result

method	loss	Feature extraction	parameters
Baseline	20704.3	x	
Supervised	6967.5	PCA	dim=100, kernel='rbf'
Supervised	6965.7	AE	dim=100, kernel='rbf'
QP	16028.5	PCA	dim=50
QP	18045.3	PCA	dim=100
QP	18045.3	PCA	dim=200

What’s next?

Try non-linear score function(1/2)

Use neural network to turn linear score function into nonlinear.



Try non-linear score function (2/2)

- Linear score function:

$$S = \Sigma w_1 x$$

- Nonlinear score function:

$$z_i^{(l)} = \sigma(w_{l-1} a_i^{(l-1)})$$

  - Activation function: could be $tanh, sigmoid, Relu$

- Objective function: the same as with linear score function

$$\Sigma_i((\Sigma_j S_{ij}) - O_i)^2 + \Sigma_j((\Sigma_i S_{ij}) - I_j)^2$$

- Train: (Stochastic) Gradient Descent

Reference

Reference

1. Supervised method: Scikit-learn SVM
2. Data: Worldbank
3. Data: United nation
4. Data: CEPII research
5. Data: Comtrade
6. Data: United nation Department of Economic and Social Affairs
7. PCA: Scikit-learn PCA
8. AE: tensorflow
9. QP: cvxopt

Thank you!