Chapter 5
Embedding

5.1 Overview

The clinically meaningful representations of medical concepts and patients are the key to health analytic applications. Standard machine learning approaches directly construct features mapped from raw data (e.g., ICD or CPT codes) or utilize some ontology mapping such as SNOMED codes. With deep neural networks' successes, people have focused on learning concept representation, namely *embedding vectors*. In this chapter, we present a set of neural network models called *embedding methods* to represent medical concepts (e.g., diagnoses, medications, and procedures) based on co-occurrence patterns in longitudinal electronic health records. The intuition behind the embedding methods is to map medical concepts co-occurring closely in EHR data to similar embedding vectors (i.e., the embedding distance between related medical concepts is small). Depending on how we define the co-occurrence, different embedding methods are proposed:

- Word2Vec [110] is based on the proximity of medical codes appearing in the records;
- GloVe [117] is a unsupervised learning algorithm for word embedding based on word co-occurrence statistics.
- t-SNE embedding [159] is a nonlinear dimensionality reduction method for visualizing high-dimensional data in 2D space.
- Med2Vec [24] leverages two-level hierarchical structures, namely clinical visits over time and co-occurrence within a visit;
- MIME [31] utilizes a three-level hierarchy, namely visits, diagnosis codes, then treatment codes.

Note that Word2Vec, GloVe, and t-SNE are general embedding methods, while Med2Vec and MIME are specifically designed to handle electronic health record data.

5.2 Word2Vec

Word2Vec learns real-valued multi-dimensional vectors that capture relations between words by training on the massive text datasets [110]. The trained real-valued vectors will have similar values for similar words such as dog and cat or would and could, but distinct values for words that are not. Like natural language text, electronic health records can be seen as a sequence of "words" (i.e., medical codes such as diagnoses, medications, and procedures). Once the sequences of medical code are formed, we can map raw medical codes (e.g., ICD9, CPT) into embedding vectors using Word2Vec.

5.2.1 Idea and Formulation of Word2Vec

Word2Vec or any embedding method aims to learn a vector representation for each word such that closely related words will have similar vector representation. Words in healthcare applications can be medical codes in EHRs. Figure 5.1 depicts a motivating example for using a better representation of medical concepts. Figure 5.1a shows one-hot encoding of N unique diagnoses using N-dimensional vectors. More specifically, each diagnosis's corresponding dimension will be 1, and all the other $N - 1$ dimensions are 0. One-hot encoding is simple but not an effective representation for machine learning. The difference between related diagnoses (like bronchitis and pneumonia) is the same as the difference between unrelated diagnoses (like pneumonia and obesity). In fact, all pairs of different medical diagnoses will have the same distance to each other (i.e., Euclidean distance of $\sqrt{2}$). Figure 5.1b shows a Word2Vec representation in that bronchitis and pneumonia share similar values compared to other diagnoses. Moreover, by

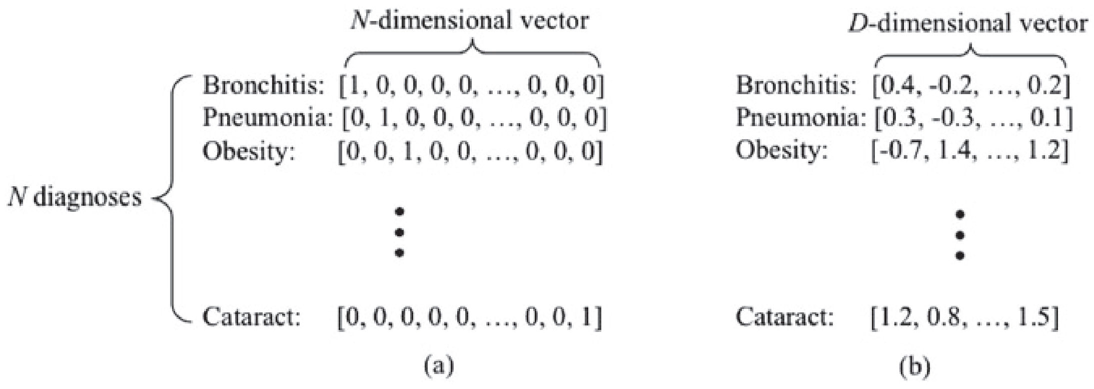

N-dimensional vector
Bronchitis: [1, 0, 0, 0, 0, ..., 0, 0, 0]
Pneumonia: [0, 1, 0, 0, 0, ..., 0, 0, 0]
Obesity: [0, 0, 1, 0, 0, ..., 0, 0, 0]
N diagnoses
Cataract: [0, 0, 0, 0, 0, ..., 0, 0, 1]
(a)
D-dimensional vector
Bronchitis: [0.4, -0.2, ..., 0.2]
Pneumonia: [0.3, -0.3, ..., 0.1]
Obesity: [-0.7, 1.4, ..., 1.2]
Cataract: [1.2, 0.8, ..., 1.5]
(b)

Fig. 5.1 Two different representation of diagnoses. Typically, raw data dimensionality N(>10,000) used in one-hot embedding is much larger than concept dimensionality D(∼100) in Word2Vec. More importantly, Word2Vec will organize similar word embeddings closer to each other in order to capture word relationship while one-hot vector can not. (**a**) One-hot encoding for diagnoses. (**b**) A better representation of diagnoses

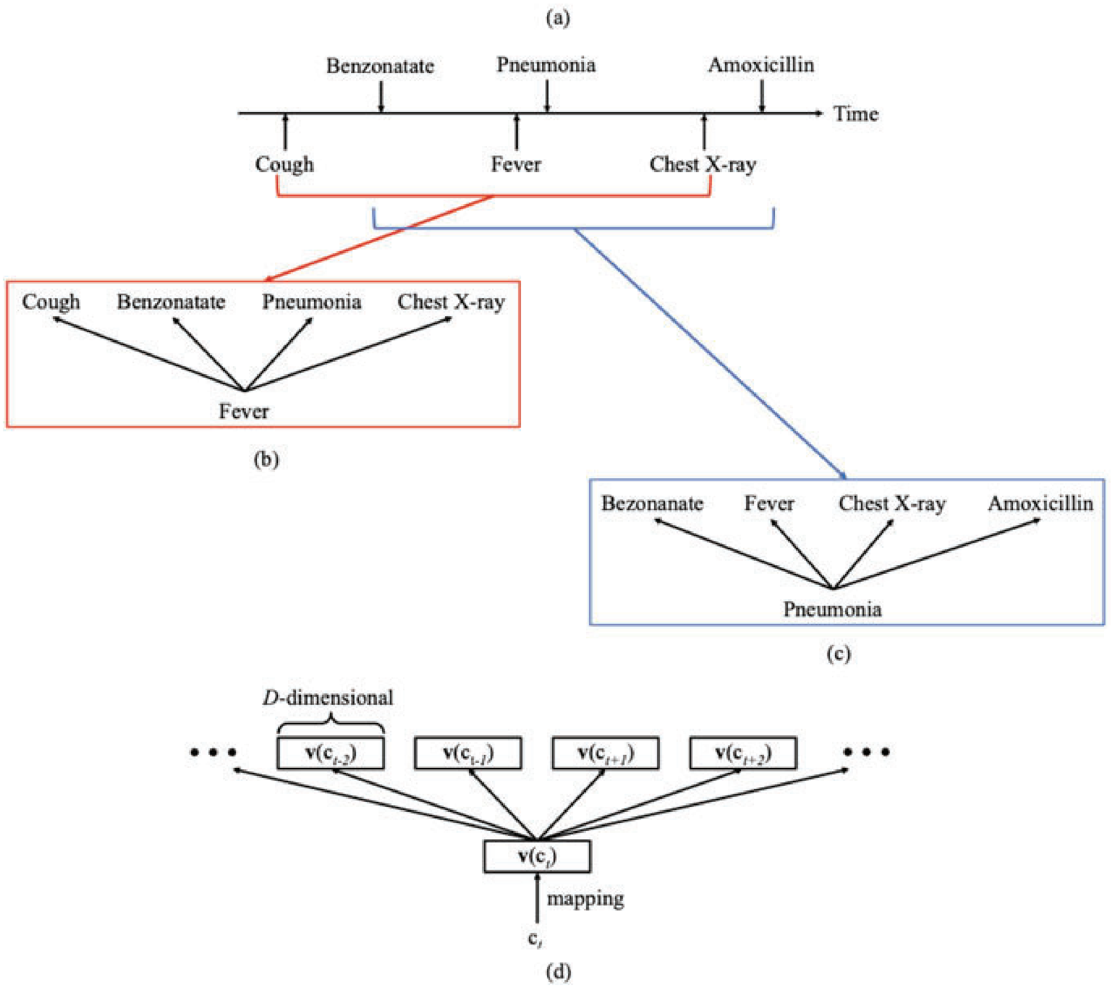


Fig. 5.2 Idea behind Word2Vec and the model architecture. (**a**) Patient medical records on a timeline. (**b**) Predicting neighboring medical concepts given *Fever*. (**c**) Predicting neighboring medical concepts given Pneumonia. (**d**) Model architecture of Skip-gram

using Word2Vec, we will better represent diagnoses and medications and procedures as multi-dimensional real-valued vectors by capturing the latent relations between them.

Next, let us look into the mathematical formulation of Word2Vec. Figure 5.2a is an example of a patient medical record in temporal order. Word2Vec assumes the meaning of a word is determined by its context (i.e., its neighboring words). Therefore, given a sequence of words, Word2Vec picks a target word and tries to predict its neighboring words. Figure 5.2b shows fever as the target word, and the other nearby words are its neighboring words. Then we slide a context window, pick the next target, and make the same context prediction shown by Fig. 5.2c. Since the goal of Word2Vec is to learn the vector representation of words, we need to map words to D-dimensional embedding vectors, where D is a user-defined value (typically between 50 and 1000 for text data). Therefore the actual prediction is conducted with the learned embedding vectors. Formally c_t denotes the word at the t-th timestep, $\boldsymbol{v}(c_t)$ the embedding vector that represents c_t. The goal of Word2Vec is to maximize the following average log probability,

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-w\le j\le w, j\ne 0}\log p(c_{t+j}|c_t)$$

where $p(c_{t+j}|c_t) = \exp[\boldsymbol{v}(c_{t+j})^\top \boldsymbol{v}(c_t)]/\sum_{c=1}^{N}\exp[\boldsymbol{v}(c)^\top \boldsymbol{v}(c_t)]$ and T is the length of the sequence of medical concepts, w the size of the context window, c_t the target medical concept at timestep t, c_{t+j} the neighboring medical concept at timestep $t + j$, $\boldsymbol{v}(c)$ the vector that represents the medical concept c, N the number of unique words. The size of the context window w is typically set to 5, giving us 10 neighboring words surrounding each target word. Note that the conditional probability is expressed as a softmax function. By maximizing the softmax score of the inner product of embeddings between each target word and its neighboring words, By maximizing this softmax function, Word2Vec learns real-valued vectors that capture the co-occurrence relations between words.

To compute Word2Vec, the denominator $\sum_{c=1}^{N}\exp[\boldsymbol{v}(c)^\top \boldsymbol{v}(c_t)]$ can be too expensive to compute as it sums over all words. A practical solution is to use *negative sampling*. The idea is to transform the word embedding problems as a binary classification task. The positive samples are the word pairs in the context window (i.e., c_{t+j}, c_t for $-w \le j \le w, j \ne 0$). And the negative samples are randomly sampled (i.e., pairing random words with the target word c_t). Then the objective function can be rewritten as

$$\arg\max \prod_{(c_w,c_t)\in D} p(y=1|c_w,c_t) \prod_{(c_w,c_t)\notin D} p(y=0|c_w,c_t) =$$

$$\arg\max \sum_{(c_w,c_t)\in D}\log\frac{1}{1+e^{-\boldsymbol{v}(c_w)^\top \boldsymbol{v}(c_t)}} + \sum_{(c_w,c_t)\notin D}\log(1-\frac{1}{1+e^{-\boldsymbol{v}(c_w)^\top \boldsymbol{v}(c_t)}})$$

where $(c_w, c_t) \in D$ indicates the positive samples (i.e., the word pairs in the dataset), $(c_w, c_t) \notin D$ the negative samples (i.e., the word pairs of random word c_w and the target word c_t). The optimization procedure for the new objective function is equivalent to logistic regression classifier, which is more efficient than the original setting of Word2Vec. Note that the word2vec paper [110] present two algorithms for word embeddings, namely *skip gram* and *continuous bag of words*. They are very similar and we presented the skip gram formulation.

5.2.2 t-Distributed Stochastic Neighbor Embedding (t-SNE)

Deep learning models often process high-dimensional data (e.g., one-hot vectors of medical codes) and produce high-dimensional data (e.g., hidden states of neural networks). It is desirable to visualize high-dimensional data to understand patterns and relationships of that input and immediate output of neural networks. The t-distributed stochastic neighbor embedding (t-SNE) is a powerful and popular method for visualizing high-dimensional data (e.g., hidden states of a neural network) [159]. More specifically, the input to t-SNE is a high-dimensional data point $\mathbf{x}_i$, and the corresponding output from t-SNE is a low-dimensional (2D) data point $\mathbf{y}_i$. Traditional dimensionality reduction methods such as principal component analysis (PCA) can also produce 2-dimensional embedding for visualization. One major drawback of PCA is that it tries to preserve pairwise distance, dominated by faraway data points, while ignoring the small local variations. However, in real data, the relations of faraway points are often not reliable hence less important. On the other hand, relationships among nearby points are more important and more reliable, which should be the focus of the visualization method.

By highlighting the local relationships, t-SNE becomes a good dimensionality reduction algorithm that can display patterns and clusters in the data in a visually appealing way. Let us define p_{ij} and q_{ij} are the joint probabilities between a pair of points i and j for high-dimensional low-dimensional space, respectively. The objective of t-SNE is to minimize the Kullback–Leibler (KL) divergence between joint probability p_{ij} and q_{ij}:

$$KL(P||Q) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

where the joint probability p_{ij} for the high dimensional data is

$$p_{ij} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)}{\sum_{k \neq l} \exp(-\|\mathbf{x}_k - \mathbf{x}_l\|^2/2\sigma^2)}$$

σ is the bandwidth of the Gaussian kernel, and the joint probability q_{ij} for the low dimensional (2D) data uses a Student t-distribution with one degree of freedom

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|)^{-1}}{\sum_{k \neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|)^{-1}}$$

Additional optimization is introduced to the input joint distribution p_{ij}. Instead of using joint probability, we use a conditional probability

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

where we can set different bandwidth σ_i for different data point i. This adaptive bandwidth turns out to be important as it can handle different density distribution of input data. The joint distribution p_{ij} is approximated by $p_{ij} = (p_{j|i} + p_{i|j})/2$.

The t-SNE method has been extensively applied to visualize complex structured and high-dimensional health data. For example, in [1], t-SNE was used to visualize

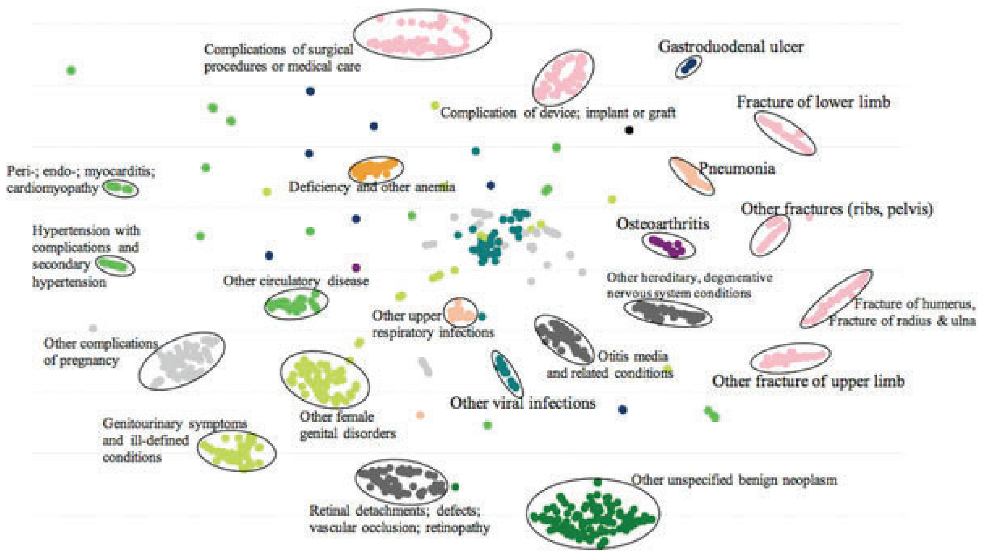


Fig. 5.3 t-SNE scatterplots of medical concepts trained by GRAM [28]

molecularly distinct and clinically relevant prognostic tumor subpopulations from mass spectrometry imaging data. Besides imaging data, t-SNE has been used to visualize genomic data. A particular example given by Taskesen et al. [151] shows when projecting tissue samples in a 2D-map (tSNE-map) using the gene expression profiles, t-SNE could retain local similarities between samples at the cost of retaining the similarities between dissimilar samples. As a result, t-SNE better preserves local (dis)similarities as they are not condensed due to the large dissimilarities in the data set. Moreover, t-SNE was also used to visualize learned phenotypes from EHR data. For example, in [94], t-SNE was used to project learned features into two-dimensional embeddings. It is easy to visually identify subpopulations of gout and leukemia that might have pathophysiologic differences.

The following examples show how researchers used t-SNE to evaluate the performance of their phenotyping algorithm visually. Figure 5.3 used t-SNE to represent the final representations of 2000 randomly chosen diseases learned by GRAM [28] for sequential diagnoses prediction using EHR data. Here t-SNE provides an intuitive way to qualitatively assess the interpretability of the learned representations of the medical codes. The color of the dots represents the highest disease categories, and the text annotations represent the detailed disease categories in CCS multi-level hierarchy. The figure shows how learned disease representations align with domain knowledge. Another demonstration in Fig. 5.4 shows how learned medical concepts from [24] could form distinct groups.

5.2.3 *Healthcare Application of Word2Vec*

Next, we present a study using Word2Vec for heart failure prediction [26].

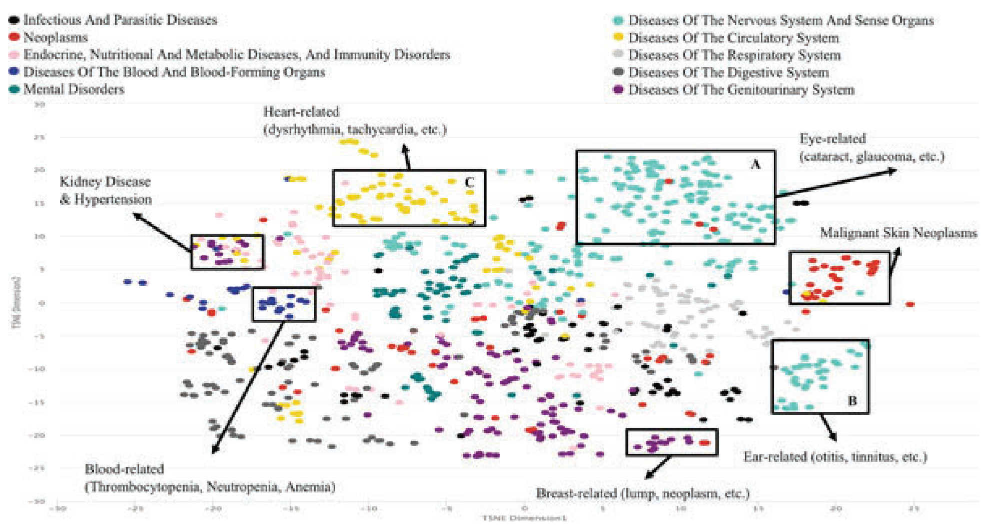
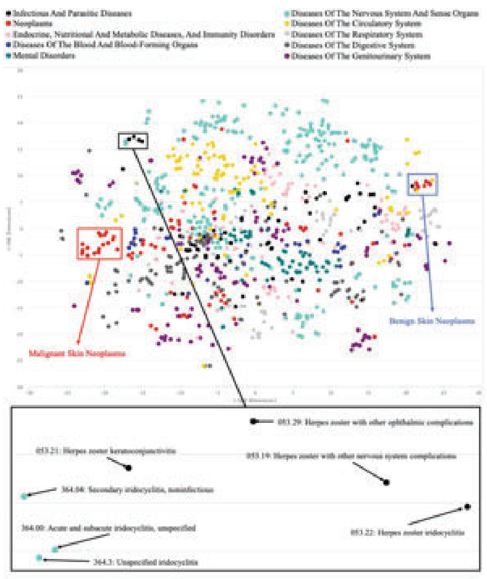


Fig. 5.4 t-SNE scatterplots of medical concepts trained by Med2vec [27]

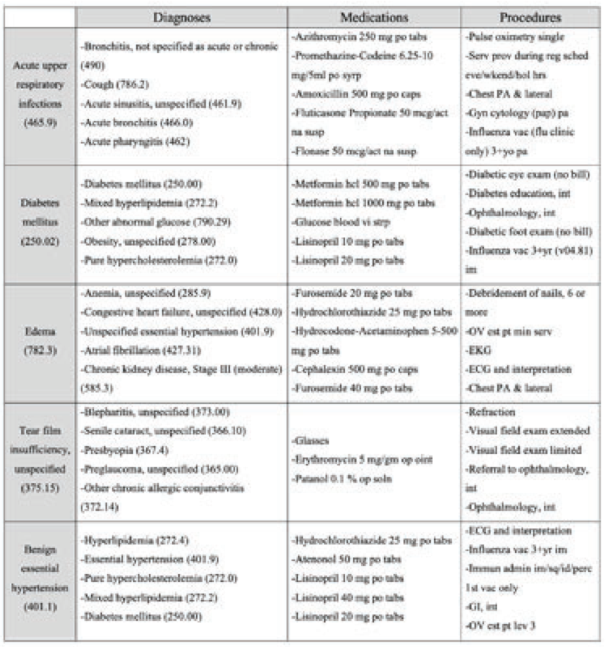
Data Data in this study were from Sutter Palo Alto Medical Foundation (Sutter-PAMF) primary care patients. Sutter-PAMF is large primary care and multispecialty group practice that has used an EHR for more than a decade. The study dataset was extracted with cases and controls identified within the interval from 2000 to 2013. The EHR data included demographics, smoking and alcohol consumptions, clinical and laboratory values, International Classification of Disease version 9 (ICD-9) codes associated with encounters, order, and referrals, procedure information in Current Procedural Terminology (CPT) codes, and medication prescription information in medical names. The dataset contained 265,336 patients with 555,609 unique clinical events in total.

Configuration of Word2Vec To apply Word2Vec to longitudinal EHR data, the authors processed medication orders, procedure orders, and problem list records of all 265,336 patients and extracted diagnosis, medication, and procedure codes assigned to each patient in a temporal order. If a patient received multiple diagnoses, medications, or procedures at a single visit, those medical codes were ordered randomly within the same timestamp. The respective number of unique diagnoses, medications, and procedures was 11,460, 17,769, and 9,370, totaling 38,599 unique medical concepts. 100-dimensional medical concept embedding vectors are used (i.e., D=100 in Fig. 5.1b).

Qualitative Assessment Figure 5.5a shows the Word2Vec diagnosis embedding vectors, where a t-SNE [109] is produced to visualize high-dimensional data in 2D space. One Thousand randomly chosen diagnoses (ICD-9 codes) are displayed in Fig. 5.5a. In it, diagnoses are generally well grouped by their disease categories. However, if some diagnoses from the same category are still quite different, they should be apart. This is shown by the red box and the blue box in Fig. 5.5a. Even though they are from the same neoplasms category, the red box indicates malignant skin neoplasms (172.X, 173.X) while the blue box indicates a benign



(a)



	Diagnoses	Medications	Procedures
Acute upper respiratory infections (465.9)	-Bronchitis, not specified as acute or chronic (490) -Cough (786.2) -Acute sinusitis, unspecified (461.9) -Acute bronchitis (466.0) -Acute pharyngitis (462)	-Azithromycin 250 mg po tabs -Promethazine-Codeine 6.25-10 mg/5ml po syrp -Amoxicillin 500 mg po caps -Fluticasone Propionate 50 mcg/act na susp -Flonase 50 mcg/act na susp	-Pulse oximetry single -Serv prov during reg sched eve/wkend/hol hrs -Chest PA & lateral -Gyn cytology (pap) pa -Influenza vac (flu clinic only) 3+yo pa
Diabetes mellitus (250.02)	-Diabetes mellitus (250.00) -Mixed hyperlipidemia (272.2) -Other abnormal glucose (790.29) -Obesity, unspecified (278.00) -Pure hypercholesterolemia (272.0)	-Metformin hcl 500 mg po tabs -Metformin hcl 1000 mg po tabs -Glucose blood vi strp -Lisinopril 10 mg po tabs -Lisinopril 20 mg po tabs	-Diabetic eye exam (no bill) -Diabetes education, int -Ophthalmology, int -Diabetic foot exam (no bill) -Influenza vac 3+yr (v04.81) im
Edema (782.3)	-Anemia, unspecified (285.9) -Congestive heart failure, unspecified (428.0) -Unspecified essential hypertension (401.9) -Atrial fibrillation (427.31) -Chronic kidney disease, Stage III (moderate) (585.3)	-Furosemide 20 mg po tabs -Hydrochlorothiazide 25 mg po tabs -Hydrocodone-Acetaminophen 5-500 mg po tabs -Cephalexin 500 mg po caps -Furosemide 40 mg po tabs	-Debridement of nails, 6 or more -OV est pt min serv -EKG -ECG and interpretation -Chest PA & lateral
Tear film insufficiency, unspecified (375.15)	-Blepharitis, unspecified (373.00) -Senile cataract, unspecified (366.10) -Presbyopia (367.4) -Preglaucoma, unspecified (365.00) -Other chronic allergic conjunctivitis (372.14)	-Glasses -Erythromycin 5 mg/gm op oint -Patanol 0.1 % op soln	-Refraction -Visual field exam extended -Visual field exam limited -Referral to ophthalmology, int -Ophthalmology, int
Benign essential hypertension (401.1)	-Hyperlipidemia (272.4) -Essential hypertension (401.9) -Pure hypercholesterolemia (272.0) -Mixed hyperlipidemia (272.2) -Diabetes mellitus (250.00)	-Hydrochlorothiazide 25 mg po tabs -Atenonol 50 mg po tabs -Lisinopril 10 mg po tabs -Lisinopril 40 mg po tabs -Lisinopril 20 mg po tabs	-ECG and interpretation -Influenza vac 3+yr im -Immun admin im/sq/id/perc 1st vac only -GI, int -OV est pt lev 3

(b)

Fig. 5.5 (**a**) Medical concept embedding vectors projected to 2D space; (**b**) Examples of diagnoses and their closest medical concepts

skin group neoplasms (216.X). The detailed figure of the red and blue boxes are in the supplementary section. As the black box shows, diagnoses from different groups are located close to one another if they are actually co-occurred often. In the black box, iridocyclitis and eye infections related to herpes zoster are closely located, which corresponds to approximately 43% of herpes zoster ophthalmicus (HZO), commonly known as shingles, patients develop iridocyclitis [154].

One important benefit of Word2Vec is that it can map different types of medical codes (e.g., diagnosis codes and procedure codes) into a common embedding space so that related medical codes of different types will be closer to each other. Figure (5.5b) shows examples of top-5 nearest neighbors from different types of medical codes to a diagnosis code. For example, embedding for the diagnosis code of "Acute upper respiratory infections" is close to "Bronchitis" (diagnosis), "Azithromycin" (antibiotic medication), "Influenza vac" (procedure), which are clinically meaningful.

Quantitative Assessment The ultimate value of embedding vectors is to generate better features for prediction or classification tasks. In this study, the task is to predict whether patients will develop heart failure (HF) based on various patient representations.

Figure 5.6 shows the AUC of various models and input feature vectors. The colors indicate the prediction results using different **patient representations**:

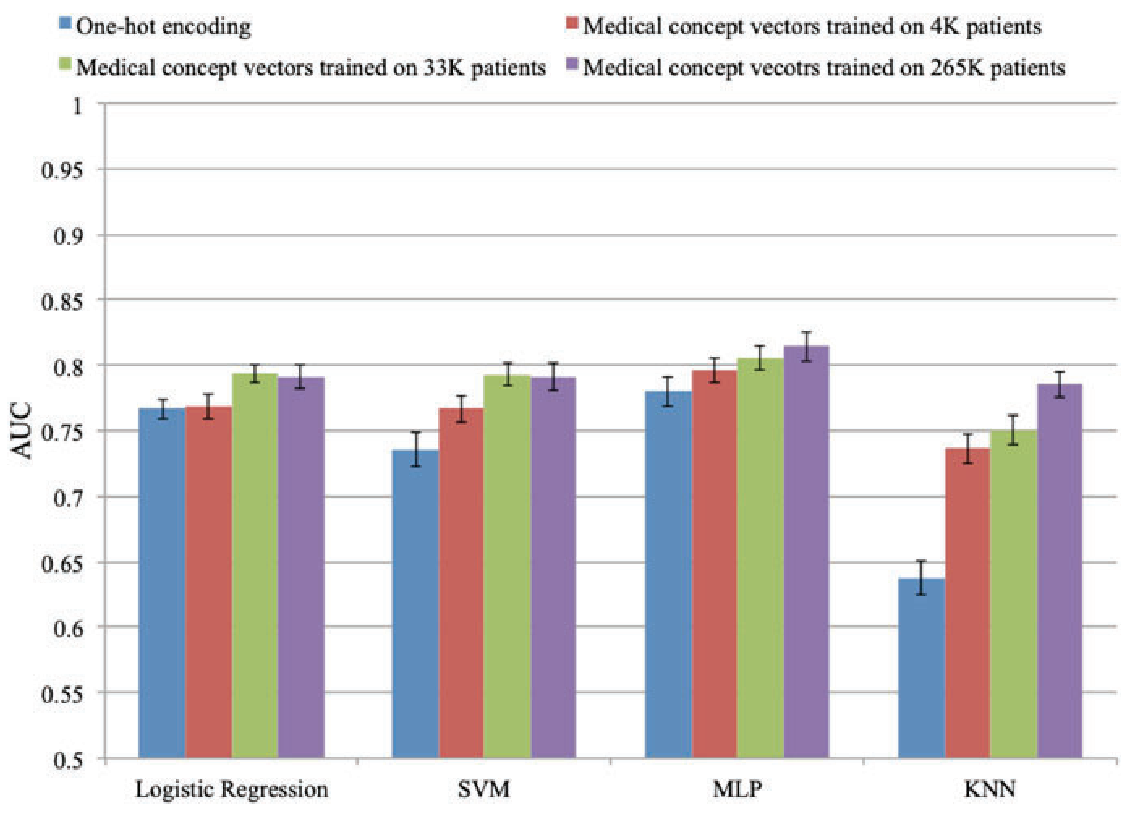


Fig. 5.6 Heart failure prediction performance of various embedding vectors

- **One-hot encoding** is to represent medical codes using one-hot encoding and then to sum over the one-hot encodings of the patient's medical codes.
- **Medical concept vectors** are simply the summation of all Word2Vec embeddings of her medical codes, where Word2Vec vectors are trained on various patient cohorts.

The error bars indicate the standard deviation derived from the six-fold cross evaluation. The power of medical concept representation learning is evident as all models show significant improvement in the HF prediction performance. Interestingly, KNN benefits the most from using the medical concept vectors, even those trained on the smallest dataset. Considering that KNN classification is based on the distances between data points, it is clear that medical concept representation using Word2Vec provides much better representations than the simple one-hot encoding. Figure 5.6 also tells us that medical concept representation is best learned with a large dataset. However, in most models, straightforward algorithms like KNN, even the medical concept vectors trained with the smallest number of patients improve prediction performance.

5.3 Med2Vec: Two-Level Embedding for EHR

Word2Vec assumes a global ordering of the input medical codes to create the embedding. However, many medical codes are documented together within a visit without any specific order, making it hard to use Word2Vec directly. In fact, EHR data of a patient follow a two-level hierarchy:

1. **Visit level:** Patient EHR data consists of a sequence of visits over time.
2. **Code level:** Each visit includes multiple medical codes, e.g., diagnosis, procedure, and medication codes.

This hierarchical structure provides two types of relational information, namely the sequential order of visits and co-occurrence of the codes within a visit. `Med2Vec` is a hierarchical embedding method that learns the representations for both medical codes and visits [24]. More formally, the set of all medical codes are denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ with size $|\mathcal{C}|$. EHR data for each patient is in the form of a sequence of visits $V_1, \ldots, V_T$ where each visit contains a subset of medical codes $V_t \subseteq \mathcal{C}$. The goal of Med2Vec is to learn two types of representations:

- **Code representations** map every code in the set of all medical codes $\mathcal{C}$ to non-negative real-valued vectors.
- **Visit representations** learn another embedding that maps every visit (a set of medical codes) to a real-valued vector of dimension n.

5.3.1 Med2Vec Method

Med2Vec first learns embedding vectors for individual medical codes such as diagnosis and procedure codes. Using the code embedding, we can map the visit binary vectors into intermediate embedding vectors, where each dimension corresponds to a binary variable for a specific medical code. Then through another nonlinear transformation, the final visit embedding is computed. The figurerefembed-fig:Med2Vec depicts the architecture of the Med2Vec model (Fig. 5.7).

Learning Code Representation A natural choice to capture the code co-occurrence information is to modify Word2Vec. In particular, we can train $\boldsymbol{W}_c \in \mathbb{R}^{m\times|\mathcal{C}|}$ (which can be used to produce intermediate visit level representations) so that the i-th column of $\boldsymbol{W}_c$ will be the representation for the i-th medical code among the total $|\mathcal{C}|$ codes. Note that given the unordered nature of the codes inside a visit, unlike the original Word2Vec, there is no ordering among medical codes within a visit.

From a set of visits $V_1, V_2, \ldots, V_T$, the code-level representations can be learned by maximizing the following log-likelihood,

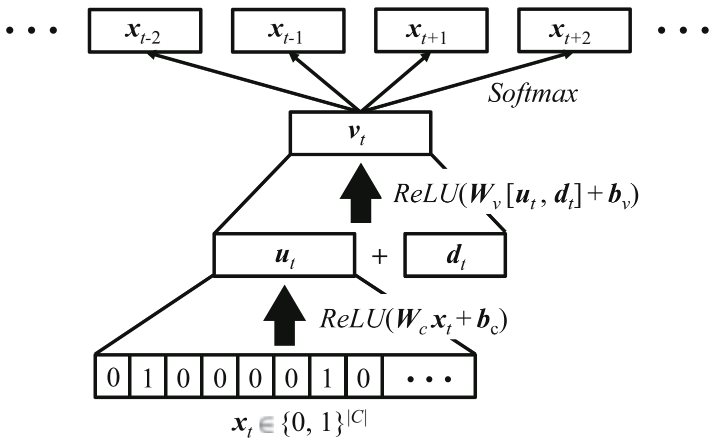


Fig. 5.7 Med2Vec model: a visit comprised of several medical codes is converted to a binary vector (i.e., the summation of one-hot embeddings of all medical codes within the visit). The binary vector is then converted to an intermediate visit representation, which is concatenated with a vector of demographic information to learn the final visit representation $\boldsymbol{v}_t$. Similar to Word2Vec, each visit from Med2Vec is trained to predict medical codes in the neighboring visits

$$\max_{\boldsymbol{W}_c} \quad \frac{1}{T}\sum_{t=1}^{T}\sum_{c_i\in V_t}\sum_{c_j\in V_t, j\neq i} \log p(c_j|c_i),$$

$$\text{where} \quad p(c_j|c_i) = \frac{\exp\left(\boldsymbol{W}_c[:, j]^\top \boldsymbol{W}_c[:, i]\right)}{\sum_{k=1}^{|\mathcal{C}|}\exp\left(\boldsymbol{W}_c[:, k]^\top \boldsymbol{W}_c[:, i]\right)}.$$

Learning Visit Representation Given a visit V_t, a multi-layer perceptron (MLP) is used to generate the corresponding visit representation $\boldsymbol{v}_t$. First, visit V_t is represented by a binary vector $\boldsymbol{x}_t \in \{0, 1\}^{|\mathcal{C}|}$ where the i-th entry is 1 only if code $c_i \in V_t$. Then $\boldsymbol{x}_t$ is converted to an intermediate visit representation $\boldsymbol{u}_t \in \mathbb{R}^m$ as follows,

$$\boldsymbol{u}_t = ReLU(\boldsymbol{W}_c\boldsymbol{x}_t + \boldsymbol{b}_c)$$

using the code weight matrix $\boldsymbol{W}_c \in \mathbb{R}^{m\times|\mathcal{C}|}$ and the bias vector $\boldsymbol{b}_c \in \mathbb{R}^m$. The ReLU unit ensures the nonnegative output for intuitive interpretation as negative values are typically hard to interpret and the ReLU unit eliminates them. Demographic information $\boldsymbol{d}_t \in \mathbb{R}^d$ including age and gender is concatenated with $\boldsymbol{u}_t$, where d is the size of the demographic information vector. Then the visit embedding $\boldsymbol{v}_t \in \mathbb{R}^n$ is computed as follows,

$$\boldsymbol{v}_t = ReLU(\boldsymbol{W}_v[\boldsymbol{u}_t, \boldsymbol{d}_t] + \boldsymbol{b}_v)$$

using the visit weight matrix $\boldsymbol{W}_v \in \mathbb{R}^{n\times(m+d)}$ and the bias vector $\boldsymbol{b}_v \in \mathbb{R}^n$, where n is the predefined size of the visit representation.

Similar to Word2Vec, given a visit representation one should be able to predict what has happened in the past, and what will happen in the future. Specifically, given a visit representation $\boldsymbol{v}_t$, we train a softmax classifier that predicts the medical codes of the visits within a context window. The cross-entropy loss is as follows,

$$\min_{\boldsymbol{W}_s, \boldsymbol{b}_s} \frac{1}{T} \sum_{t=1}^{T} \sum_{-w \le i \le w, i \ne 0} -\boldsymbol{x}_{t+i}{}^{\top} \log \hat{\boldsymbol{y}}_t - (\mathbf{1} - \boldsymbol{x}_{t+i})^{\top} \log(\mathbf{1} - \hat{\boldsymbol{y}}_t),$$

where $\hat{\boldsymbol{y}}_t = \text{Softmax}(\boldsymbol{W}_s \boldsymbol{v}_t + \boldsymbol{b}_s)$ are the predicted labels, $\boldsymbol{W}_s \in \mathbb{R}^{|\mathcal{C}| \times n}$ and $\boldsymbol{b}_s \in \mathbb{R}^{|\mathcal{C}|}$ are the weight matrix and bias vector for the softmax classifier, w the predefined context window size.

Clinical Use Case of Med2Vec Similar to Word2Vec, Med2Vec is mainly used as an alternative embedding method for longitudinal EHR data. In the papers [24], the authors have demonstrated the benefit of predictive performance and interpretability of the embedding vectors themselves. Here we present one result from [24] where the prediction performance of Med2Vec and other baselines are compared. The first row of Fig. 5.8 shows the Recall@30 for predicting the future medical codes. First, in all of the experiments, Med2Vec achieves the highest performance, although it is constrained to be positive and interpretable. The second observation is that Med2Vec's performance is robust to the hyperparameters' choice in a wide range of values. Compared to Skip-gram's more volatile performance, we can see that including the visit information in training improves the performance and stabilizes it.

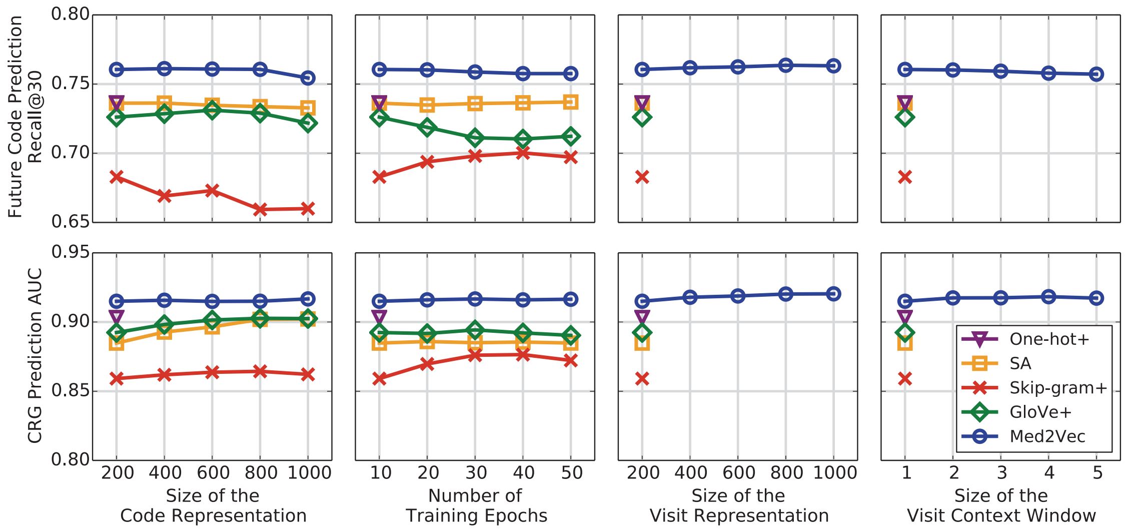


Fig. 5.8 Multiple baselines are compared, including one-hot encoding, stacked autoencoder (SA), Skip-gram, a version of Word2Vec, and GloVe, another efficient variant of embedding based on matrix factorization. The top row and the bottom row respectively show the Recall@30 for predicting the future medical codes and the AUC for predicting the CRG class when changing different hyperparameters

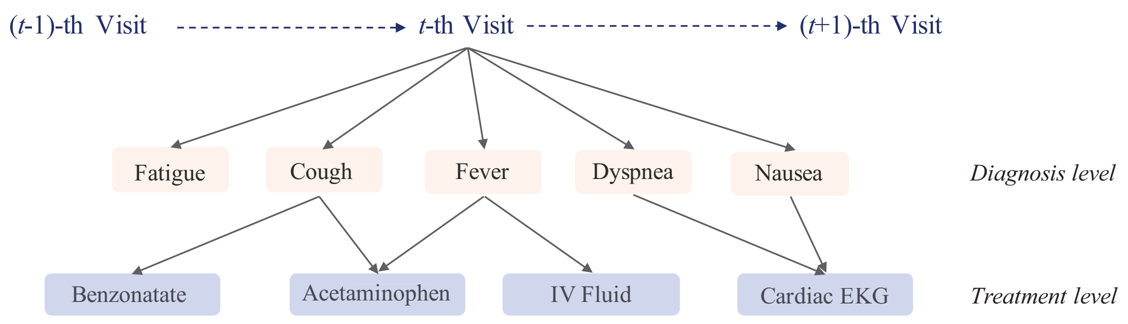


Fig. 5.9 Illustration of a single visit. Red denotes diagnosis codes, and blue denotes medication/procedure codes. A visit encompasses a set of codes and a hierarchical structure and heterogeneous relations among these codes. For example, while both *Acetaminophen* and *IV fluid* form an explicit relationship with *Fever*, they also are correlated with each other as descendants of *Fever*

5.4 MiME: Embed Internal Structure

Med2Vec utilizes the two-level structure of longitudinal EHR, namely, visits and codes within a visit. However, there is more nuance within a visit. The relationships between diagnoses and treatments, such as medication and procedures (see Fig. 5.9). MiME is another embedding method that leverages the inherent multilevel structure of EHR data and, in particular, the encoded relationships among medical codes [31]. This work also jointly performs auxiliary prediction tasks that rely on this inherent EHR structure without the need for external labels. Their overall objective is to learn robust embedding vectors even when the EHR dataset is small.

5.4.1 *Notations of MIME*

Assume a patient has a sequence of visits $\mathcal{V}^{(1)}, \ldots, \mathcal{V}^{(t)}$ over time, where each visit $\mathcal{V}^{(t)}$ contains a varying number of diagnosis objects $\mathcal{O}_1^{(t)}, \ldots, \mathcal{O}_{|\mathcal{V}^{(t)}|}^{(t)}$. Each $\mathcal{O}_i^{(t)}$ consists of a single diagnosis code $d_i^{(t)} \in \mathcal{A}$ and a set of associated treatments (medications or procedures) $\mathcal{M}_i^{(t)}$. Similarly, each $\mathcal{M}_i^{(t)}$ consists of varying number of treatment codes $m_{i,1}^{(t)}, \ldots, m_{i,|\mathcal{M}_i^{(t)}|}^{(t)} \in \mathcal{B}$. We omit the superscript (t) indicating the t-th visit to reduce clutter when we are discussing a single visit.

In Fig. 5.9, there are five diagnosis codes, hence five diagnosis objects $\mathcal{O}_1^{(t)}, \ldots, \mathcal{O}_5^{(t)}$. More specifically, the first diagnosis object $\mathcal{O}_1$ has $d_1^{(t)} = \textit{Fatigue}$ as the diagnosis code, but no treatment codes. $\mathcal{O}_2$, on the other hand, has diagnosis code $d_2^{(t)} = \textit{Cough}$ and two associated treatment codes $m_{2,1}^{(t)} = \textit{Benzonatate}$ and $m_{2,2}^{(t)} = \textit{Acetaminophen}$. In this case, we can use $g(d_2^{(t)}, m_{2,1}^{(t)})$ to capture the interaction between diagnosis code *Cough* and treatment code *Benzonatate*, which will be fed to $f(d_2^{(t)}, \mathcal{M}_2^{(t)})$ to obtain the vector representation of diagnosis object

Table 5.1 Notations for MIME. Note that the same dimensionality z is used in many places due to the use of skip-connections

Notation	Definition
$\mathcal{A}$	Set of unique diagnosis codes
$\mathcal{B}$	Set of unique treatment codes (medications and procedures)
$\mathbf{h}$	A vector representation of a patient
$\mathcal{V}^{(t)}$	A patient's t-th visit, which contains diagnosis objects $\mathcal{O}_1^{(t)}, \ldots, \mathcal{O}_{\lvert\mathcal{V}^{(t)}\rvert}^{(t)}$
$\mathbf{v}^{(t)} \in \mathbb{R}^z$	A vector representation of $\mathcal{V}^{(t)}$
$\mathcal{O}_i^{(t)}$	i-th diagnosis object of t-th visit consisting of diagnosis code $d_i^{(t)}$ and treatment codes $\mathcal{M}_i^{(t)}$
$\mathbf{o}_i^{(t)} \in \mathbb{R}^z$	A vector representation of $\mathcal{O}_i^{(t)}$
$p(d_i^{(t)} \mid \mathbf{o}_i^{(t)}),\ p(m_{i,j}^{(t)} \mid \mathbf{o}_i^{(t)})$	Auxiliary predictions, respectively for a diagnosis code and a treatment code based on $\mathbf{o}_i^{(t)}$
$d_i^{(t)} \in \mathcal{A}$	diagnosis code of diagnosis object $\mathcal{O}_i^{(t)}$
$\mathcal{M}_i^{(t)}$	a set of treatment codes associated with i-th diagnosis code $d_i^{(t)}$ in visit t
$m_{i,j}^{(t)} \in \mathcal{B}$	j-th treatment code of $\mathcal{M}_i^{(t)}$
$g(d_i^{(t)}, m_{i,j}^{(t)})$	A function that captures the interaction between $d_i^{(t)}$ and $m_{i,j}^{(t)}$
$f(d_i^{(t)}, \mathcal{M}_i^{(t)})$	A function that computes embedding of diagnosis object $\mathbf{o}_i^{(t)}$
$r(\cdot) \in \mathbb{R}^z$	A helper notation for extracting $d_i^{(t)}$ or $m_{i,j}^{(t)}$'s embedding vector

$\mathbf{o}_2^{(t)}$. Using the five diagnosis object embeddings $\mathbf{o}_1^{(t)}, \ldots, \mathbf{o}_5^{(t)}$, we can obtain a visit embedding $\mathbf{v}^{(t)}$. Also, some treatment codes (*e.g., Acetaminophen*) can be shared by two or more diagnosis codes (*e.g., Cough, Fever*) if the doctor ordered a single medication for more than one diagnosis. Then each diagnosis object will have its own copy of the treatment code attached to it, in this case, denoted, $m_{2,2}^{(t)}$ and $m_{3,1}^{(t)}$, respectively. Table 5.1 provides all the notations of MIME.

5.4.2 Description of MIME

Multilevel Embedding MIME explicitly captures the hierarchy between diagnosis codes and treatment codes depicted in Fig. 5.9. Figure 5.10 illustrates how MIME builds the representation of $\mathcal{V}$ (omitting the superscript (t)) in a bottom-up fashion via multilevel embedding. In a single diagnosis object $\mathcal{O}_i$, a diagnosis code d_i and its associated treatment codes $\mathcal{M}_i$ are used to obtain a vector representation of $\mathcal{O}_i$, $\mathbf{o}_i$. Then multiple diagnosis object embeddings $\mathbf{o}_0, \ldots, \mathbf{o}_{\lvert\mathcal{V}\rvert}$ in a single visit are used to obtain a visit embedding $\mathbf{v}$, which in turn forms a patient embedding $\mathbf{h}$ with other visit embeddings. The formulation of MIME is as follows:

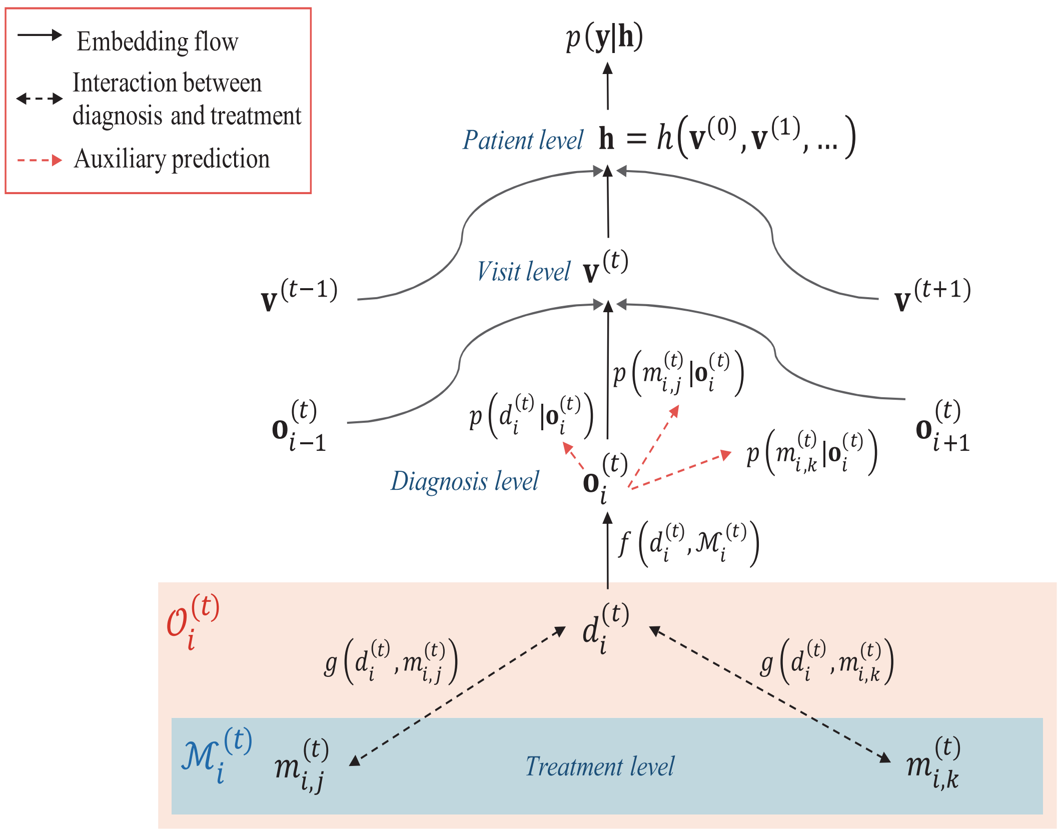

Embedding flow
Interaction between diagnosis and treatment
Auxiliary prediction
$p(\mathbf{y}|\mathbf{h})$
Patient level $\mathbf{h} = h(\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \ldots)$
Visit level $\mathbf{v}^{(t)}$
$\mathbf{v}^{(t-1)}$
$\mathbf{v}^{(t+1)}$
$\mathbf{o}_{i-1}^{(t)}$
$\mathbf{o}_{i+1}^{(t)}$
$p(d_i^{(t)}|\mathbf{o}_i^{(t)})$
$p(m_{i,j}^{(t)}|\mathbf{o}_i^{(t)})$
$p(m_{i,k}^{(t)}|\mathbf{o}_i^{(t)})$
Diagnosis level $\mathbf{o}_i^{(t)}$
$f(d_i^{(t)}, \mathcal{M}_i^{(t)})$
$\mathcal{O}_i^{(t)}$
$d_i^{(t)}$
$g(d_i^{(t)}, m_{i,j}^{(t)})$
$g(d_i^{(t)}, m_{i,k}^{(t)})$
$\mathcal{M}_i^{(t)}$
$m_{i,j}^{(t)}$
Treatment level
$m_{i,k}^{(t)}$

Fig. 5.10 Prediction model using MIME. Codes are embedded into multiple levels: diagnosis-level, visit-level, and patient-level. Final prediction $p(\mathbf{y}|\mathbf{h})$ is based on the patient representation $\mathbf{h}$, which is derived from visit representations $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \ldots$, where each $\mathbf{v}^{(t)}$ is generated using MIME framework

$$\mathbf{v} = \sigma\Bigg(\mathbf{W}_v\Big(\underbrace{\sum_i^{|\mathcal{V}|} f(d_i, \mathcal{M}_i)}_{\text{F: used for skip-connection}}\Big)\Bigg) + F \tag{5.1}$$

$$f(d_i, \mathcal{M}_i) = \mathbf{o}_i = \sigma\Bigg(\mathbf{W}_o\Big(\underbrace{r(d_i) + \sum_j^{|\mathcal{M}_i|} g(d_i, m_{i,j})}_{\text{G: used for skip-connection}}\Big)\Bigg) + G \tag{5.2}$$

$$g(d_i, m_{i,j}) = \sigma\big(\mathbf{W}_m r(d_i)\big) \odot r(m_{i,j}) \tag{5.3}$$

where Eqs. (5.1), (5.2), and (5.3) describe MIME in a top-down fashion, respectively corresponding to *Visit level*, *Diagnosis level* and *Treatment level* in Fig. 5.10 and $\odot$ denotes the element-wise multiplication. In Eq. (5.1), a visit embedding $\mathbf{v}$ is obtained by summing diagnosis object embeddings $\mathbf{o}_1, \ldots, \mathbf{o}_{|\mathcal{V}|}$, which are then transformed with $\mathbf{W}_v \in \mathbb{R}^{z \times z}$, and σ is a non-linear activation function such as sigmoid or rectified linear unit (ReLU). In Eq. (5.2), $\mathbf{o}_i$ is obtained by summing $r(d_i) \in \mathbb{R}^z$, the vector representation of the diagnosis code d_i, and the effect of the interactions between d_i and its associated treatments $\mathcal{M}_i$, which are then transformed with $\mathbf{W}_o \in \mathbb{R}^{z \times z}$. The interactions captured by $g(d_i, m_{i,j})$ are added to the $r(d_i)$, which can be interpreted as adjusting the diagnosis representation according to its associated treatments (medications and procedures). Note that in both Eqs. (5.1) and (5.2), the term F and G denote skip-connections [67]. In Eq. (5.3), the interaction between a diagnosis code embedding $r(d_i)$ and a treatment code embedding $r(m_{i,j})$ is captured by element-wise multiplication $\odot$. Weight matrix $\mathbf{W}_m \in \mathbb{R}^{z \times z}$ sends the diagnosis code embedding $r(d_i)$ into another latent space, where the interaction between d_i and the corresponding $m_{i,j}$ can be effectively captured. With Eq. (5.3) in mind, G in Eq. (5.2) can also be interpreted as $r(d_i)$ being skip-connected to the sum of interactions $g(d_i, m_{i,j})$.

Joint Training with Auxiliary Tasks Patient embedding $\mathbf{h}$ is often used for specific prediction tasks, such as heart failure prediction or mortality. The representation power of $\mathbf{h}$ comes from properly capturing each visit $\mathcal{V}^{(t)}$, and modeling the longitudinal aspect with the function $h(\mathbf{v}_0, \ldots, \mathbf{v}_t)$. Since the focus of this work is on modeling a single visit $\mathcal{V}^{(t)}$, we perform auxiliary predictions as follows:

$$\hat{d}_i^{(t)} = p(d_i^{(t)} | \mathbf{o}_i^{(t)}) = \mathrm{softmax}(\mathbf{U}_d \mathbf{o}_i^{(t)}) \tag{5.4}$$

$$\hat{m}_{i,j}^{(t)} = p(m_{i,j}^{(t)} | \mathbf{o}_i^{(t)}) = \sigma(\mathbf{U}_m \mathbf{o}_i^{(t)}) \tag{5.5}$$

$$L_{aux} = -\lambda_{aux} \sum_{t}^{T} \Bigg(\sum_{i}^{|\mathcal{V}^{(t)}|} \Big(CE(d_i^{(t)}, \hat{d}_i^{(t)}) + \sum_{j}^{|\mathcal{M}_i^{(t)}|} CE(m_{i,j}^{(t)}, \hat{m}_{i,j}^{(t)}) \Big) \Bigg) \tag{5.6}$$

Given diagnosis object embeddings $\mathbf{o}_1^{(t)}, \ldots, \mathbf{o}_{|\mathcal{V}^{(t)}|}^{(t)}$, while aggregating them to obtain $\mathbf{v}^{(t)}$ as in Eq. (5.1), MIME predicts the diagnosis code $d_i^{(t)}$, and the associated treatment code $m_{i,j}^{(t)}$ as depicted by Fig. 5.10. In Eqs. (5.4) and (5.5), $\mathbf{U}_d \in \mathbb{R}^{|\mathcal{A}| \times z}$ and $\mathbf{U}_m \in \mathbb{R}^{|\mathcal{B}| \times z}$ are weight matrices used to compute the prediction of diagnosis code $\hat{d}_i^{(t)}$ and the prediction of the treatment code $\hat{m}_{i,j}^{(t)}$, respectively. In Eq. (5.6), T denotes the total number of visits the patient made, $CE(\cdot, \cdot)$ the cross-entropy function and λ_{aux} the coefficient for the auxiliary loss term. We used the softmax function for predicting $d_i^{(t)}$ since in a single diagnosis object $\mathcal{O}_i^{(t)}$, there is only one diagnosis code involved. However, there could be multiple treatment codes associated with $\mathcal{O}_i^{(t)}$, and therefore we used $|\mathcal{B}|$ number of sigmoid functions for predicting each treatment code.

5.4.3 Experiment Results of MIME

This section highlights some experiment evaluation of MIME. More detailed evaluation can be seen at [31].

Baseline Models First, we use Gated Recurrent Units (GRU) described in Chap. 7 with different embedding strategies to map visit embedding sequence $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(T)}$ to a patient representation $\mathbf{h}$:

- **raw**: A single visit $\mathcal{V}^{(t)}$ is represented by a binary vector $\mathbf{x}^{(t)} \in \{0, 1\}^{|\mathcal{A}|+|\mathcal{B}|}$. Only the dimensions corresponding to the codes occurring in that visit is set to 1, and the rest are 0.
- **linear**: The binary vector $\mathbf{x}^{(t)}$ is linearly transformed to a lower-dimensional vector $\mathbf{v}^{(t)} = \mathbf{W}_c\mathbf{x}^{(t)}$ where $\mathbf{W}_c \in \mathbb{R}^{b\times(|\mathcal{A}|+|\mathcal{B}|)}$ is the code embedding matrix. This is equivalent to taking the vector representations of the codes (i.e. columns of the embedding matrix $\mathbf{W}_c$) in the visit $\mathcal{V}^{(t)}$, and summing them up to derive a single vector $\mathbf{v}^{(t)} \in \mathbb{R}^b$.
- **sigmoid, tanh, relu**: The binary vector $\mathbf{x}^{(t)}$ is transformed to a lower-dimensional vector $\mathbf{v}^{(t)} = \sigma(\mathbf{W}_c\mathbf{x}^{(t)})$ where we use either *sigmoid*, *tanh*, or *ReLU* for $\sigma(\cdot)$ to add non-linearity to **linear**.
- **sigmoid**$_{mlp}$, **tanh**$_{mlp}$, **relu**$_{mlp}$: We add one more layer to **sigmoid**, **tanh** and **relu** to increase their expressivity. The visit embedding is now $\mathbf{v}^{(t)} = \sigma(\mathbf{W}_{c_2}\sigma(\mathbf{W}_{c_1}\mathbf{x}^{(t)}))$ where σ is either sigmoid, tanh or ReLU. We do not test **linear**$_{mlp}$ since two consecutive linear layers can be collapsed to a single linear layer.

Second, two EHR embedding methods Med2Vec [24] and GRAM [28] are also compared.

Heart Failure Prediction The objective is to predict the first diagnosis of heart failure (HF), given the 18-months observation records discussed. Among 30,764 patients, 3,414 were case patients diagnosed with HF within a 1-year window after the 18-months observation. The remaining 27,350 patients were controls. The logistic regression is scored against the patient representation $\mathbf{h}$ to obtain a probability score between 0 (no HF onset) and 1 (HF onset).

To evaluate MIME's performance from another perspective, we created four datasets E_1, E_2, E_3, E_4 from the original data. Each dataset consisted of patients with varying maximum sequence length T_{max} (i.e., the maximum number of visits). In order to simulate a new hospital collecting patient records over time, we increased T_{max} for each dataset such that 10, 20, 30, 150 for E_1, E_2, E_3, E_4 respectively. Each dataset had 6299 (414 cases), 15794 (1177 cases), 21128 (1848 cases), 27428 (3173 cases) patients respectively. For MIME$_{aux}$, we used the same 0.015 for the auxiliary loss coefficient λ_{aux}.

Figure 5.11 shows the test PR-AUC for HF prediction across all datasets. We can readily see that MIME outperforms all baseline models across all datasets. However, the performance gap between MIME and the baselines are larger in datasets E_1, E_2 than in datasets E_3, E_4, confirming our assumption that exploiting the inherent structure of EHR can alleviate the data insufficiency problem. Especially for the smallest dataset E_1, MIME$_{aux}$ (0.2831 PR-AUC) demonstrated significantly better performance than the best baseline **tanh**$_{mlp}$ (0.2462 PR-AUC), showing 15% relative improvement.

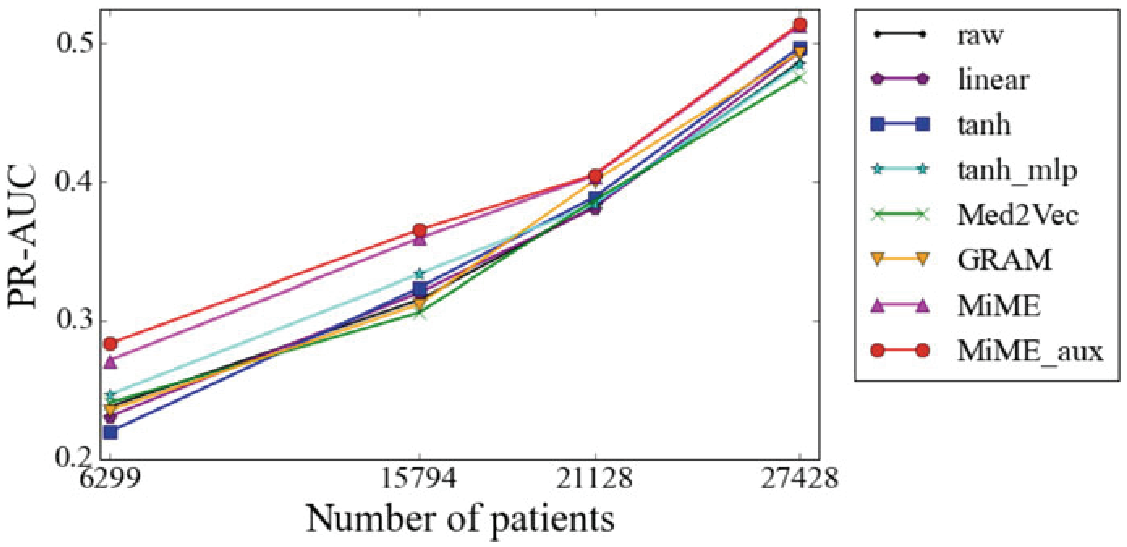


Fig. 5.11 Test PR-AUC of HF prediction for increasing data size

5.5 Exercises

1. What is the key difference between Med2Vec and Word2Vec?
2. What is the key difference between Med2Vec and MiME?
3. What is NOT true about one-hot encoding?
   (a) One-hot encoding vector is a multi-dimensional vectors of all 0s except for one dimension of 1.
   (b) One-hot encoding will group vectors of similar concepts together.
   (c) Distance between any two one-hot encoding vectors is always the same.
   (d) One-hot encoding is commonly used to encode categorical data.
4. What is NOT true about Word2vec?
   (a) One-hot encoding vector is a multi-dimensional vectors of all 0s except for one dimension of 1.
   (b) Larger context window means words that are farther apart will be considered similar in the skip gram calculation.
   (c) The denominator $\sum_{\text{all words}(w)} \exp(v_w \top v_t)$ in $p(c|t)$ can be easily computed for Skip-gram.
   (d) Negative sampling treats the pairs of context words and the target word as positive examples, and pairs of random words and the input target word as negative examples.
5. Which of the following is NOT a suitable medical applications of Word2vec?
   (a) Similarity search: To find similar medical concepts based on the Euclidian distance of the word2vec embeddings of those medical concepts.
   (b) Algebra operation: To perform summation and subtraction of word2vec embeddings in order to better understand a combination of medical concepts.

   (c) Features for predictive modeling: To utilize word2vec embeddings as input feature vectors to support downstream classification or regression models.
   (d) Visualization: to directly visualize the word2vec embeddings of medical concepts in order to understand the relationship among medical concepts.

6. What are the challenges in visualizing high-dimensional data? (multiple correct answers)

   (a) To preserve the relationship of original data when visualizing in 2D space.
   (b) To ensure similar data points mapped to nearby location in 2D space.
   (c) To provide computationally efficient methods to produce the visualization.
   (d) To produce aesthetically pleasing effects when visualizing the data.

7. Which statement is NOT true comparing PCA and t-SNE?

   (a) PCA is a dimensionality reduction algorithm that tries to preserve global distance between all pairs of data points.
   (b) t-SNE tries to project high-dimensional data into 2D space while preserving local distance.
   (c) t-SNE is more appropriate for visualization high-dimensional data.
   (d) t-SNE is computationally more efficient than PCA.

8. Which is NOT true about Med2Vec method?

   (a) It models two level hierarchy of medical data namely visit level and patient level.
   (b) Med2Vec is most suitable to model clinical notes
   (c) Med2Vec is designed to model sequences of clinical codes.
   (d) Mec2Vec is a generalization of word2vec for electronic health record data.

9. Which is NOT true about MiME: Multilevel medical embedding method?

   (a) It leverages the dependency from diagnosis to treatments.
   (b) It models EHR data with multi-level hierarchy (treatment -> diagnosis -> visit—patient).
   (c) It introduces auxiliary prediction task to predict diagnosis and treatment within a visit.
   (d) It provides a direct and intuitive interpretation for each prediction.