02-750 Week 5
Automation of Scientific Research

Aidan Jan

February 12, 2026

Compound Screening

Goal: find compounds that bind to and modify a given target.
Screening can be done **in vitro** or **in silico** (i.e., computationally)

- *in vitro* gives definite answers, but is expensive
- *in silico* screen is faster, but it is always necessary to validate hits *in vitro*

Challenge: Most compounds won't bind to the target (i.e., they are *inactive*)

- Thus, most of the *in vitro* screening effort is wasted. In theory, *in silico* screening could increase the yield from *in vitro* screens
- But *in silico* screening requires models, which must be trained from **highly imbalanced data sets**.

Linear Classifiers

- Let $x \in \mathbb{R}^n$ be a point in an n-dimensional space
- Let $w \in \mathbb{R}^n$ be a weight vector
- Let $b \in \mathbb{R}$ be a bias term
- The set of points satisfying $w \cdot x + b = 0$ form a **hyperplane**
- A hyperplane can be used as a binary classifier for instance x by simply determining whether $w \cdot x + b < 0$ or $w \cdot x + b \geq 0$

There are many different algorithms for learning linear classifiers

- Ex. Naive Bayes; LDA; Logistic Regression; SVMs

Support Vector Machines (SVMs)

- SVM learning algorithms find the hyperplane that maximizes the **margin** between the two classes (aka structural risk minimization)
  - The margin is distance between hyperplane and the nearest training points
  - The points achieving this distance are called the support vectors

For SVMs,

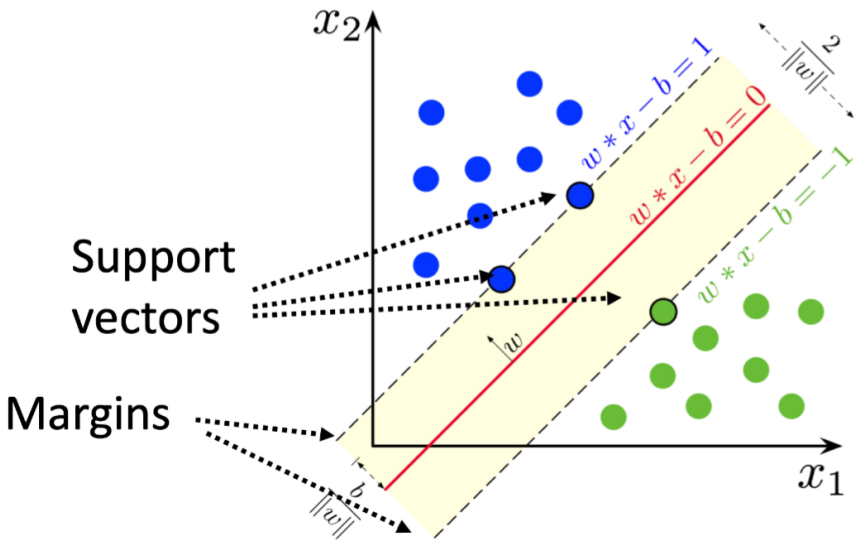
- Hypothesis class $\mathcal{H}$: linear classifiers
- Loss function: **Hinge Loss**

$$\updownarrow(x_i; w, b) = \max(0, 1 - y \cdot (w \cdot x_i - b))$$

- Risk objective: structural risk minimization

$$argmin_{w,b} = \frac{1}{n}\sum \updownarrow(x_i; w, b) + \lambda\|w\|^2$$

- Search function: quadratic programming



- Given the hyperplane, the **signed distance** between point x to the plane is:

$$\frac{w \cdot x + b}{\|w\|_2}$$

- The **unsigned distance** is:

$$\left|\frac{w \cdot x + b}{\|w\|_2}\right|$$

Experimental Set-up

- Source data: DuPont Thrombin data from KDD 2001
  - Binary features: $x \in \{0, 1\}^d$, where $d = 139,351$
- Initialization: random batches until the 1st active compound is found
- Batch query selection
  - Batch size: 5% of the unlabeled samples
  - 4 query selection strategies were used (see next slide)
- Two rounds of experiments were performed (Rounds 0 and 1)
  - Pool size: *Round 0*: 1316 (40 active); *Round 1*: 643 (150 active)
- Experiments are repeated 10 times to compute average performance

Query Selection Strategies

- Random Sampling
- Proximity-Based
  - i.e., points closest to known active compounds
- Select points with the largest positive distance from hyperplane
  - Assuming positive distance means active
- Near boundary selection (i.e., margin sampling).

Results:

- Active learning performed better than passive learning.
- Passive learning performed better than random guessing (no training).

Summary of Warmuth *et al.*

"Active Learning with Support Vector Machines in the Drug Discovery Process"

- Drug screening is a good candidate for Active Learning
  - Statistical models can predict activity quickly, if not perfectly
  - Activity assays are expensive (relative to computational predictions)
- However, drug screening does pose challenges to machine learning
  - The number of inactive compounds is usually *much* larger than those that are active, so a majority label classifier has high accuracy, but it's the minority label that we care about
  - Chemical space is large
  - Some encodings of compounds are high dimensional
- The paper demonstrates that SVM-based Active Learning can potentially overcome these challenges
- The querying strategy should be matched to your goals
  - If your goal is to maximize the number of active compounds discovered, use the **Largest Positive Score** criterion
  - If your goal is to maximize the classification accuracy of the model, use the **Near Boundary** criterion (aka margin sampling)

High Throughput Screening

- For a single target, engineer assay to detect effects on target
  - Protein, cellular, whole organism
- Compound library contains millions of compounds
- Automation
  - Liquid handling robotics
  - Plate reading mechanisms

Enormous experimental spaces are common

For drug discovery, we often have to test millions of drugs, and even if we only target one protein, we have to test on many others as well for side effects.

- We can't do all of these experiments.
- However, we can predict what we can't observe.

Perhaps we're not running the right experiments!

- We can choose to execute experiments expected to yield hits
- Execute experiments expected to yield the most improvement in accuracy of predicted efforts

Goal: Use active learning to improve the hit discovery rate for a large number of targets tested using diverse methods against a large number of compounds

- On PubChem database, we have:
  - 177 Assays (108 *in vitro*, 69 *in vivo*). Sign of score modified to reflect type of assay (inhibition or activation)
  - 133 Unique protein targets
  - 20000 Compounds
  - $\sim$1000000 experiments (30% coverage)
  - Compare discovery rate and accuracy across different methods. Discovery is defined as a drug-protein pair whose |rank score| > 80.

Out of the million experiments, around 0.096% returned active compounds, or protein pairs with |rank score| $>$ 80.

- Experiments were chosen by assuming activity can be predicted based on lienar combination of features.
- Use lasso regression (Tibshirani, 1996) to select features

Accuracy Assessment

- Train a model based on all observed data
- For all experiments:
  - |rank score| $> 80 \rightarrow$ positives
  - |rank score| $\leq 80 \rightarrow$ negatives
- Set a threshold t on predictions.
  - |predicted rank score| $> t \rightarrow$ positive prediction
  - |predicted rank score| $\leq t \rightarrow$ negative prediction
- Sweep across thresholds calculating True positive rate and False positive rate

Selection Methods

- **Uncertainty sampling:** select experiments for which prediction most uncertain.
  - Across n-folds, hold out a random portion of experimental results and make n predictions for each unobserved experiment
  - Select unobserved experiments with largest standard deviation in predictions
- **Density sampling:** select unobserved experiments from regions of feature space which have been least explored regardless of experimental values
  - Randomly select 2000 observed experiments
  - Randomly select 2000 unobserved experiments
  - Features represented for each experiment by concatenating compound and protein features
  - Select experiments with minimal values for:

$$\frac{\text{mean distance to Unobserved}}{\text{mean distance to Observed}}$$

- **Diversity sampling:** select most diverse set of experiments
  - Features represented for each experiment by concatenating compound and protein features
  - K-means ($K = 384$) on experiments
  - Select experiments nearest to each discovered centroid
- **Hybrid selection:** select half of experiments using greedy select and half using uncertainty sampling
- **Memory limited approach:** make selections using prediction learned from observations from the previous m rounds

Conclusions of Kangas *et al.*

"Efficient Discovery of Responses of Proteins to Compounds using Active Learning"

- Hits can be more rapidly identified using active learning while combining information from experiments for multiple targets
- Hits can be rapidly identified using a system that utilizes information from heterogeneous sources

Dropout as a Bayesian Approximation

- New theoretical framework casting dropout training in deep neural networks as approximate Bayesian inference in deep Gaussian processes
  - Develop the tools necessary to represent uncertainty in deep learning
- Perform an extensive assessment across learning tasks, model architectures, etc.
  - Show a considerably improvement in predictive loglikelihood and RMSE compared to existing state-of-the-art methods

Overview of Gaussian Process

Gaussian Process Regression (aka. Kriging)

- Informally, a Gaussian Process (GP) extends the concept of a **multivariate Gaussian probability distribution** over d-dimensional *vectors* to a probability distribution over continuous *functions*
- Key idea: Any d-dimensional continuous function, $f : \mathbb{R}^d \to \mathbb{R}$, can be interpreted as the realization of an **infinite** number of random variables, one for each point in $\mathbb{R}^d$
- Gaussian Processes are often used as **priors** in the context of Bayesian, nonparametric, nonlinear regression
  - Bayesian means that learning and inference are performed by applying Bayes' Theorem, which forces us to specify a prior on our unknowns.
- Recall that a Gaussian distribution is completely specified by two parameters: μ and Σ (the mean and covariance): $X \sim \mathcal{N}(\mu, \Sigma)$
- Similarly, a GP is completely specified by two parameters, m and K

$$f \sim \mathcal{GP}(m, K)$$

  - $m(x) = \mathbb{E}[f(x)]$ is the **mean function**
  - $K(x, x')$ is the **covariance function**, which can be used to compute the uncertainty (the gray area)
  - K is a **kernel**, that is, it is a measure of how similar points x and x' are.
- GP's assume that if x and x' are 'close', then $|f(x) - f(x')|$ is small
- When making predictions about an unobserved x', we condition the GP on the data, $D_L = \{(x_1, y_1), \cdots, (x_n, y_n)\}$
  - The prediction, $\hat{f}(x')$, is a distance-weighted average of $y_1, \cdots, y_n$ where the weighting is determined by the kernel function

What's the Problem?

A model can be uncertain in its predictions even with a high softmax output

- Passing a point estimate of a function (a) through a softmax (b) results in extrapolations with **unjustified** high confiednce for points far from the training data
- However, passing the distribution through a softmax better reflects classification uncertainty far from the training data

Summary: Gal and Ghahramani (ICML, 2016)

"Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning"

- Give a complete theoretical treatment of the link between Gaussian processes and dropout
  - Provide tools to model uncertainty with dropout NNs

Deep Batch Active Learning for Drug Discovery

Goal: Enable the use of active learning with advanced neural network models

$$\underset{\mathcal{B}\subset\mathcal{V},|\mathcal{B}|=N}{\arg\max}\ \log(\det(\mathrm{cov}(\mathcal{B})))$$

Here, $\mathcal{V}$ is the unlabeled dataset at a specific iteration and N is the batch size

- Randomly generated a collection of batches as starting points, each containing N distinct samples independently chosen from a distribution proportional to the quantile of the variances
- Select the best $M < N$ of these batches as starting points for optimization
- For each starting point, optimize the batch element-wise
  - i.e., changing the first element to optimize the covariance, then changing the second, and so on, doing several passes until the process reaches equilibrium
- Select the highest scoring final batch

Approximation of the Posterior Distribution

$$\hat{y} = \frac{1}{S}\sum_{t=1}^{S} f_\theta(x, m_s)$$

where S is the number of predictions by sampling m_S masks, $\hat{y}$ is the predicted output of the model, and f_θ is the model with parameters θ

Conclusions of Bailey *et al.*

"Deep Batch Active Learning for Drug Discovery"

- Developed two novel active learning batch selection methods
- Comprehensive evaluation across several tasks