Machine Learning Concepts in Chemoinformatics

Martin Vogt
B-IT Life Science Informatics
Rheinische Friedrich-Wilhelms-Universität Bonn

BigChem Winter School 2017
25. October

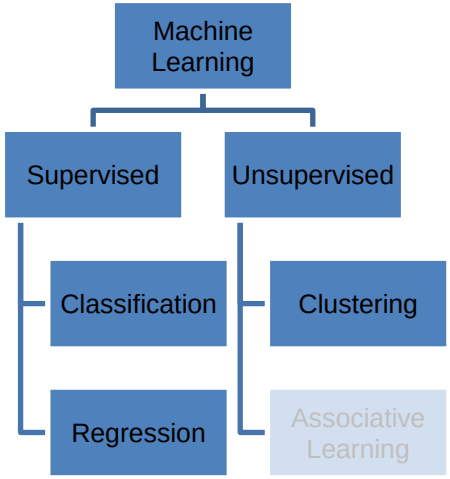
Data Mining in Chemoinformatics

- Goal: construct models that enable the identification of relationships between chemical structure and activity

- Traditional QSAR techniques (multiple linear regression) are not generally applicable
  - data sets (e.g. HTS sets) are usually too large
  - data sets are usually structurally diverse

- Machine learning techniques are required

Types of Machine Learning Algorithms



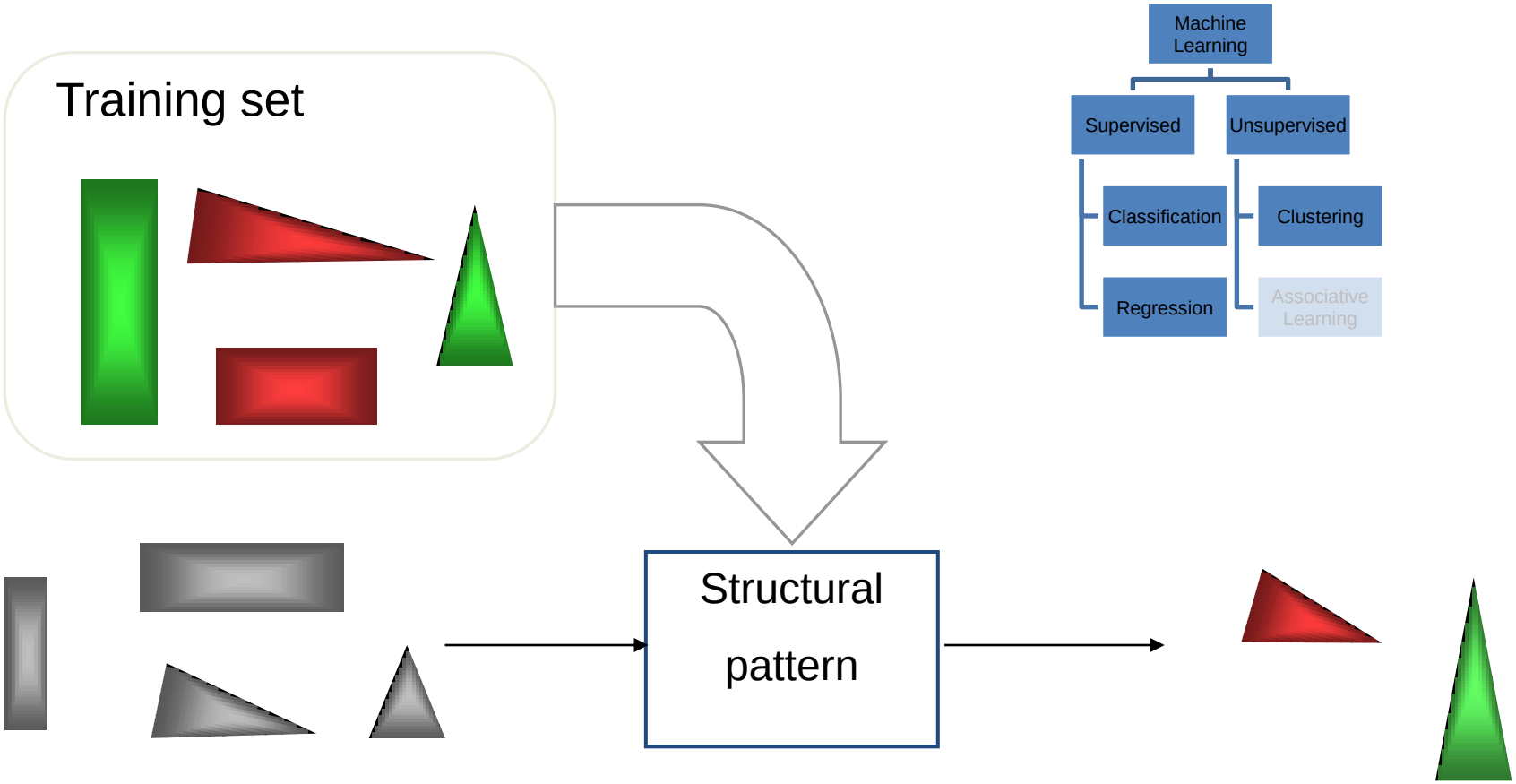
- annotated training sets given

  (input/output pairs: x / $f(x)$)
- deduce function f from training data
- produce the correct output $f(x)$ for an input x

- only unlabeled training examples are given
- determine how data are organized / find patterns in the data

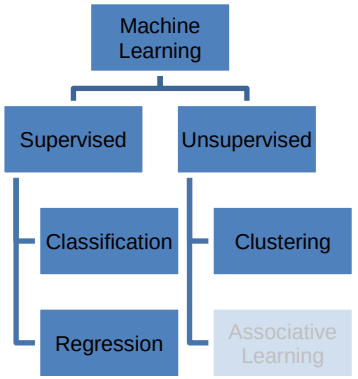
Classification

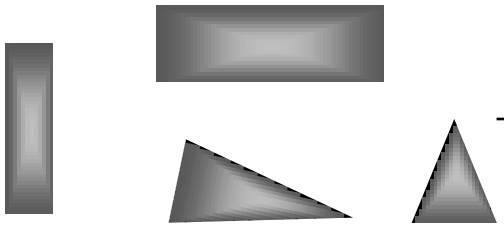
- Prediction of a class based on classified examples



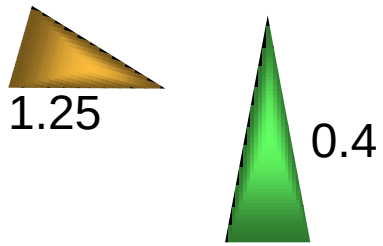
Regression

- Prediction of a numerical property based on examples with specific values



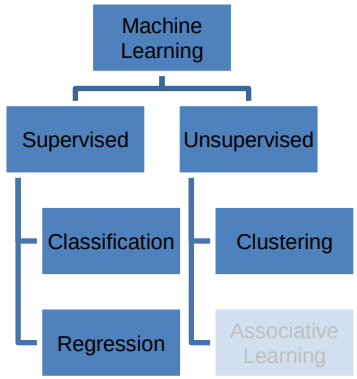


Structure-value relations



Clustering

- Organize data into groups of similar objects



Structural pattern

Machine learning steps

- Data
- Representation / Distance metric
- Objective function
- Machine learning method
- Performance evaluation
- Model selection: parameter optimization

Data and representation

- Select data for training
- Data representation
  - vector of features
    - features can be categorical or numerical
  - something else (computer-readable representation)
- Distance metric for representation
  - assess similarity between objects

Objective function

- Mathematical formulation of what to learn, e.g.
  - classification: minimize the number of misclassifications
  - regression: minimize the difference between correct and predicted quantity
  - clustering: minimize the distance within clusters while maximizing the distance between clusters

- ML method suited for minimizing the chosen objective function, e.g.
  - SVM for minimizing classification errors
  - linear regression for minimizing the „sum of squared errors“
  - hierarchical clustering…

Optimization method

- ML methods perform a tradeoff between
  - variance: sensitivity to training data
    - high variance -> overfitting
  - bias: error in (simplified) model assumptions
    - model performs as well on test as training data
    - high bias -> underfitting
- Regularization parameters
  - some methods have hyperparameters controlling the complexity of a model
  - the higher the complexity the better the performance on the training data
  - simpler models might not perform so well on training data, but might perform comparable on test data

Classification: Naive Bayes

- Models feature distributions for different classes
- Assumes that features are distributed differently
- Distributions are modeled based on training data
  - normal distributions

$$L(A \mid x_i) \propto p(x_i \mid A) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x-\mu_i)^2}{\sigma_i^2} \right)$$

  - Bernoulli distributions

$$L(A \mid v_i = 1) \propto P(v_i = 1 \mid A) = p_i$$

- Independence of features is assumed

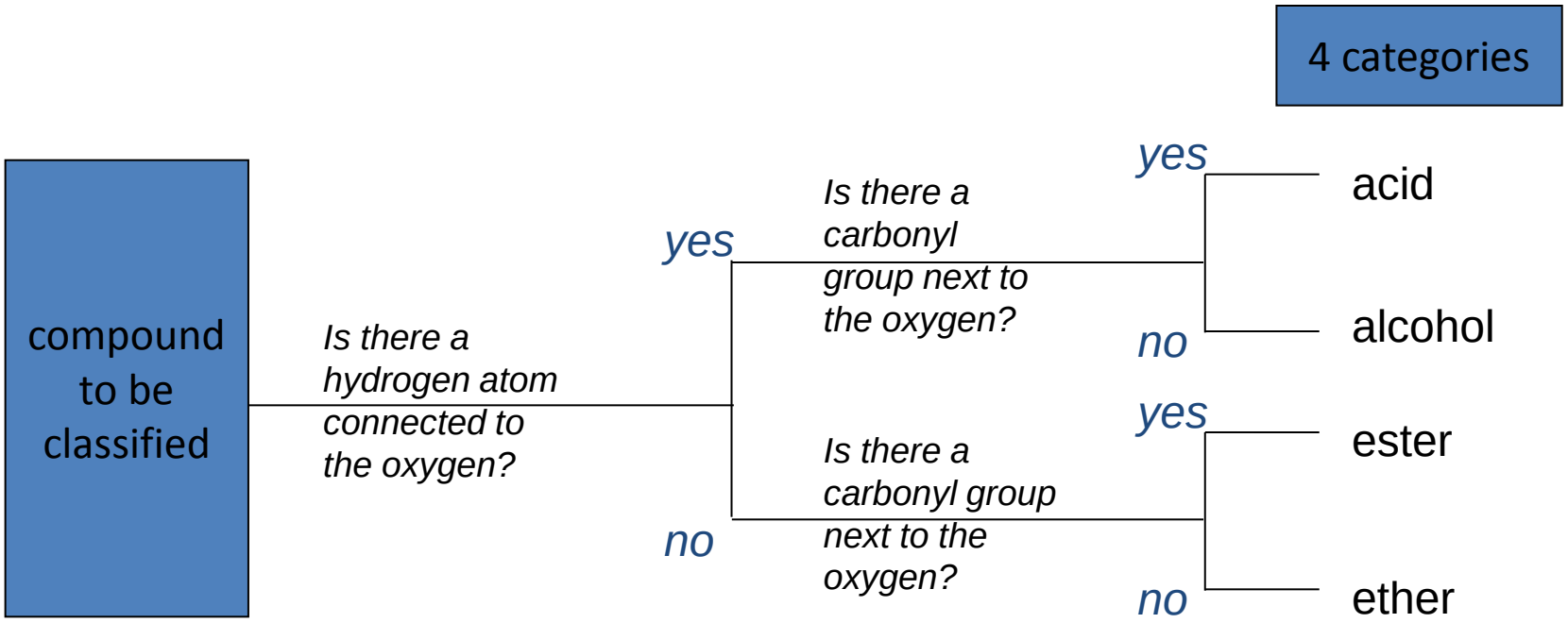
$$L(A \mid x) \propto p(x \mid A) = \prod_{i=1}^{n} p(x_i \mid A)$$

Classification: Naive Bayes

- Naive Bayesian classification are easy to use
- Naive Bayes makes strong assumptions
  - continuous features are normally distributed
  - feature distributions are conditionally independent
- These assumptions can introduce a strong bias into the model

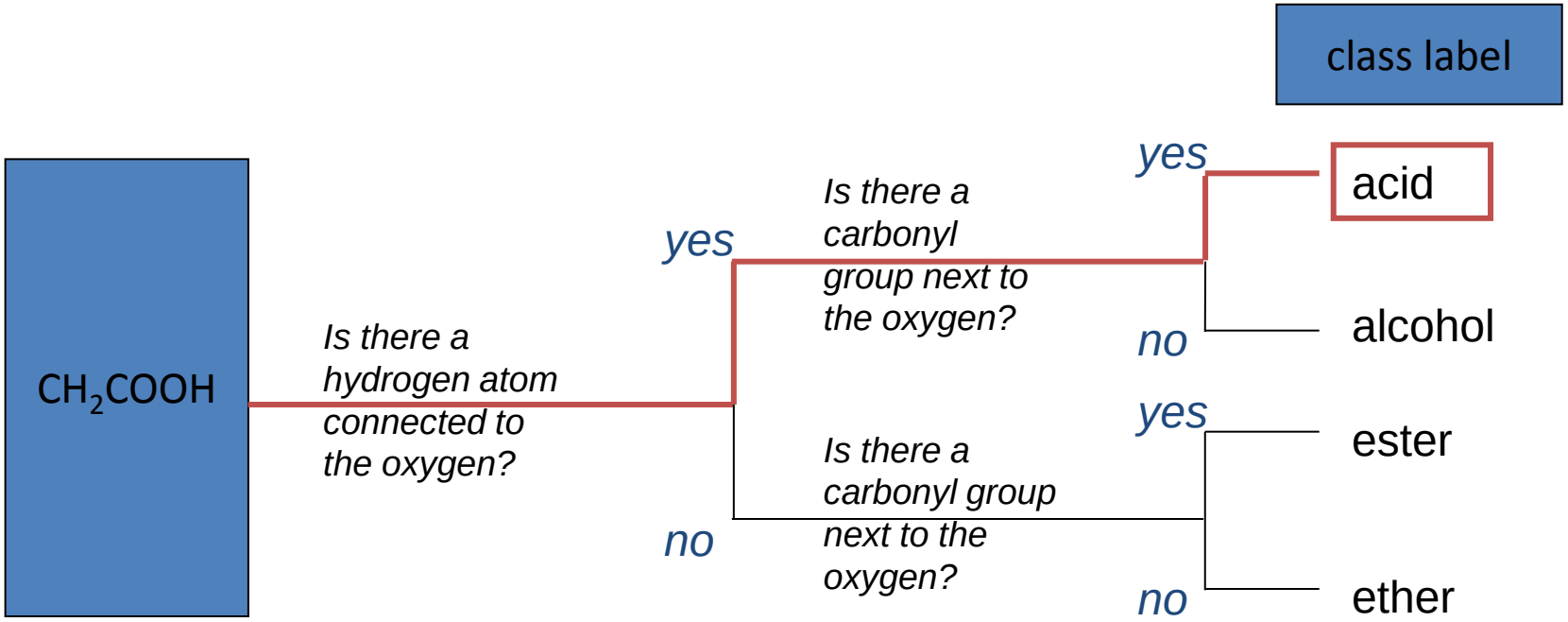
Classification: Decision Tree

- Simple example
  - classification of oxygen-containing compounds



Classification: Decision Tree

- Given a query object (a molecule, e.g.)
  - traverse the tree and test the attribute values of the object
  - assign the class label of the respective leaf to the object



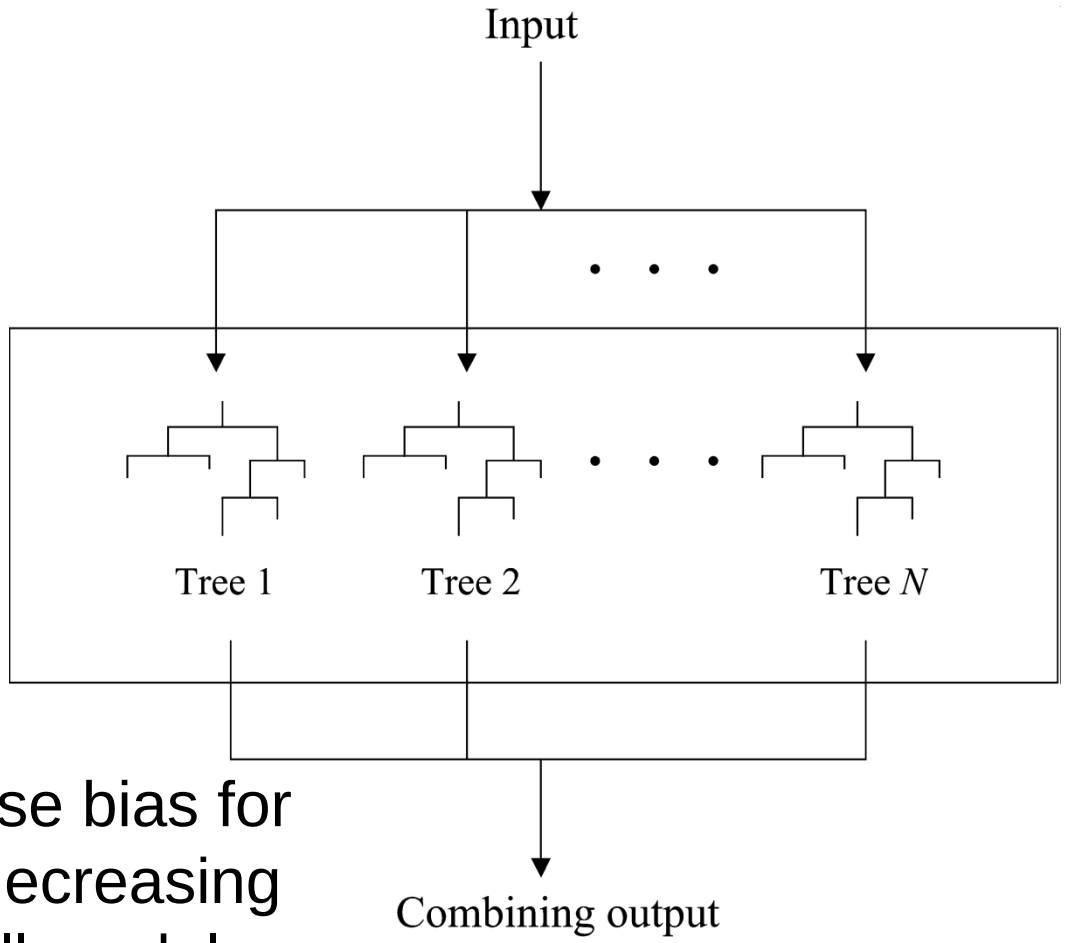
Classification: Decision Trees

- Decision trees are easy to use
  - different types os features: numerical categorical
  - no explicit metric required
  - „white box“: Relevant features are observable
  - prone to overfitting (high variance)

- Hyperparameters have to be set
  - depth of tree
  - number of features to consider

Classification: Random Forest

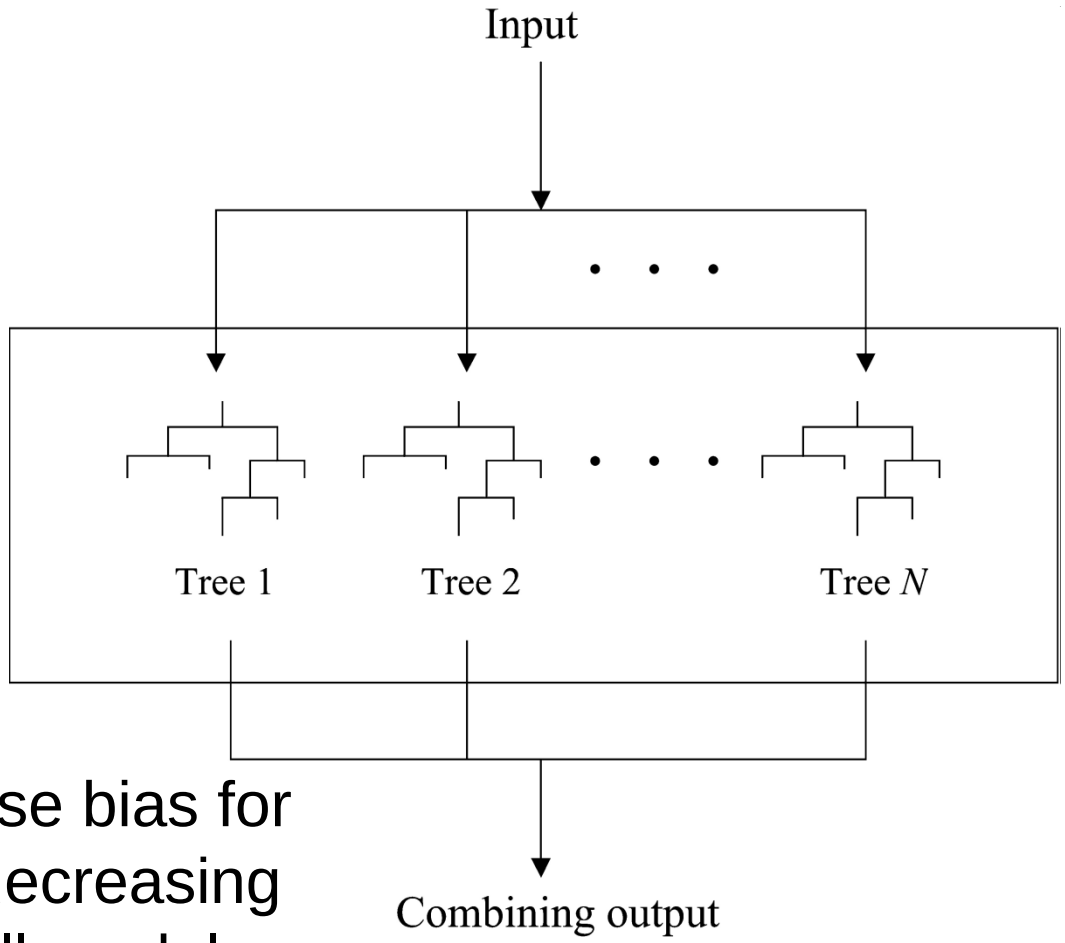
- A machine learning ensemble classifier
  - consisting of many decision trees
  - trees build from subsamples of training data
  - tree decisions based on subset of features



- Ensemble models increase bias for individual models while decreasing the variance of the overall model

Classification: Random Forest

- A machine learning ensemble classifier
  - combining output class labels of the individual trees to one final output class label
  - **consensus prediction** (class predicted by the majority of trees)



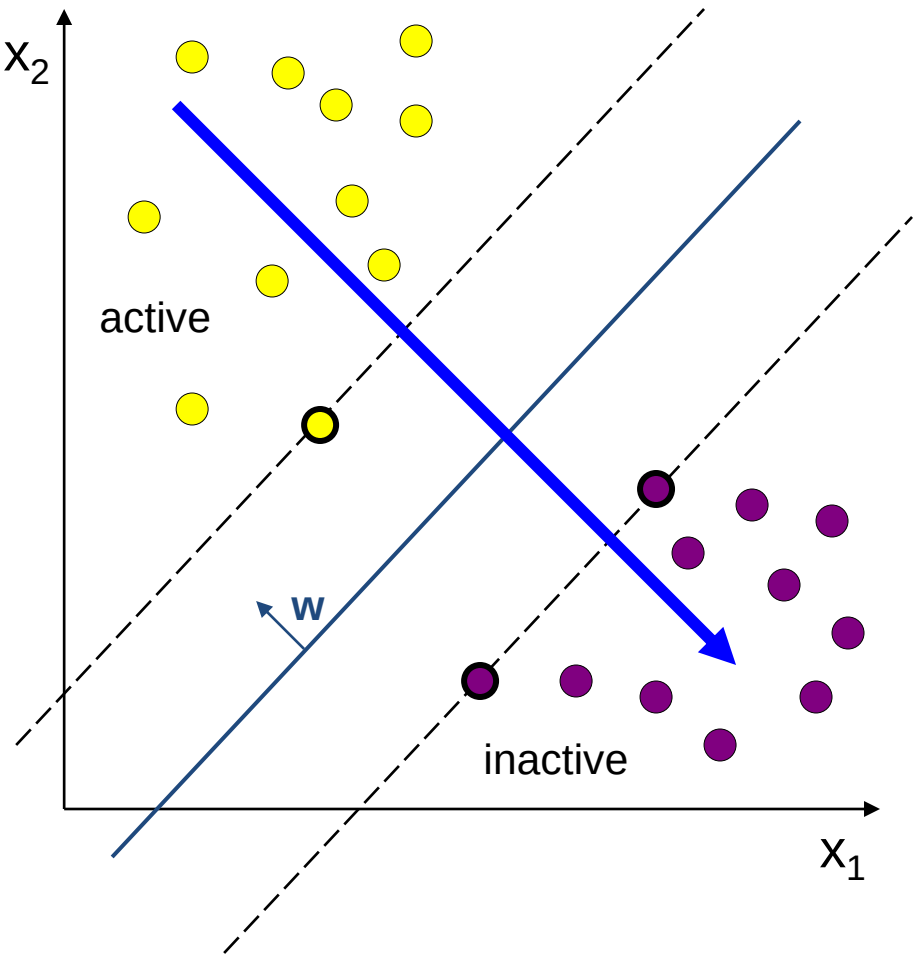
- Ensemble models increase bias for individual models while decreasing the variance of the overall model

Classification: Support Vector Machines (SVM)

- Supervised binary classification approach

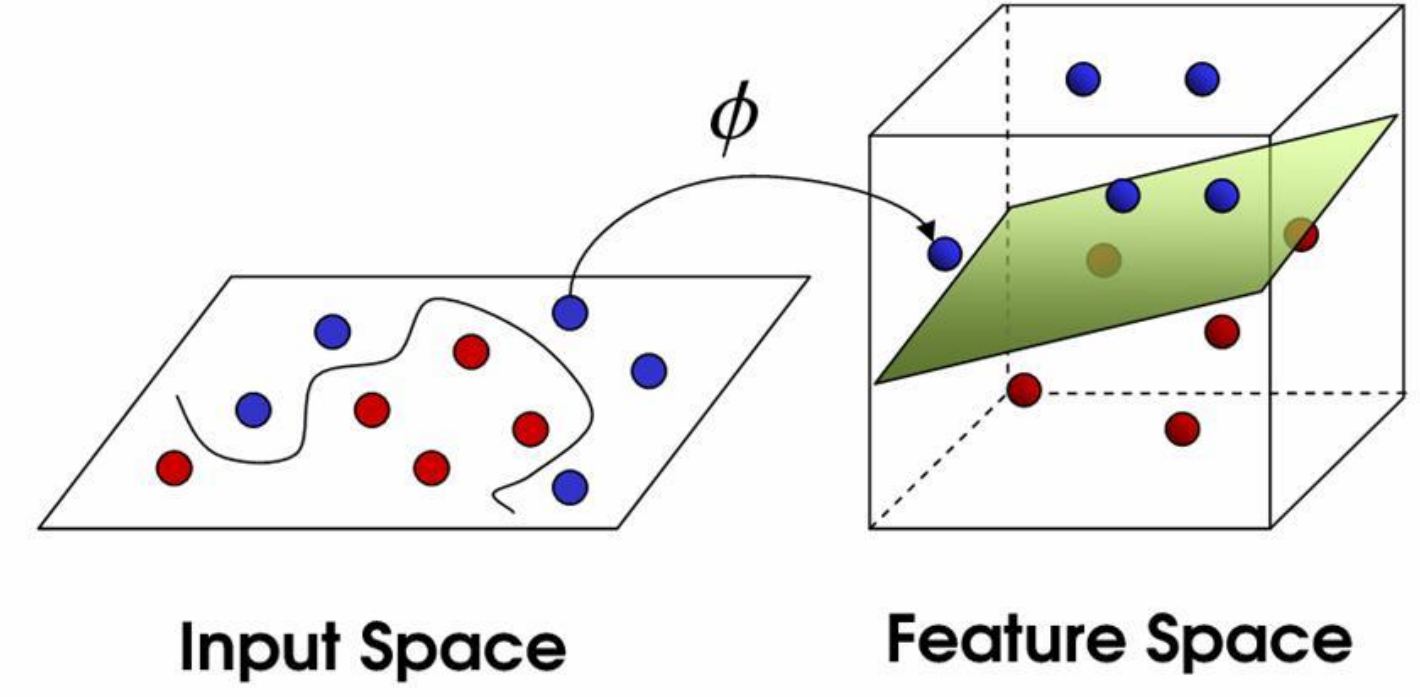
Idea:

- Derivation of a separating hyperplane
- Projection of test compounds for
  – classification
  – ranking
- Slack variables allow for misclassification of some data during modeling



Classification: SVM
Feature Space Transformation

- A reasonable linear separation of data is not always possible (even if limited classification errors are allowed)
- Projection of data into higher dimensional feature space often permits a linear separation



Classification: SVM
Popular Kernel Functions

- Linear kernel (standard scalar product):

$$K_{\text{Linear}}(\mathbf{x},\mathbf{x}') = \langle \mathbf{x},\mathbf{x}' \rangle$$

- Gaussian radial basis function:

$$K_{\text{Gaussian}}(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$$

- Polynomial kernel:

$$K_{\text{Polynomial}}(\mathbf{x},\mathbf{x}') = (\langle \mathbf{x},\mathbf{x}' \rangle + 1)^d$$

- Tanimoto kernel:

$$K_{\text{Tanimoto}}(\mathbf{x},\mathbf{x}') = \frac{\langle \mathbf{x},\mathbf{x}' \rangle}{\langle \mathbf{x},\mathbf{x} \rangle + \langle \mathbf{x}',\mathbf{x}' \rangle - \langle \mathbf{x},\mathbf{x}' \rangle}$$

Classification: SVM

- SVMs have hyperparameters that influence the complexity of a model:
  - Coefficient controlling the sensitivity to errors
  - Some kernels like Gaussian or polynomial kernel are parameterized

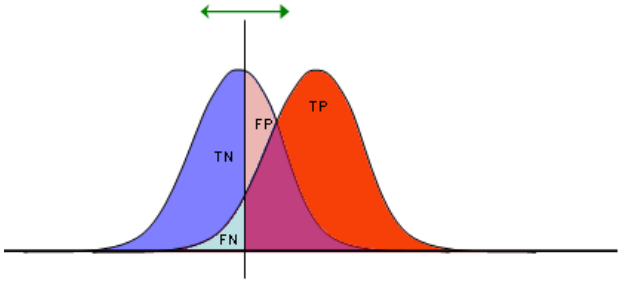
Performance measures

Confusion matrix	Predicted class: Negative	Predicted class: positive
True class: Negative	True negatives (TN)	False positives (FP)
True class: Positive	False negatives (FN)	True positives (TP)

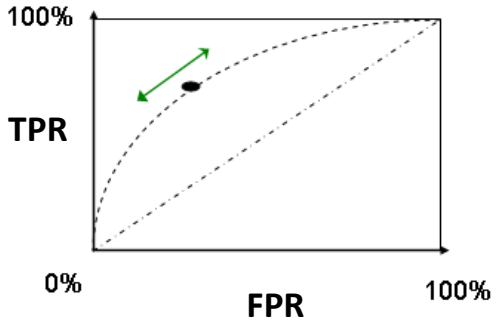
- Sensitivity (true positive rate) $TPR = \frac{TP}{TP+FN}$
- Specificity (true negative rate) $TNR = \frac{TN}{TN+FP}$
- Precision (positive predictive value): $PPV = \frac{TP}{TP+FP}$
- Accuracy: $Acc = \frac{TP+TN}{TP+FN+FP+FN}$
- Balanced accuracy: $Acc_B = 0.5\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) = 0.5(TPR + TNR)$
- F1-score: $F_1 = 2\frac{PPV \cdot TPR}{PPV+TPR}$ (harmonic mean of PPV and TPR)
- Matthews correlation coefficient: $MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TN+FP)(FN+TP)(TN+FN)(TP+TP)}}$

Receiver operating characteristic (ROC)

- Some ML methods (can) yield scores or probabilities of a class
- A variable threshold is used for categorization
- ROC:
  - Vary threshold
  - Plot FPR (x) vs. TPR (y)
- A curve above diagonal indicates positive performance
- Random classification corresponds to diagonal



TP	FP
FN	TN
1	1



Model evaluation

- How can we tell whether a model is good?
  - a number of metrics exist for measuring the performance of models
    - classification: (balanced) accuracy, precision, recall, ROC, correlation coefficient,…
    - regression: mean squared/absolute error
    - clustering: silhouette coefficient,…
- A model might be good on the training data, will it be good on test data?

Model evaluation

- Cross validation:
  - select specific model and parameter settings
  - split training data into n parts, repeatedly
    - retain 1 part, train on n-1 parts
    - measure performance on the 1 part, which was not part of the training
  - assign the average performance

- Cross validation properties:
  - works on the internal training set
  - performance is evaluated on data not used for training
  - checks the generalization ability of the model

Parameter optimization / Model selection

- Cross validation can be directly used to compare different ML methods

- Many ML methods possess hyperparameters
  - control the complexity of a model
  - cross validation can and should be used to tune these parameters

- Few hyperparameters can be tested using grid search
  - systematically explore possible parameter values
  - determine performance with cross validation
  - choose best settings for final model
- Other strategies exist
  - random search, gradient-based optimization, …

b-it LSi
Life
Science
Informatics

LIMES
Life & Medical Sciences Institute

ML for Virtual Screening: Data, Representations and Metrics

UNIVERSITÄT BONN

Machine Learning in Chemoinformatics

Central theme
Investigation of the relationships between similarity and properties of molecules

Similarity may refer to

- structure
- shape
- physicochemical properties
- pharmacophoric features
- ...

Properties may refer to

- biological activity
- target selectivity
- oral availability
- toxicity
- ...

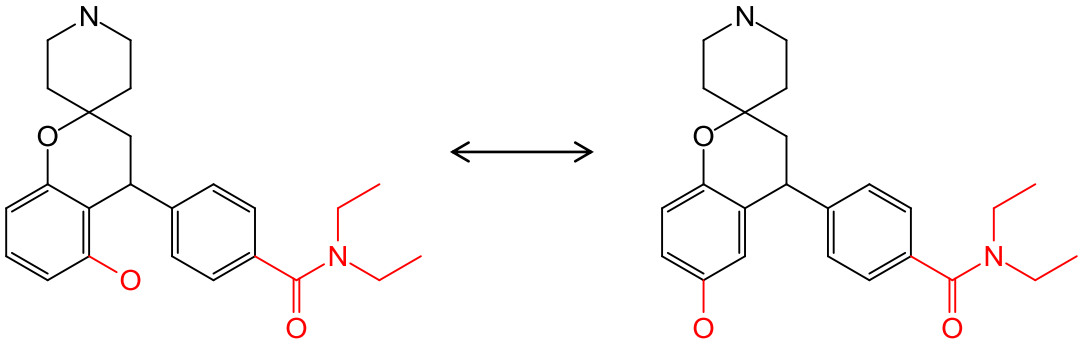
Machine Learning in Chemoinformatics

Central theme
Investigation of the relationships between similarity and properties of molecules

General rule

Similarity property principle

Similar structures show
similar activities



pK_i 9.08 9.37

Data: Public Sources

- ChEMBL
  - only activity annotations
  - analog series bias
  - for specific target: not representative sample of active chemical compounds

- PubChem
  - HTS assay data less biased
  - less confident

Data: Normalization

- Molecular representations can vary
  - protonation states
  - salts
- Consistent representation required
  - washing:
    - remove counter ions
    - consistent protonation states
      - nitrogen, oxygen, sulfur
    - hydrogen suppressed representation
- Activity annotations might not be comparable
  - prefer pK_i over IC50
    - IC50 depends on assay conditions like enzyme and substrate concentrations

Representation & Distance Metric

- Fingerprints
- Descriptors
- 3-D conformations
- Graph structures

- Representation limits what can be perceived
  - only information encoded in the representation is available
- Distance metric depends on the representation

Representation: Descriptors

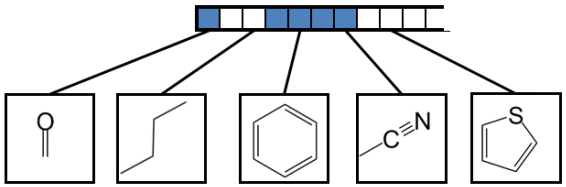
- Numerical property descriptors
  - physicochemical descriptors
    - logP(O/W)
    - molecular weight, …
  - topological descriptors
    - connectivity indices
    - shape descriptors,…
  - count descriptors
    - # of nitrogen atoms
    - # of rings,…
- numerical vector: $(x_1, x_2, \ldots, x_n)$

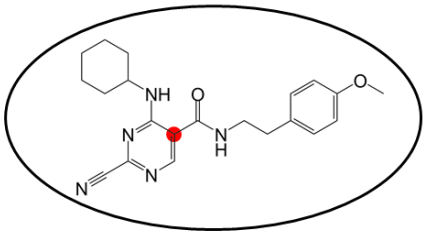
Distance metric: Descriptors

- Euclidean distance
  - $d(x, y) = \sqrt{\sum(x_i - y_i)^2}$
- Important: Normalization
  - normalized Euclidean distance
  - standard deviations of sample data: s_i
  - $d(x, y) = \sqrt{\frac{\sum(x_i - y_i)^2}{s_i^2}}$
- Kernel functions:
  - radial basis function (RBF): $\phi(r) = e^{-\gamma\|x-y\|^2}$

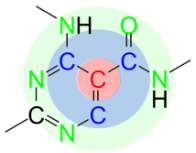
Representation: Fingerprints

Substructure FP





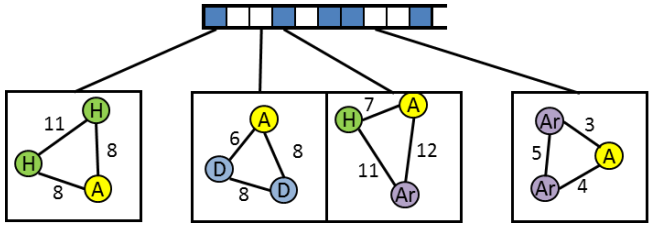
Atom environment FP



1. Layer: c
2. Layer: c(c)(c)C
3. Layer: c(cn)(c(N)n)C(=O)N
...

-20397436 1724213238 946822330

Pharmacophore FP



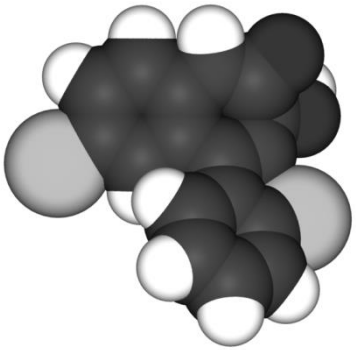
=> binary vector (0,0,1,1,0,1,0,1) or feature set {2,3,5,7}

Distance Metric: Fingerprints

- Hamming/Manhattan distance
  - number of differing features
  - $Tc(A,B) = |A \triangle B| = \sum |a_i - b_i| = \|a - b\|^2$
- Tanimoto coefficient
  - ratio of the number of features two molecules have in common to the number of all occuring features
  - $Tc(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{\sum a_i b_i}{\sum a_i + \sum b_i - \sum a_i b_i}$
- Kernel function (RBF)
  - $r = \|a - b\|$
  - Gaussian: $\phi(r) = e^{-\gamma r^2}$

Representation: Conformations

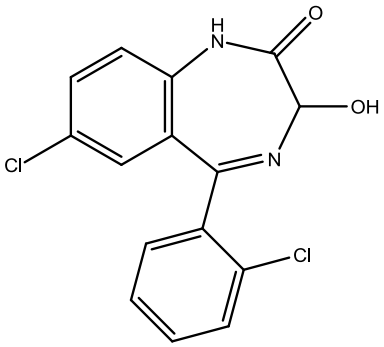
- Volumetric shapes



- Metric:
  - Shape superposition
  - ROCS

Representation: Graph structures

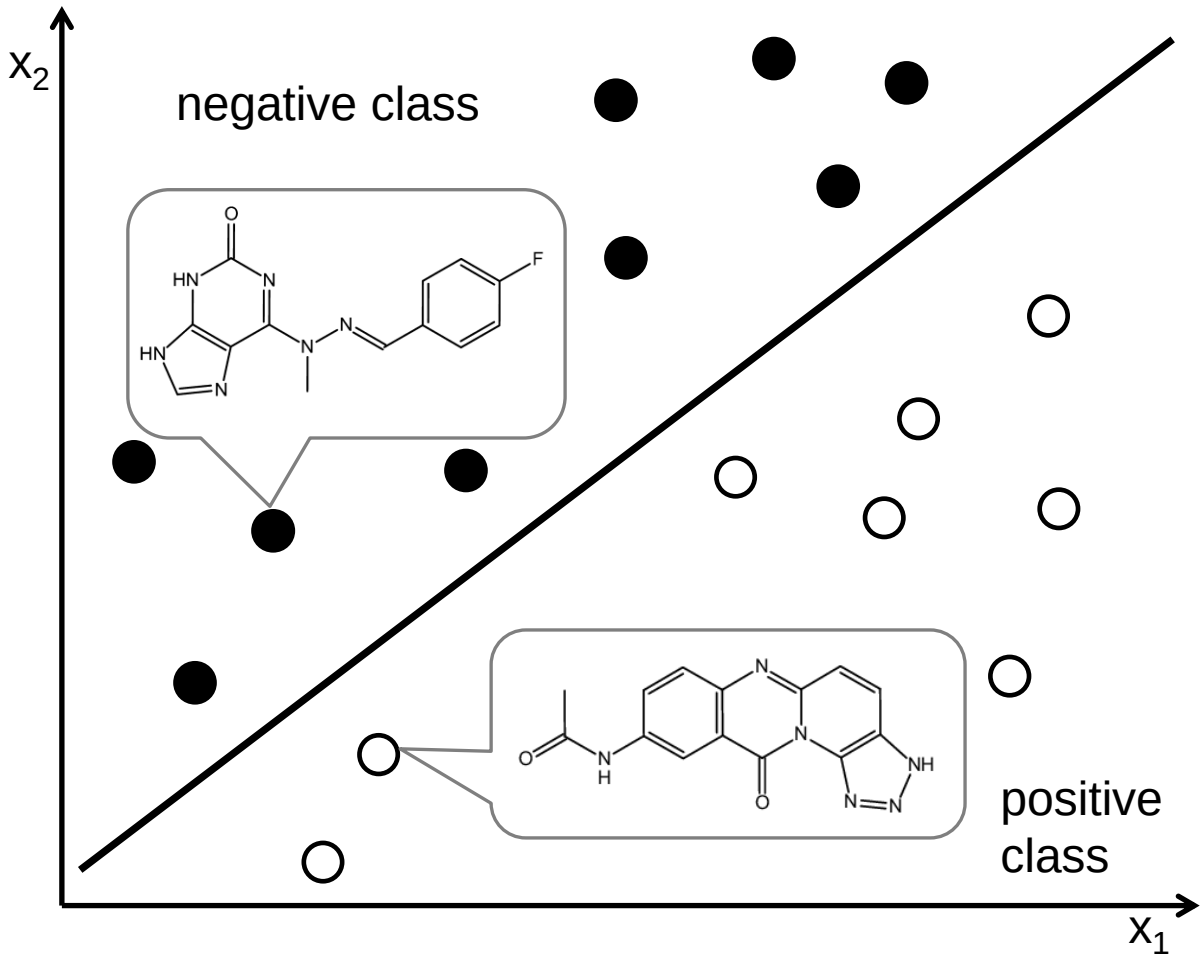
- 2D Graph representations



- Metrics:
  - Maximum common substructure (MCS):
    - ratio of number of bonds in MCS to total number of bonds
  - Graph kernels:
    - random walk kernel

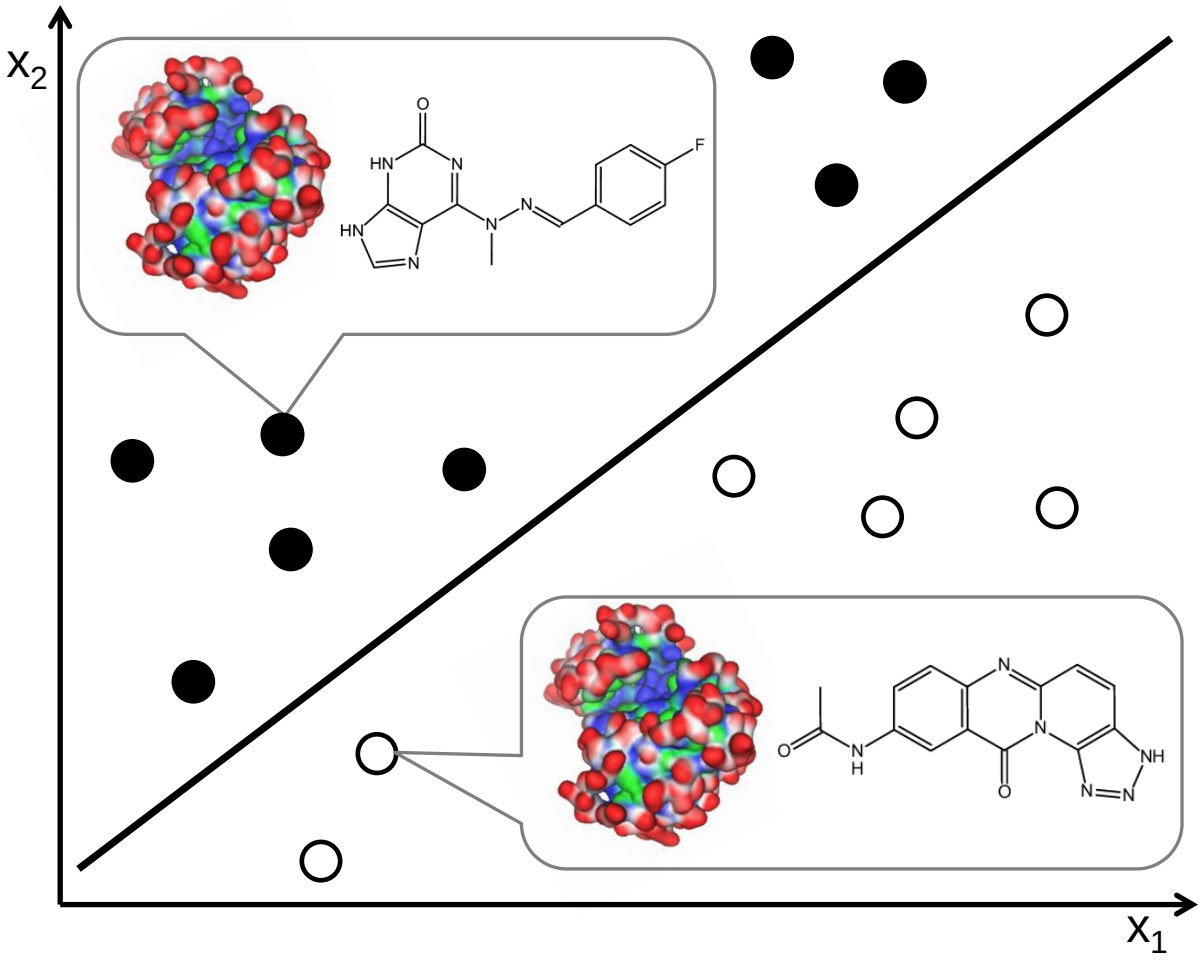
SVM in Compound Space

- Compounds as data points
- Negative class: inactive compounds
- Positive class: active compounds
- Compound reference space described by FP features



SVM in Target-Ligand Space

- Data points are target-ligand pairs
- Positive class: active compounds with true targets
- Negative class: inactive compounds with pseudo targets



Target-Ligand Kernel (TLK)