Problems about Data Mining
Draft

Data Mining Solution Committee(DMSC) @ YB Theory Institution(YBTI)

May 10, 2018

Learning and search methods

1) What is the loss function? Give 3 examples(Least Square, Logistic, Hinge) and describe their shapes and behaviors.

   Loss function: Loss function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. In machine learning loss function is a function that measures the difference between the prediction result and the true result

   Shape:

   a. Least Square: A quadratic surface

   b. Logistic: Sigmoid
   If "Logistic" means Logistic loss, then for an instance x and its label y and hypothesis $h(x)$:

   $$E(x) = \frac{1}{ln2}ln(1 + e^{-yh(x)}) \tag{1}$$

   $$y\log h(x) + (1 - y)\log[1 - h(x)]$$

   c. Hinge: $\max(0, 1 - z)$

   Behaviors:

   a. Least Square: It makes the assumption that the data distribution is Gaussian distribution and uses a homogeneous way to view data.

   b. Logistic: More robust than Least Square. No assumption about data distribution.

   c. Hinge: More robust. No universal assumption about distance...(YB's points of view)

2) Using these losses to approach the actual linear boundary, inevitably some risks will be incurred; give 2 different approaches to remedy the risk using the SVM-based hinge loss as an example.

   a. Using soft-margin

   b. Using L1 or L2 norm to reduce dimension(YB's points of view)

    c.   Using kernel method

3) How many possible models are there given a set of training data? What is the key assumption of PAC learning for model selection?

problem1: Infinite if no restriction on error.
problem2: The object concept c exists and is unique.(YB points of view)
problem2:

- Noise-free dataset
- The solution set is subset the hypothesis set: $C \subseteq H$.

4) Describe biases and variances issue in learning, and how can we select and validate an appropriate model?

Suppose the true target given x is $t(x)$, and our model is $h(x)$, the probability density of x is $p(x)$. We use the square loss, then:

$$E(L) = \int (h(x) - t(x))^2 p(x) dx$$

$$= \int (h(x) - E[t(x)] + E[t(x)] - t(x))^2 p(x) dx$$

$$= \int (h(x) - E[t(x)])^2 p(x) dx + \int (E[t(x)] - t(x))^2 p(x) dx$$

The second term is the noise of data label. So we only consider the first term

$$\int (h(x) - E[t(x)])^2 p(x) dx = \int (h(x) - E[h(x)] + E[h(x)] - E[t(x)])^2 p(x) dx$$

$$= \int (h(x) - E[h(x)])^2 p(x) dx + \int (E[h(x)] - E[t(x)])^2 p(x) dx$$

The first term is the variance and the second term is the bias^2.

We can use cross-validation or leave one alone to select and validate the model.

5) How to control model complexity in linear and logistic regression?

    (a) Using regularization term.
    (b) Using L1 and L2 norm to reduce dimension. (YB's points of view... same thing as the first one)

6) Using the Least Square as the objective function, we try to find the best set of parameters; what is the statistical justification if the underlying distribution is Gaussian?

Suppose our labels are generated using $y = f(x) + \epsilon$ and $\epsilon \sim N(0, I)$. So $y \sim N(f(x_i), I)$. So the likelihood:

$$L = \prod_i p(y_i|x_i) = \prod_i \frac{1}{(2\pi)^{D/2}} exp\{-\frac{1}{2}(y_i - f(x_i))^T((y_i - f(x_i))\} \tag{2}$$

Where D is the dimension of x. The log likelihood

$$\log L = -\frac{1}{2}\sum_i (y_i - f(x_i))^T((y_i - f(x_i)) + C \tag{3}$$

Where C is some constance. To max log likelihood is to min negative log likelihood:

$$\min \quad -\log L = \frac{1}{2}\sum_i (y_i - f(x_i))^T((y_i - f(x_i)) + C \tag{4}$$

or

$$\min \quad \frac{1}{2}\sum_i (y_i - f(x_i))^T((y_i - f(x_i)) \tag{5}$$

Which is the Least Square method.

7) What does the convexity means in either Least Square-based regression or Likelihood-based estimation?

Convexity means the global optimum is unique and we can use Gradient-based method easily to find it.

8) Gradient Descent has a number of different implementation, including SMO, stochastic methods, as well as a more aggressive Newton method, what are some of the key issues when using any Gradient-based searching algorithm?

- The value of hyper-parameters like learning rate (step size). How to jump out of the local minimum. The convexity of the problem.
- parallel computation and speed up.

9) What are the five key problems whenever we are talking about modeling (Existence, Uniqueness, Convexity, Complexity, Generalizability)? Why they are so important?

Existence shows whether our model can converge, Uniqueness shows the difficulty of training. If the problem is convex, we can solve it easily and the global minimum always exists and is unique, Complexity shows the cost of training, Generalizability shows whether our model can achieve good result in new in test dataset and is robust in solving real world problems.

10) Given a probabilistic interpretation for logistic regression, how is it related to the MLE-based generative methods?

Logistic regression learns the posterior distribution $p(y|x)$ directly, is a discriminative method. The MLE-based generative method makes the assumption that the class density $p(x|C)$ is a Gaussian distribution and all these density shares a same covariance matrix. Then using MLE to get the parameters. The Logistic regression and the MLE-based generative method have similar formula:

$$p(y = 1|x) = \frac{1}{1 + exp\{-\theta^T x\}} \tag{6}$$

Logistic regress without regularizer $\leftarrow$ Gaussian Naive Bayes

11) Compare the generative and discriminative methods.

    a. Generative methods try to model joint probabilistic distribution using Bayes formula, while discriminate methods try to model conditional probabilistic distribution.

    b. In general, generative methods requires much more training instances than discriminate methods, and thus suffers higher computational complexity.

    c. However, generative methods usually provide us with more insight into how data is generated.

    d. Discriminative methods can either have probabilistic interpretation or not. Generative models are purely based on probability theories.

12) For the regular and multinomial Naive Bayes, what are their key assumptions? (The second problem has been ruled out)Why the multinomial method can be more context sensitive?

Key assumptions:
There must exist a universal prior that is accepted by every one. (according to YB)

13) **(important)** What are the key advantages of linear models? What are the key problems with the complex Neural Network?

- Key advantages of linear model: Such framework minimizes interactions between different factors, and also has very low computational complexity.
- Key problems with complex NN:
  - It sustains the curse of combinatorial explosion such as network topology and a dramatic huge number of parameters.
  - Multiplication decreases the degree of convexity and therefore the model becomes more sensitive to the initial value.

14) What are 3 alternative to approach a constrained maximization problem?

    1. Solving its dual problem (Lagrange Multiplier)
    2. Find its equivalent problems (modify objective function)
    3. Using kernel tricks

15) What is the dual problem? What is strong duality?

Given an optimization problem:

$$\min_x f(x)$$
$$s.t. \ g_i(x) \le 0, i = 1, ..., k$$
$$h_i(x) = 0, i = 1, ..., l$$

Define its Lagrange function as

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{k} \alpha_i g_i(x) + \sum_{i=1}^{l} \beta_i h_i(x)$$

- Primal problem: $\min\limits_{x} \max\limits_{\alpha \geq 0, \beta} L(x, \alpha, \beta)$
- Dual problem: $\max\limits_{\alpha \geq 0, \beta} \min\limits_{x} L(x, \alpha, \beta)$
- Strong duality:
  For a minimization problem, denote the optimal value of the primal problem by p^*, and correspondingly the optimal value of the dual problem by d^*. We always have $d^* \leq p^*$. The strong duality means $d^* = p^*$.

16) What are the KKT conditions? What is the key implication of them? Including the origin of SVM. (What is the key implication = What is support vector)

Consider an optimization problem problem:

$$\min_{x} f(x)$$
$$s.t. \ g_i(x) \leq 0, i = 1, ..., k$$
$$h_i(x) = 0, i = 1, ..., l$$

Its Lagrangian function is defined as

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{k} \alpha_i g_i(x) + \sum_{i=1}^{l} \beta_i h_i(x)$$

KKT conditions: for $\forall i \in \{1, ..., k\}$

$$\frac{\partial L}{\partial x} = 0, \frac{\partial L}{\partial \beta} = 0$$
$$g_i(x) \leq 0$$
$$\alpha_i \geq 0$$
$$\alpha_i g_i(x) = 0$$

The original optimization problem is equivalent to optimizing its Lagrangian function constrained by the KKT conditions.
In the case of SVM, the optimization problem is:

$$\min_{w,b} \|w\|$$
$$s.t. \ 1 - w^T x_i + b \leq 0$$

the KKT conditions are:

$$\alpha_i \geq 0$$
$$y_i(w^T x_i + b) - 1 \geq 0$$
$$\alpha_i(y_i(w^T x_i + b) - 1) = 0$$

By solving its dual problem, we get $w = \sum_{i=1}^{m} \alpha_i y_i x_i$, then the final model:

$$f(x) = w^T x + b = \sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$$

For any data point (x_i, y_i), if $\alpha_i = 0$, it won't appear in the final trained model. If $\alpha_i > 0$, then it's a support vector lying on the border of the maximum margin. Finally, only support vectors will appear in the formula for prediction, which implies the nature of sparsity of SVM.

17) What is the idea of soft margin SVM? How it is a nice example of regularization?

Soft margin SVM allows misclassification by introducing penalty on those misclassified cases. It improves the ability to tolerate noisy data and issues a model even when the problem is nonlinear. In soft margin SVM, the optimization object becomes

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} l(y_i(w^T x_i + b) - 1)$$

where $l(\cdot)$ is loss function and C is the penalty constant. $\frac{1}{2}\|w\|^2$ in the above formula can be regarded as a l_2 regularization term.

18) The idea of kernel? Why not much additional computational complexity?

- Notice that in the entire computational process of SVM, x only emerges in the form of inner product $x_i^T x_j$. Therefore, we can introduce a kernel function $k(x_i, x_j)$ to replace the original $\langle x_i, x_j \rangle$. This is usually called kernel trick.
- In essence, a kernel function corresponds to a mapping feature space. As we only replace $x_i^T x_j$ by $k(x_i, x_j)$, rather than first mapping original data point to the new feature space and then computing the inner product, so there is not much additional computational complexity.
- Since $K(x, y) = \phi(x)^T \phi(y)$, the kernel trick actually eliminates some common terms to reduce the computation. For example, the Gaussian Kernel $\exp\{-\frac{|x-y|^2}{2}\}$ actually decreases the computation from infinite many times to finite times

19) [*] What is the general idea behind the kernel? What key computation do we perform? Why is it so general in data modeling?

- General idea: mapping data in the original feature space to a new feature space
- Key computation: replace $x_i^T x_j$ by $k(x_i, x_j)$
- locality, linearity and convexity. (etc. nonlinear $\rightarrow$ linear)

20) [*]Why we often want to project a distance "measure" to a different space?

In many situations, the real data space is usually non-Euclidean. Therefore, we want to project a distance measured in Euclidean space to its real space. (nonlinear $\rightarrow$ linear)

Probabilistic graphical model

1) Compare the graphical representation with feature vector-based and kernel-based representation.

- graphical: intuitive way of representing and visualizing the relationships between variables (conditional independence etc)
- vector-based, kernel-based: statistical view (coordinate transformation, variance, etc) on distribution of data.
- vector-based: dimension reduction.
- kernel-based: projection and coordinate transformation

2) Explain why some time a marginal distribution has to be computed in a graphical model.

- It is the target. Only part of variables interest us.
- It is prerequisite for other tasks. $P(\mathcal{H}|\mathcal{X}) \propto P(\mathcal{X}|\mathcal{H})P(\mathcal{H})$ and finding $Q(\mathcal{H})$ is very import in EM algorithm.
- Sometimes, it is used to eliminate the influence of hidden variables.

3) Why a graphical model with latent variables can be a much harder problem?

From statistical view, latent variables are unobserved, usually indicating complex distribution. In terms of training, both latent variables and model parameters are supposed to be found, which is naturally more difficult than models with no hidden variables

- Most of such kind of problems are solved by EM algorithm which does not guarantee a global solution
- The hidden variables are obtained from expectation which is only an approximation method and not accurate enough.
- The problem is non-convex.

4) What is the key assumption for graphical model? using HMM as an example, how much computational complexity has been reduced because of this assumption?

Key Assumption: conditional independence + Markov property

For HMM, The assumption is that given H_{t-1}, H_t is conditionally independent of $H_{t-2}, H_{t-3}, \ldots$. i.e. $P(H_t|H_{t-1}, H_{t-2}, \ldots, H_1) = P(H_t|H_{t-1})$. This simplification turns the varied and high-cost computation into a matrix multiplication operation, which dramatically drop the computational complexity. From $O(K^T)$ to $O(TK^2)$

5) Why does EM not guarantee a global solution? What is a simple proof for that?

Because the model in maximizing step is not guaranteed to be convex. To illustrate, $\theta = \arg\max_\theta E_{\sim Q}[log \frac{P(X,H;\theta)}{Q(H)}]$ and model $P(X, H; \theta)$ may not be convex and either does the expectation function E. EM is guaranteed to converge to a point with zero gradient which may not be a local minimal (saddle point) and let alone the global optimal.

Jensen Inequality illustrates that $f(E(x)) \leq E(f(x))$ is valid iff f(x) is convex. But the model f here is not guaranteed to be convex. $f(x) = \log g(x)$ and while log is convex, $g(x)$ may not be convex. Therefore, EM algorithm based on Jensen Inequality and coordinate ascend does not guarantee a global solution.

6) Why is K-mean only an approximate and local solution for clustering?

K-means is an instance of EM algorithm, which converges to a zero gradient point. This property only guarantees a local optimal solution.

Since K-means, basing on Gaussian Distribution assumption, is the special case of EM algorithm, it cannot cover all the cases of the given dataset. i.e. There may be some better distribution to model the problem which is the true global optimum. [YB] Even based on Gaussian Assumption, different initial data centers will arrive in different local optimal solutions.

7) **(Important)**How to interpret the HMM-based inference problem from a Bayesian perspective. using the forward/backward algorithm?

HMM falls into a subclass of Bayesian Network named Dynamic Bayesian Network, i.e. the joint inference of a series hidden states can be written as $P(H_{1:t}) = P(H_1)P(H_2|H_1)\cdots P(H_t|H_{t-1})$.

8) Show how to estimate a given hidden state for a given series of observations using the alpha and beta factors. Since

$$\alpha_t(i) \equiv P(O_{1:t}, h_t = H_i)$$
$$\beta_t(i) \equiv P(O_{t+1:T}|h_t = H_i)$$

and

$$P(h_t = H_i|O_{1:T}) \propto P(O_{1:T}|h_t = H_i)P(h_t = H_i)$$
$$\propto P(O_{1:t}|h_t = H_i)P(O_{t+1:T}|h_t = H_i)P(h_t = H_i)$$
$$\propto P(O_{1:t}, h_t = H_i)P(O_{t+1:T}|h_t = H_i)$$

Therefore,

$$P(h_t = H_i|O_{1:T}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_i \alpha_t(i)\beta_t(i)}$$

9) For a Gaussian graphical model, what is the implication of sparsity for such a graphical model? How is such sparsity achieved computationally?

- Most edges have zero weights. i.e. sparse adjacent matrix. (Pruning)
- Applying SVD in dimension reduction. Simplify the computation.

10) What would be the risk using a L1 as a relaxation for the sparsity estimation?

Since it's an approximation, it may not achieve the best result. For $|x_1| + |x_2| = C$, both are sometimes not equal to zero. In this case, the result cannot be the global optimal under L_0 norm, because the original objective here can be actually transformed into minimize $f(x_1, x_2) + 2$.

YB: sequential risk $\Rightarrow$ some weights may be dropped if the order of training data gets wrong.

Dimension reduction and feature representation

1) PCA is an example of dimensional reduction method, give a full derivation of PCA with respect to its eigenvectors; explain SVD and how it is used to solve PCA.

Considering time and laziness, I only give some links here. I will rewrite here when I have more time.
Simple version PCA: https://www.cnblogs.com/steed/p/7454329.html
Full version PCA: https://www.cnblogs.com/hadoop2015/p/7419087.html
SVD: https://www.cnblogs.com/pinard/p/6251584.html
Low-rank approximation:...

To use PCA, we need to find the d largest eigenvalues of the Cov matrix. SVD is the method to get them.

$$\max_{\|u\|=1} \sum \|x^{(i)T} u\|$$
$$\Leftrightarrow \max_{\|u\|=1} \sum u^T x^{(i)} x^{(i)T} u$$
$$\Leftrightarrow \max_{\|u\|=1} u^T \cdot X^T X \cdot u$$
$$X = UDV^T$$
$$\Rightarrow \max_{\|u\|=1} u^T \cdot V D U^T U D V^T \cdot u$$
$$U^T U = I$$
$$\Rightarrow \max_{\|u\|=1} u^T \cdot V D^2 V^T \cdot u$$
$$\|u^T v\| \le \|u\| \cdot \|v\| = 1$$
$$\Rightarrow u_k = v_k$$

Therefore, the result principal vectors u_k equal singular vectors of X, v_k.

2) [*] Compare regular PCA with the low-ranked PCA, what would be advantage using the low-ranked PCA and how it is formulated?

The low-ranked PCA is robust PCA (I guess). In short, the target is

$$min_{A,E} rank(A) + \lambda \|E\|_0 \quad s.t. A + E = D$$

In practice, we release it to a convex problem:

$$min_{A,E} \|A\|_* + \lambda \|E\|_1 \quad s.t. A + E = D$$

$\|A\|_*$ is the nuclear norm of A, the sum of singular values. Robust PCA can recover low rank data from large, noisy matrix. YB says: it's a soft iterative PCA that loses less information.

Extend reading: SPCA can restrict the process of linear combination when calculate the eigenvectors. So that the results have a stronger real-world meaning and easier to be explained.
Sparse PCA: https://blog.csdn.net/zhoudi2010/article/details/53489319

3) For a low rank-regularized PCA, what would be the limit of dimension reduction for a given p and n of your data?

Challenges of computational expediency, statistical accuracy and algorithmic stability when p>>n.

4) What is the key motivation (and contribution) behind deep learning, in terms of data representation?

Deep learning is a kind of data representation learning. It automatically discovers the representations needed for feature detection or classification from raw data. This replaces manual feature engineering and allows a machine to both learn the features and use them to perform a specific task. And DL does well in complex representation.
Split and combination. Learn complex, abstract feature combination.

5) [*] What would be the true features of an object modeling problem? Why does the feature decomposition in deep learning then a topological recombination could make a better sampling? What would be the potential problems making deep learning not a viable approach?

Have strong information, low-rank
DL can dig out the underlying relationship within the data and get the "true" feature
Problems: Basis function selection, ways of combination (topology), BP(gradient vanish and explode), memory (RNN, LSTM), Computational complexity, Over/Under-fitting

6) Explain the importance of appropriate feature selection being compatible with model selection in the context model complexity.

The feature should compatible with the representation ability of the model. Otherwise, over-fitting or under-fitting will be likely to happen.
Higher dimensional data are more likely to over-fitting.
The feature selection should match the model selection.

7) What is the key motivation behind a kernel-based method in data representation?

Use a kernel to map low dimensional data to high dimension and make it linear separable.

8) What would be the ultimate and best representation for a high dimensional and complex problem?

Become low dimension while keep most of the information?
YB: "First decompose the problem then learn the most important and fundamental feature"

9) Given two examples to highlight the importance of selecting appropriate dimensions for feature representations.

head location: reconstruct 2D pic to 3D space.

truth table: only 3 dimensions are critical other boolean variables are useless. (dimension reduction and feature selection)

10) For a typical big data problem($p >> n$), what considerations we will have to take when trying to select an appropriate model(for instance, to perform a SVM)?

dimensionality reduction. Regularization. low D solution to high D problem.

General problems

1) In learning, from the two key aspects, data and model, respectively, what are the key issues we normally consider in order to obtain a better model?

data representation, feature selection, sample complexity
model selection and complexity

2) Why all machine learning problems are ill-posed?

trying to solve inverse problems (from data to model) and infer general rules from few data

3) Describe from the classification, to clustering, to HMM, to more complex graphical modeling, what we are trying to do for a more expressive model?

humans solve complex problems using priori, while machines do so using combinations of basic functions

Using model combination to get a more complex and expressive model, and also fit more complex problem.

4) What are the potential risks we could take when trying to perform a logistic regression for classification using a sparsity-based regularization?

L0: NP-Complete problem
L1: overlooking certain parameters due to different order of data. (sequential risk)

5) What are the potential risks we could take when trying to perform a linear regression using a sparsity-based regularization?

the same as 4)
sequence risk.
measures is not consistent.
model risk: λ can be either too great or too small, causing under fitting or over fitting problems.

6) Give 5(change to 4) different structural considerations a search can be constrained with corresponding simple scalars.

L1 norm, L2 norm, Frobenius norm, Nuclear norm(The sum of singular value).

7) Give all universal, engineering, and computational principles that we have learned in this course to obtain both conceptually low-complexity model and computationally tractable algorithms.

Locality, gradient, linearity, convex, low-rank, combination, binary, priori (Bayes), Markov, expectation, recursion, measure

8) Why data representation is at least equally as important as the actual modeling, the so-called representation learning?

model selection and feature selection are closely associated with one another; data representation needs to be compatible with the model and capture necessary features
learning the combination of features and the relationships between features
/*machine learning tasks such as classification often require input that is mathematically and computationally convenient to process*/

9) How does the multiple-layer structure (deep learning) become attractive again?

People realized that structure cannot be imposed on models, so deep learning first learns the structure of the data and finds the relationships. With more layers, the representation of features becomes richer (?)
/*the increase in computational resources and the utilization of GPU acceleration, big data*/

10) What is the trend for AI research and development for the next 5-10 years?

deep learning theory (???)
"parallel" (???)
reinforcement learning (???)

Questions

We do not revise this section, and the outcome of applying these answers exceeds our responsibility.

1) SVM is a linear classifier with a number of possible risks to be incurred, particularly with very high dimensional and overlapping problems. Use a simple and formal mathematics to show and justify (a) how a margin-based linear classifier like SVM can be even more robust than Logistic regression? (b) How to control the overlapping boundary?

(a) For example, suppose there are only 2 data points x_1 and x_2, belonging to opposite classes. SVM will find the maximal margin between these two points. In mathematical form,

$$\arg\min_{\mathbf{w},b} \mathbf{w}^T\mathbf{w} \quad s.t. y_1(\mathbf{w}^T\mathbf{x}_1 + b) \geq 1$$

$$y_2(\mathbf{w}^T\mathbf{x}_2 + b) \geq 1$$

The maximum margin lies when the line $\mathbf{w}^T\mathbf{x} + b = 0$ passes through the midpoint of x_1 and x_2 perpendicularly.
For logistic regression, it tries to maximize the logistic likelihood function:

$$\arg\max_{\mathbf{w},b} \sum_{i=1}^{2} \left(y_i \ln \frac{1}{1 + e^{\mathbf{w}^T\mathbf{x}_i+b}} + (1 - y_i) \ln \frac{e^{\mathbf{w}^T\mathbf{x}_i+b}}{1 + e^{\mathbf{w}^T\mathbf{x}_i+b}} \right)$$

In this example, logistic regress will get the same result as SVM.
However, if a third point x_3 is added, suppose x_3 has the same sign as x_1, and lies farther from the margin. More formally, let x_3 be defined by the following equation, which is not unique,

$$\mathbf{x}_3^T(\mathbf{x}_2 - \mathbf{x}_1) = 2\mathbf{x}_1^T(\mathbf{x}_2 - \mathbf{x}_1)$$

, and $y_3 = y_1$. This additional point will influence both methods differently.

For SVM, x_3 would only introduce one more restriction $y_3(\mathbf{w}^T\mathbf{x}_3 + b) \geq 1$. The original solution still satisfies this condition, and the result remains to be perpendicular line at midpoint.

But for logistic regression, an additional term $i = 3$ is added to the likelihood function. If x_3 does not lie on the same line as x_1 and x_2, the result is likely to change.

Therefore, we can see that SVM is more robust to unimportant data points lying far from the margin.

(b) When the boundary is overlapping between 2 classes, we can use the hinge loss function to impose a soft margin on SVM.

$$\arg\min_{\mathbf{w},b} \mathbf{w}^T\mathbf{w} + C\sum_i \xi_i \quad s.t. \forall i, y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\forall i, \xi_i \geq 0$$

If all $\xi_i = 0$, the target function is the same as hard margin SVM. When there are overlapping classes and a linear classification is not feasible on all data points. There are some points with $\xi_i > 0$, indicating an acceptable compromise on misclassification.

2) Why a convolution-based deep learning might be a good alternative to address the dilemma of being more selective towards the features of an object, while remaining invariant toward anythings else irrelevant to the aspect of interests? Why a linear regression with regularization would result in features which are usually conceptually and structurally not meaningful?

Convolution networks aim to concentrate on local structures of an object, while not caring about the position of this structure. This is achieved by sharing the same set of parameters between parallel units corresponding to spacial translation.

The structure of a deep neuron network models the classification of objects by a hierarchical set of features. Each feature is restricted to the input on a small fraction of the matrix, and should be invariant if it occurs on different positions. When a convolution network is trained, the same unit on the next layer will be adjusted according to all the possible positions that feature may appear, being invariant to the specific position.

However, if a linear regression is trained with regulations, as linear regression is much simpler than deep networks, the complexity of this model will almost certainly be insufficient to express the real distribution of data. In other words, the result will not be meaningful due to underfitting.

3) There are a number of nonlinear approaches to learn complex and high dimensional problems, including kernel and neural networks. (a) Please discuss the key differences in feature selection between these two alternatives, and their suitability. (b) What are the major difficulties using a complex neural network as a non-linear classifier?

(a) The kernel method deals with nonlinearity by using nonlinear kernel mappings, while neural networks introduces nonlinear activation functions on at least one layer.

Kernel function tries to map data points to a higher dimensional space where they are linearly separable. The idea is to shatter features holistically so that a simple relationship can be found. This methods suits the conditions when data are readily separable after kernel transformation, which requires that the distribution of input data should not be too complex.

Neural networks, conversely, works by decomposing features into small parts and layers, and solves the small problems individually. This assumes a hierarchical structure of features and has proved to suit well in many pattern recognition problems.

(b) It is complex to propagate training error through a neuron network with many layers and units. Stochastic and gradient based methods can be used to handle this problem.

Also, complex neural networks have many interacting parts, which may lead to unstable behavior. It is hard to validate the training result to guarantee its ability of generalization.

4) For any learning problems: (a) why a gradient-based search is much more favorable than other types of searches? (b) what would be the possible ramifications of having to impose some kinds of sequentiality in both providing data and observing results?

    (a) Gradient-based search takes advantage of the local tendency of a neighborhood. It is generally more efficient with faster rate of convergence, and will always end with a local optimum, compared with other searching methods such as simulated annealing or genetic algorithm. Gradient based search is most favorable on convex target functions, where there are efficient methods to find a global optimum.

    (b) Imposing sequentiality would force the neuron network to share its parameters across different parts of the sequence. This gives rise to recursive neuron networks (RNN). As a result, receiving an input of variable length is converted to internal transition of states after receiving just one input at a time.

5) Please use linear regression as the example to explain why L1 is more aggressive when trying to obtain sparser solutions compared to L2? Under what conditions L1 might be a good approximation of the truth, which is L0?

In linear regression, L2 norm scales each dimension of the unregularized result by a factor of $\frac{\lambda_i}{\lambda_i + \alpha}$, where λ_1 is the eigenvalue on this basis. This compresses the result in each direction, and the effect is more obvious on insignificant directions with small eigenvalues. (Eigenvalues are variances on the direction of the corresponding eigenvector, and the basis with a small eigenvalue has many points being together and is not as informative.)

However, L1 norm, instead of multiplying by a small factor, directly subtracts a term $\frac{\alpha}{\lambda_i}$ from the unregularized result, and replace the difference with 0 if it is less than 0. This would force many dimensions with very small eigenvalues to be assigned 0 directly. In comparison, L2 norm would only make these dimensions to be close to, but not exactly 0. L1 norm is more aggressive in the sense of getting a sparse solution with many 0 entries.

L0 norm only cares about the number of non-zero entries on the solution, but not their values. This norm will lead to sparse solutions but is hard to solve by gradient methods. L1 can be used to approach L0 when the distinction between important and insignificant features is large. If the pattern is strong enough on dimensions with large eigenvalues so that a penalization of regularization does not affect the result significantly, L1 norm will select the same set of features as L0.

6) What is the key difference between a supervised vs. unsupervised learnings(where we do not have any ideas about the labels of our data)? Why unsupervised learning does not guaranty a global solution?(use mathematical formulas to discuss).

Supervised learning aims to predict the value or class tag of output y given an input x. Its objective is to learn a mapping from x to y. However, without knowing y, unsupervised learning only aims to learn a simpler form of representation of input x. Hopefully, this representation is in lower dimension, sparser or independent among the dimensions.

Unsupervised learning cannot guarantee a global solution because it inevitable loses some amount of information when trying to find a simpler representation. Suppose we use PCA to reduce dimension given 4 data points (10,0), (-10,0), (0,1), (0,-1).

The input matrix X of PCA is

$$\mathbf{X} = \begin{pmatrix} 10 & 0 \\ -10 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}$$

The largest eigenvector is (1,0) with eigenvalue 200, while the second largest eigenvalue is only 2. Therefore, we only use the x coordinate (projection on the eigenvector with largest eigenvalue) as representation of data.

Now, the question is to find the point with greatest output $z = x + 100|y|$. The PCA representation leaves out y, so we can only use $z = x$ to estimate the output of each node, and (10,0) is expected to have greatest

output $z = 10$. However, in fact, nodes (0,1) and (0,-1) have the greatest output $z = 100$.

7) For HMM, (a) please provide a Bayesian perspective about the forwarding message to enhance an inference (using a mathematical form to discuss), how to design a more generalizable HMM which can still converge efficiently?

    (a) Suppose there are 2 states, X and Y, and 2 observations, A and B, of this HMM model. Suppose the initial distribution of states to be π_0, observation matrix to be O, and transition matrix to be T. Given observation A at time 1, the possibility of the hidden state can be calculated by

$$P(S_1|A_1) = \frac{P(A_1, S_1)}{P(A_1)} = \alpha O(A)\pi_1 = \alpha O(A)T\pi_0$$

This formula is normalized with coefficient $\alpha = 1/P(A_1)$.
Forwarding message improves accuracy of inference by making use of the observations before. Suppose the observation is A at time 0. Using Bayesian inference, the posterior distribution of state at time 0 can be calculated as

$$\pi_0^* = P(S_0|A_0) = \frac{P(A_0, S_0)}{P(A_0)} = \alpha' O(A)\pi_0$$

Therefore, the posterior distribution can be regarded as a refined estimation of the state at time 0 given observation result at this time. The distribution of states at time 1 given A_0 and A_1 is

$$P(S_1|A_0, A_1) = \frac{P(A_1, S_1|A_0)}{P(A_1)} = \alpha'' O(A)T\pi_0^*$$

    (b) Is there anyone who knows the question? Good likelihood, good prior(he said).

8) Using a more general graphical model to discuss (a) The depth of a developing prior-distribution as to its contribution for a possible inference (b) how local likelihoods can be used as the inductions to facilitate the developing inference?

    (a) I do not know what this question is about. Completely.
    (b)

9) Learning from observation is an ill-posed problem, however we still work on it and even try to obtain convex, linear, and possibly generalizable solutions. Please discuss what key strategies in data mining we have developed that might have remedied the ill-posed nature at least in part? Why in general linear models are more robust than other more complex ones?

Soft margin: overlapping class
Kernel method: mapping nonlinear relationship to linear one
Regularization: dimension reduction to reduce complexity
Cross validation: finding a model that generalizes well to testing data

Linear models are closed in terms of combination (any combination of linear models is also linear). Therefore, components of a linear model are independent and are not involved into complex interactions. As a result, linear models do not become overly complex, and avoids consequent stability issues.

Linear models are consistent with the assumption of locality, smoothness of measure in sample space, and often generalizes well to new input.

10) Using logistic regression and likelihood estimation for learning a mixture model(such as the Gaussian Mixture Model), please using Bayesian perspective to discuss the differences and consistencies of the two approaches; why logistic function is a universal posterior for many mixture models?

Suppose the prior probability of Gaussian Mixture model $P(A) = p, P(B) = 1 - p$, and the input $X|A \sim N(\mu_a, \sigma_a^2), X|B \sim N(\mu_b, \sigma_b^2)$.

Given input x,

$$P(x \, fromA) = \frac{P(A)P(x \sim X_A)}{P(x)} = \frac{p\frac{1}{\sqrt{2\pi}\sigma_a}e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}}{p\frac{1}{\sqrt{2\pi}\sigma_a}e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_b}e^{-\frac{(x-\mu_b)^2}{2\sigma_b^2}}}$$

$$= \frac{1}{1 + \frac{(1-p)\sqrt{\sigma_a}}{p\sqrt{\sigma_b}}e^{-\frac{(x-\mu_b)^2}{2\sigma_b^2} + \frac{(x-\mu_a)^2}{2\sigma_a^2}}}$$

Let the feature vector of input data be $\mathbf{x} = (1, x, x^2)^T$, the posterior probability of classification has the form

$$P(x \, fromA) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

, where the weight vector

$$\mathbf{w} = \left(\ln \frac{p\sqrt{\sigma_b}}{(1-p)\sqrt{\sigma_a}} + \frac{\mu_b^2}{2\sigma_b^2} - \frac{\mu_a^2}{2\sigma_a^2}, -\frac{\mu_b}{\sigma_b^2} + \frac{\mu_a}{\sigma_a^2}, \frac{1}{2\sigma_b^2} - \frac{1}{2\sigma_a^2}\right)^T$$

Maximum likelihood estimation estimates the input by returning the class with maximal posterior probability that the input is generated from this class. Logistic regression, however, is trained with a sigmoid loss function, and gets the classification result simply by calculating the value of this sigmoid function.

These two methods handle this problem with different ideas, but they are consistent in that MLE is indeed equivalent to maximizing a logistic target function with properly chosen feature.

In light of the deduction above, the posterior has the form of a logistic function as long as the probability density function of data generation is an exponential function.