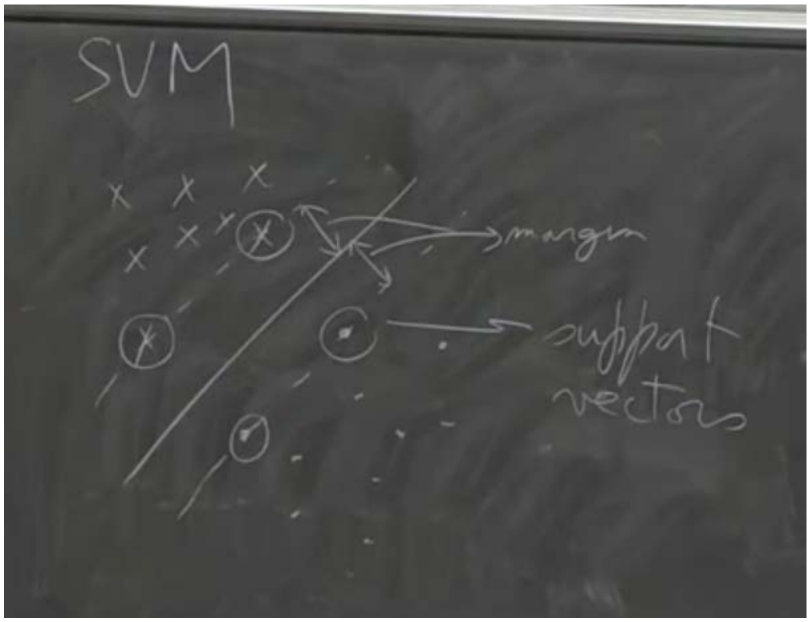
بسم الله الرحمن الرحيم

هنتكلم انهارده علي ال SVM .. الكيرنال ميثودز اللي بتشتغل علي الكلاسيفكيشن .. زي ال SVM .. مهم جداً .. بس النويرال نتورك دخلت بعدو خلصت الدنيا خلاص .. لما اتكلمنا علي الكيرنال ميثوذدز بالنسبه للسكيلابلتي هي cubic في حجم الداتا اللي عندك ... هل احنا لازم نعمل الميثودز ديه بتعتمد علي كل الداتا و لا هنقدر نشتغل علي جزء بس من الداتا .. ال SVM هو نوع من الكيرنال ميثودز وبنعتبره sparse و بيشتغل علي جزء من الداتا و بيرمي الباقي .. قبل ما نخش في الرياضه ونعقد الدنيا ... تعال نشوف نرسم شويه كدا ....

عندنا 2 كلاسيز من الداتا ... عاوزين نجيب linear separator يفصل ال 2 كلاسيز ... الطريقه الي هنفصل بيها هو اننا نقيس المسافه لأقرب نقطه والمسافه ديه هي المارجن .. اصغر مسافه لأقرب نقطه و بعدين نعمل نفس الحوار للناحية التانيه .. ال SVMs بتلاقي linear sepearator line ف النص ... و النقط اللي بتبقا علي خط المارجن بتبقا هي support vector .. النقط هي عباره عن vectors و دول بيدولك linear separator لل decision boundary ..



ال SVMs هي برضو اسمها max margin classifier عشان هي بتلاقي linear line بيماكسيميز المسافه لأقرب داتا بوينت .. يعني الأوبتيمازيشن بروبلم هتبقا هدفها انها ت maximize the margin ... ليه هنختار الماكس مارجن .. عشان الداتا ممكن تبقا نويزي ... فلو هرسم خط بيفصل .. وفي نقط فعلاً قريبه من الخط .. بس لو عندك إيرور فبالتالي النقط ديه ممكن تبقا في الناحيه التانيه من الخط .. بس لو عندك wide margin .. ففي فرصه احسن انك تخلي النقط اللي هتغلط فيها في الناحيه الصح .. عشان نحسب ال distance بتاعت المارجن ... أول حاجه احنا عندنا لينير سيباراتور .. ده اللي هو wT phi(x) ...الدوت بروضكت بيطلعلك يا بوزيتيف يا نيجاتيف علي حسب الدايركشن بتاع ال 2 فيكتورز ... فبالتالي لو عندك نقطه رايحه لكلاس واحد اديلها 1 ولو رايح للكلاس التاني اديلها 1- .... فالفورميلا عشان تحسب المسافه بتاعت أي نقطه ل ال linear separator .. طيب اللي احنا عاوزينو هو اننا نلاقي شوية الويتس اللي هت maximize the margin .. المقصود بماكس المارجن هو المارجن اصلا هي اقل مسافه من النقط للخط ..فال minimum بالنسبه لل datapoint و دت الماكس بالنسبه للويتس عشان تلاقي الخط اللي يديلك أوسع مارجن ... هو ده الأوبجكتيف اللي هنعملو أوبتيمازيشن

Margin

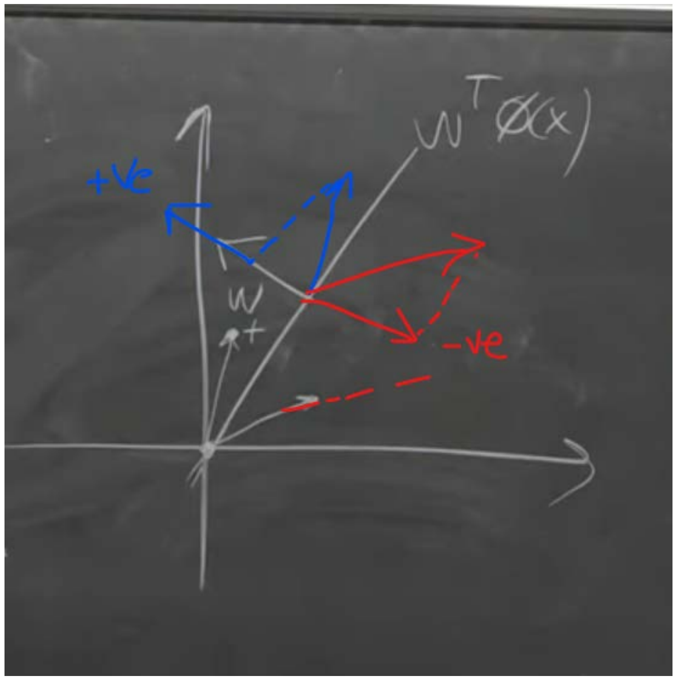
- Linear separator: $\boldsymbol{w}^T \phi(\boldsymbol{x}) = 0$
- Distance to linear separator:

$$\frac{y\boldsymbol{w}^T \phi(x)}{||w||} \quad \text{where } y \in \{-1,1\}$$

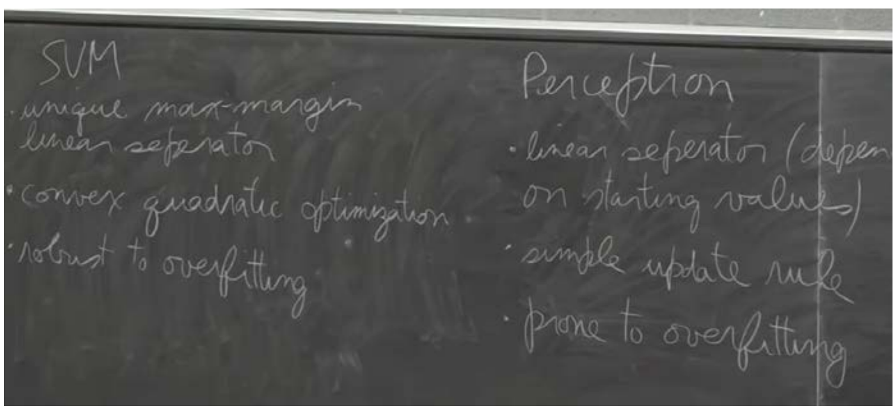
- Maximum margin:

$$max_{\boldsymbol{w}} \frac{1}{||\boldsymbol{w}||} \left\{ \min_n y_n \, \boldsymbol{w}^T \phi(\boldsymbol{x}_n) \right\}$$

الدكتور هيتكلم شويه علي البرسبترون .... هو برضو بيلاقي لينير سيبيريتور ... بس ده بيعتمد علي ال starting values بتاعت الويتس ... فاللي بتوصلو في الآخر هيبقا مختلف علي حسب ال starting values .. المهم ان ال update rule هو انك هتجمع او هتطرح كل misclassified datapoint ...



الدكتور بيقول ان البرسيبترون prone to overfitting .. لو عندنا SVM احنا عندنا unique max-margin و عندنا linear separator .... ال update rule هي مش سهله ... و بتبقا عندك convex quadratic optimization و الدكتور بيقول انو roubust to overfitting ...



Maximum Margin

- Unique max margin linear separator

بل

$$max_{\boldsymbol{w}} \frac{1}{||\boldsymbol{w}||} \left\{ \min_n y_n \, \boldsymbol{w}^T \phi(\boldsymbol{x}_n) \right\}$$

Practical

- Alternatively, we can fix the minimal distance to 1 and minimize $||\boldsymbol{w}||$

$$\boxed{\min_{\boldsymbol{w}} \frac{1}{2} ||\boldsymbol{w}||^2}$$

at Least dist 1

s.t. $y_n \, \boldsymbol{w}^T \phi(\boldsymbol{x}_n) \geq 1 \quad \forall n$

- This is a convex quadratic optimization problem that can easily be solved by many optimization packages

طيب دلوقت عاوزين نشتغل بال dual representation ... احنا مش لازم ندفع تمن لل feature space جديد .. انما هنستخدم new mapping phi وهنعمل ريمابنج للداتا من الأوريجنال سبيس لل new space .. وهناك كل الحسابات هتبقا in terms of dot product ... وتقدر تبدّلها بكيرنال فانكشن .. ده اللي عملناه ل Gaussian processes و ديه كانت مجرد ال dual version بتاعت ال Bayesian linear regression وهنا مع ال SVM في برضو dual version ... في advantage للدوال ده .. ان ال optimization is sparse عشان الداتا بوينتس اللي علي المارجن بس هي اللي تهمني .. الأوبتيمايزيشن تاني هي اننا نقلل ال w مع الوضع في الاعتبار الشرط اللي اشتغلنا عليه ...

المهم دلوقت اللي نقدر نعملو هو اننا نعيد كتابة الحوار ده بطريقه اننا ن move the constraint into the objective .... في ال optimization مش بيبقا convenient ان يبقا عندنا constraint .. مثلا لو بصينا علي جريدينت ديسسنت .. وانت بتحسب direction بتاع ال steepest descent خد خطوه هناك .. لو عندك كونسترينت .. انت بتليميت ال gradient و بتلميت الأوبتيمايزيشن فالدنا بتبقا معقده أكتر .. فشيل ال constraints اللي عندك عشان الدنياي تبقا اسهل ... المهم انك ممكن تطلع ب equivalent optimization problem وهنا احنا هنشيل ال constraint ... هنحطه في ال objective ك penalty term ...... وهنا بنعمل lagrangian وديه new objective .. اللي هي هتبقا ماسكه الترم بتاع المنيمم عادي .. انما ال constraints دخل علي هيئة penalty .. هتقيس المسافه اللي عندك .. لو مش أكبر من واحد فهتحط penalty و الماجنتيود بتاعها هيبقا an ... لو المسافه كان أكبر من الواحد .. حط ال an = 0 عشان ميبقاش عندنا أي penalty ...

Dual derivation

- Transform constrained optimization

$$\min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2 \quad \text{s.t. } y_n\, \boldsymbol{w}^T\phi(\boldsymbol{x_n}) \geq 1 \quad \forall n$$

into an unconstrained optimization problem

- Lagrangian

اختار a بت max ال lagrangian

$$\boxed{\max_{\boldsymbol{a}\geq \boldsymbol{0}}} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{a})$$

where $L(\boldsymbol{w}, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_n \underbrace{a_n[y_n\, \boldsymbol{w}^T\phi(\boldsymbol{x_n}) - 1]}_{\text{penalty for violating the } n^{th} \text{ constraint}}$

newname
10/14/2021 8:16:46 AM

هنحل كلوزد فورم ال inner minimization ... ده بيحصل في كلوزد فورم .. ديه convex quadratic objective ... خد بالك ان quadratic هنا عشان ال L2Norm .. الكونفكس معناه ان عندك جلوبال اوبتيمال منيمم ... المهم في الآر وصلنا ال w هتبقا عامله ازاي ... بقت عباره عن لينير كومباينيشن من الداتا بوينتس في النيو فيتشر سبيس .. ال an هنا مش بتساوي صفر .. والداتا بوينتس هنا هم السابورت فيكتورز .. فال w هتبقا linear combination of support vector ...

- Solve inner minimization: $\min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{a})$

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}||\boldsymbol{w}||^2 - \sum_n a_n[y_n\, \boldsymbol{w}^T\phi(\boldsymbol{x_n}) - 1]$$

- Set derivative to 0

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_n a_n y_n \phi(\boldsymbol{x}_n)$$

- Substitute $\boldsymbol{w}$ by $\sum_n a_n y_n \phi(\boldsymbol{x}_n)$ in $L(\boldsymbol{w}, \boldsymbol{a})$ to obtain:

$$L(\boldsymbol{a}) = \sum_n a_n - \frac{1}{2}\sum_n \sum_{n'} a_n a_{n'} y_n y_{n'} k(\boldsymbol{x}_n, \boldsymbol{x}_{n'})$$

Dual Problem

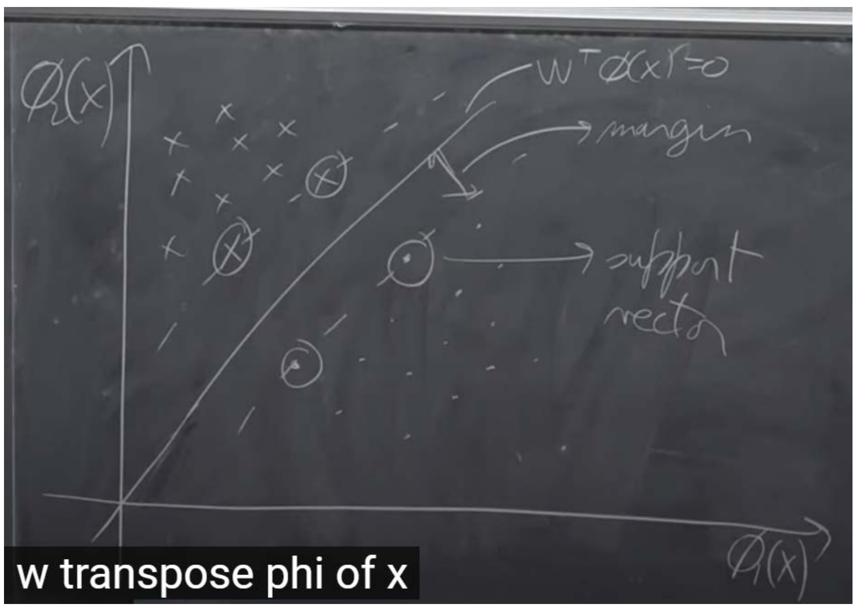
- We are then left with an optimization in $\boldsymbol{a}$ only known as the **dual problem**

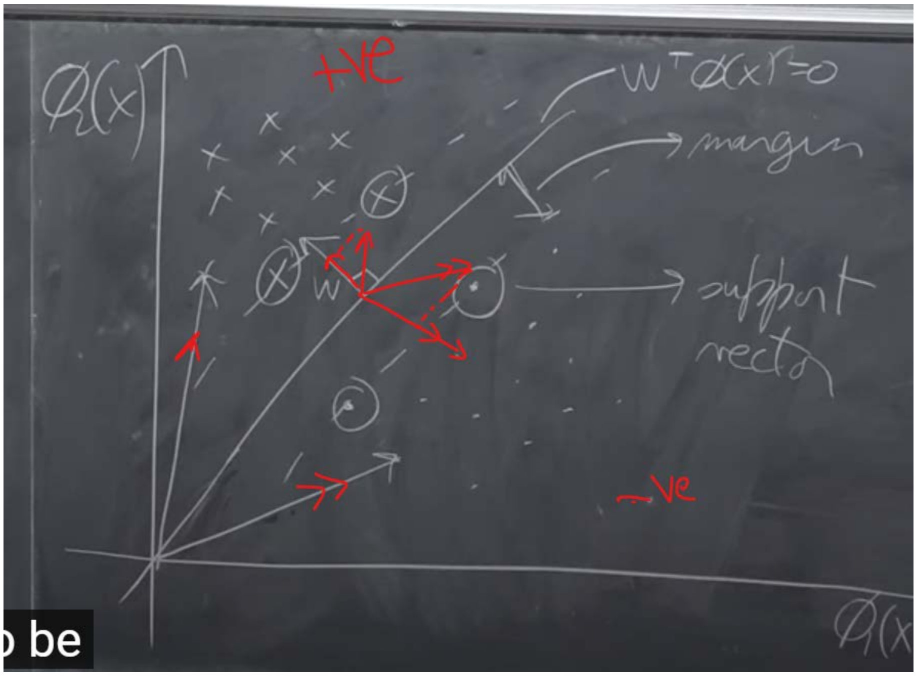
$$\max_{\boldsymbol{a}} L(\boldsymbol{a})$$

$$\text{s.t. } a_n \geq 0$$

عشان نبت
optimize
a مش w

- **Sparse optimization**: many a_n's are 0





Classification

newname
10/14/2021 9:25:13 AM

kernel functions captures the similarity between pairs of data point ... it's dot product into feature space.. If data points are similar, their vectors are going to point more or less in the same direction, there for the kernel function will be positive. If not similar, the kernel function is negative ...

- Primal problem

$$y_* = sign(\boldsymbol{w}^T \underbrace{\phi(\boldsymbol{x}_*)}_{\text{features}})$$

- Dual problem

$$y_* = sign\left(\sum_n a_n y_n \phi(\boldsymbol{x}_n)^T \phi(\boldsymbol{x}_*)\right)$$

$$y_* = sign\left(\sum_n a_n y_n k(\boldsymbol{x}_n, \boldsymbol{x}_*)\right)$$

sum of degree for similarity